

Petra Perner (Ed.)

Machine Learning and Data Mining in Pattern Recognition

15th International Conference on Machine
Learning and Data Mining, MLDM 2019,
New York, NY, USA, July 20-25, 2019
Proceedings
Volume I

ibai Publishing

www.ibai-publishing.org

Volume Editor

Petra Perner
Institute of Computer Vision and Applied Computer Sciences, IBaI
PF 30 11 14
04251 Leipzig
E-mail: pperner@ibai-institut.de

ISSN (Print) 1864-9734
ISSN (Online) 2699-5220
ISBN 978-3-942952-62-0



The German National Library listed this publication in the German National Bibliography.
Detailed bibliographical data can be downloaded from <http://dnb.ddb.de>.

ibai-publishing
Prof. Dr. Petra Perner
PF 30 11 38
04251 Leipzig, Germany
E-mail: info@ibai-publishing.org
<http://www.ibai-publishing.org>

Copyright © 2019 ibai-publishing
ISSN 1864-9734
ISBN 978-3-942952-62-0

All rights reserved.
Printed in Germany, 2019

15th International Conference on Machine Learning and Data Mining, MLDM 2019

July 20-24, 2019, New York, USA
www.mldm.de

Chair

Petra Perner
Institute of Computer Vision and Applied Computer Sciences, IBA
Germany

Program Committee

Reneta Barneva	The State University of New York at Fredonia, USA
Michelangelo Ceci	Universtiy of Bari, Italy
Ireneusz Czarnowski	Gdynia Maritime University, Poland
Roberto Corrizo	Universtiy of Bari, Italy
Christoph F. Eick	Universtiy of Houston, USA
Mark J. Embrechts	Rensselaer Polytechnic Institute and CardioMag Imaging, Inc, USA
Ana Fred	Technical University of Lisboa, Portugal
Giorgio Giacinto	University of Cagliari, Italy
Aminata Kane	Concordia University, Canada
Piet Kommers	University of Twente, The Netherlands
Olga Krasotkina	Russian State University, Russia
Dimitris Karras	Chalkis Institute of Technology, Greece
Adam Krzyzak	Concordia University, Canada
Valerio Pascucci	University of Utah, USA
Gianvito Pio	University of Bari, Italy
Francis E.H. Tay	National University of Singapore, Singapore
Turki Turki	King Abdulaziz University, Saudi Arabia
Zeev Volkovich	ORT Braude College of Engineering, Israel
Patrick Wang	Northeastern University, USA

Preface

The fifteenth event of the International Conference on Machine Learning and Data Mining MLDM was held in New York (www.mldm.de) running under the umbrella of the Worldcongress “The Frontiers in Intelligent Data and Signal Analysis, DSA2019” (www.worldcongressdsa.com).

For this edition the Program Committee received 245 submissions. After the peer-review process, we accepted 65 high-quality papers for oral presentation. The topics range from theoretical topics for classification, clustering, pattern mining to specific data mining methods for the different multimedia data types such as image mining, text mining, video mining and web mining. Extended versions of selected papers will appear in the *International Journal Transactions on Machine Learning and Data Mining* (www.ibai-publishing.org/journal/mldm).

A tutorial on Data Mining and a tutorial on Case-Based Reasoning were held before the conference that took pleasure to high participation of researchers and practitioners from industry, social and public services.

We like to thank all presenter for your high-quality presentations and the audience for your high-professional questions and inspiring comments. All that has made the conference to a living and dreadful event. The presenters and the audience went home with a full bag of new insights into different topics the research and inspiring ideas for new work and research. Besides that, gave the banquet an excellent opportunity to network among the participants and set up new co-operations.

We like to thank all reviewers for their highly professional work and their effort in reviewing the papers. We also thank members of Institute of Applied Computer Sciences, Leipzig, Germany (www.ibai-institut.de) who handed the conference as secretariat. We appreciate the help and understanding of the editorial staff of ibai-publishing house (www.ibai-publishing.org) that prepared and published the proceeding books in two volumes.

We invite you to join us in 2020 in New York to the next Worldcongress (www.worldcongressdsa.com) “The Frontiers in Intelligent Data and Signal Analysis, DSA2020” that combines under his roof the following three events: International Conferences Machine Learning and Data Mining MLDM, the Industrial Conference on Data Mining ICDM , and the International Conference on Mass Data Analysis of Signals and Images in Artificial Intelligence and Pattern Recognition with Application in with Applications in Medicine, r/g/b Biotechnology, Food Industries and Diagnostics, Biometry and Security, Agriculture, Drug Discover, and System Biology MDA-AI&PR.

Table of Content

Arabic Rule-based Named Entity Recognition System Using GATE <i>Hatem M. ElSherif, Khaled Mohammad Alomari, Ahmad Qasim AlHamad and Khaled Shaalan</i>	1
An Unsupervised Statistical Moving Shadow Detection Method for Video Analysis <i>Hang Shi and Chengjun Liu</i>	16
Social Activism Analysis: An Application of Machine Learning in the World Values Survey <i>Francielle M. Nascimento, Dante A. C. Barone, and Henrique Carlos de Castro</i>	28
Deep Dilated Convolutional Nets for the Automatic Segmentation of Retinal Vessels <i>Ali Hatamizadeh, Hamid Hosseini, Zhengyuan Liu, Steven D. Schwartz, and Demetri Terzopoulos</i>	39
Clonal Selection Algorithm Applied to Object Recognition in Mobile Robots <i>Jose Guillermo Guarnizo and Luis Fernando Nino</i>	49
Customized use of Expectation Maximization for Glioma Identification from Brain MRIs <i>Nidhi Gupta, Pushpraj Bhatele, and Pritee Khanna</i>	63
A Novel Phonetic Algorithm for Predicting Chinese Names using Chinese PinYin <i>Hua Zhao and Fairouz Kamareddine</i>	78
Detecting Customer Churn Signals for Telecommunication Industry Through Analyzing Phone Call Transcripts with Recurrent Neural Networks <i>Junmei Zhong, William Li</i>	93
A comparison of linear and machine learning models for the simulation of soil moisture <i>Milan Cisty, Frantisek Cyprich and Veronika Soldanova</i>	104
Vehicle Classification in Video Using Deep Learning <i>Mohammad O. Faruque, Hadi Ghahremanzhad, and Chengjun Liu</i>	117
A Model for Diagnosis of Alzheimer’s Disease Using Haralick Texture Features and Meta Feature Selection <i>Iago Richard Rodrigues Silva, Marilia Nayara Clemente de Almeida Lima, Wellington Pinheiro dos Santos, and Roberta Andrade de Araújo Fagundes</i>	132

Mining Data Stream to Detect Behavior Change in a Real-Time Strategy Game <i>Eldane Vieira, Rita Maria Silva Julia, and Elaine Ribeiro de Faria</i>	146
Packet2Vec: Utilizing Word2Vec for Feature Extraction in Packet Data <i>Eric L. Goodman, Chase Zimmerman, and Corey Hudson</i>	161
Handbags Classification Model via Deep Learning <i>Dieinison J. F. Braga, Leodecio Braz S.S., Criston Pereira de Souza1, and Ticiana L. Coelho da Silva</i>	176
Long Short-Term Memory-based Multi-Period Price Prediction for Portfolio Management <i>Zhengyong Jiang and Frans Coenen</i>	187
Fraud Detection Using Explainable Machine Learning Algorithms <i>Luciano C M Andrade, Andr'e C P L F Carvalho</i>	201
A Fast AEMST Algorithm for High-Dimensional Datasets by Removing Redundant Distance Computations <i>Xiaochun Wang, Xia Li Wang, Xuan Xiong Lin</i>	216
Server Failure Prediction Framework for IT Infrastructure Supporting Critical Applications <i>Mythili Krishnan and Madhan Kumar Srinivasan</i>	229
A New Minimum Spanning Tree Algorithm Constructed in A Backward Fashion <i>Aozhong Wang, Xiaochun Wang, Xia Li Wang</i>	243
Analysis of Linear and Non-Linear Classifiers in Imbalanced Data to Predict Diabetes Induced Complications <i>Tahsinur Rahman, Aniq Zaida Khanom, Sheikh Mastura Farzana, Sharowar Md. Shahriar Khan and Dr. Md. Ashrafal Alam</i>	255
FF-SVM: New FireFly-based Gene Selection Algorithm for Microarray Cancer Classification <i>Nada Almugren and Hala Alshamlan</i>	270
Fast K-medoids Clustering Algorithm Using Triangle Inequality <i>Xiaochun Wang, Xia Li Wang, Xuan Xiong Lin</i>	282
Design and Implementation of an Intrusion Detection System using Deep Neural Network <i>Hyun-chul Chang and Sungbum Park</i>	296

VII

Community detection in networks with fuzzy boundaries <i>Pradumn Kumar Pandey, T. Ramalingeswara Rao and Satyajeet Maharana ...</i>	304
Precise Feature Selection and Case Study of Intrusion Detection in an Industrial Control System (ICS) Environment? <i>Terry Guo, Animesh Dahal, and Ambareen Siraj</i>	319
Entropy-based Approach for Parameter-free Attribute Clustering <i>Adison Khomprasert, Thanawin Rakthamanon, Kitsana Waiyamai</i>	333
Seq2SQL - Evaluating Different Deep Learning Architectures Using Word Embeddings <i>Kevin Stower and Dirk Krechel</i>	343
Sparse Multimodal Classification of EEG Signals from Rapid Serial Visual Presentation of Diagnostic Images <i>Valentina Sulimova, Sergey Bukhonov, Olga Krasotkina, Vadim Mottl, Annette Sterr, Kevin Wells, David Windridge</i>	355
Neural-Attention Multi-Instance Learning for Predicting User Demographics from Highly Noisy Tweets <i>Qing Chen, Mingxuan Sun, and Jian Zhang</i>	367
Explainable Artificial Intelligence for Match Analysis in Association Football <i>Bruno Marques and Dante Barone</i>	382
Using Machine Learning Algorithms to Predict the Likelihood of Recurrent Falls in Older Adults <i>Leeanne Lindsay, Sonya Coleman, Brian Taylor, Dermot Kerr, Anne Moorhead</i>	392
Tree Based Clustering On Large, High Dimensional Datasets? <i>Lee A. Carraher, Sayantan Dey, and Philip A. Wilsey</i>	397
Linear complexity algorithms for high dimensional SVM and regression problems with smart sparse regularization <i>Vadim Mottl, Olga Krasotkina, Valentina Sulimova, Alexey Morozov, Ilya Pugach, Alexander Tatarchuk</i>	412
Authors Index	431

Arabic Rule-based Named Entity Recognition System Using GATE

Hatem M. ElSherif¹[0000-0002-4402-9425], Khaled Mohammad Alomari²[0000-0001-6677-6301],
Ahmad Qasim AlHamad³ [0000-0002-7083-5375] and Khaled Shaalan⁴ [0000-0003-0823-8390]

¹Faculty of Engineering & IT, The British University in Dubai, Dubai, UAE
hatem.m.elsherif@gmail.com

²Faculty of IT, Abu Dhabi University, Abu Dhabi, UAE
khaled.alomari@adu.ac.ae

³Faculty of IT, Abu Dhabi University, Abu Dhabi, UAE
aqd14@yahoo.com

⁴Faculty of Engineering & IT, The British University in Dubai, Dubai, UAE
khaled.shaalan@buid.ac.ae

Abstract. As Arab users presence and participation continue to grow in the digital world, the electronic content comprising Arabic textual information significantly also increases due to the transition of traditional media channels to the online Arabic news and social networking. Such a rich source of information attracted many Arabic natural language processing researchers to tackle the named entity recognition problem, as it has a very influential role in the implementation of information processing systems. The proposed Arabic named entity recognition system adopts the rule-based approach using the General Architecture for Text Engineering (GATE) as a development environment. The system is applied on the ANERcorp and the results in terms of F-measure achieved 83% for Person NE, 89% for Organisation NE, and 92% for Location NE.

Keywords: Arabic Named Entity Recognition, Gate, Natural Language Processing

1 Introduction

Named Entity Recognition (NER) is a subtask of Information Extraction that was first introduced by the Sixth Message Understanding Conference (MUC-6) [1]. It is the task of both recognition and classification of definite named entities such as locations, organizations and persons names, numeric expressions including monetary amounts and dates and time expressions [2], [3].

It is widely used in the Natural Language Processing (NLP) applications to identify names in the context and classify them according to a specific predefined set of categories. Due to the continuous changes in the requirements, the Arabic Named Entity Recognition (ANER) task is considered in its early stages with all the respect to the many successful ANER research and systems. This study presents the challenges of

implementing ANER system adopting the rule-based approach using General Architecture for Text Engineering (GATE)¹. This study used the ANERcorp² which formatted according to the Conference on Computational Natural Language Learning (CoNLL) shared task framework CoNLL-2002³, i.e. following “Begin, Inside, Outside (BIO)” chunking representation format [4]. Illustrative examples extracted from the ANERcorp are shown in (Table 1); providing the Arabic word, English translation and the word tag using the CoNLL-2002 tagging format. The type classifies the words according to the CoNLL-2002 task into four types (persons/PERS, locations/LOC, organizations/ORG and miscellaneous/MISC) followed by the named entity chunking.

Table 1. Example of ANERcorp CoNLL tagging

Ara- bic	English	TAG	BIO	Type	Chunk
رجح	He favored	O	Outside	-	
جافيير	Javier	B-PERS	Beginning	Person	Person Name
سولانا	Solana	I-PERS	Inside	Person	
مجلس	Council	B-ORG	Beginning	Organisation	Organisation
امن	Security	I-ORG	Inside	Organisation	
القاهرة	Cairo	B-LOC	Beginning	Location	Location
دولر	USD	B-MISC	Beginning	Miscellanies	Miscellanies

1.1 Approaches for Named Entity Recognition

The various approaches proposed for Named Entity Recognition tasks can be grouped under three main categories, as follows: The Rule-Based (Handcrafted) Approach, Machine Learning Approach and Hybrid Approach [5]–[8]. Where several researchers conducted a comprehensive survey for NER approaches [2], [9]–[12].

1.2 Arabic Named Entity Recognition Issues and Challenges

A comprehensive review by Farghaly and Shaalan [13] provides insight on ANLP challenges facing researchers. On the other hand, the Arabic NER has more challenging issues like using the Latin characters as replacement to the Arabic script [14], Capitalization orthographic feature used in Roman script languages to recognize named entities such as proper names, acronyms, and abbreviations [15], the analysed Arabic word could have more than one interpretation that consists of root and prefixes, suffixes and clitics characters [16], Arabic language has a great level of transcriptional ambiguity leading to various ways of writing and translating foreign NE [17], Ambiguity due to different diacritic signs for the same word can represent different

¹ <http://gate.ac.uk/>

² <http://users.dsic.upv.es/~ybenajiba/>

³ <http://www.cnts.ua.ac.be/conll2002/ner/>

meanings [9], and the availability of free resource are limited while licensed resources are very costly [18].

2 ANER Systems and Related work

Many ANER systems were developed using various NER approaches and techniques. In this section, we provide an example for ANER systems.

Maloney and Niv [19], presented the TAGARAB system as one of the earliest efforts to implement an Arabic NER system. The system rely on rule-based approach using pattern matching engine consisting of two main modules, the morphological tokenizer and Name finder, to recognize Person, Organization, Location, Number and Time. The system used fourteen random texts from the AI-Hayat to evaluate the system performance. The results show better accuracy performance combining these two modules (NE finder and morphological tokenizer) than only using them individually.

ANERsys system used the ANERcorp. 125,000 tokens were used for training and 25,000 tokens for testing, the system used the Maximum Entropy approach is to identify a named entity using the two words before and after a token; excluding the stop words. The training set was weighted using YASMET. CONLL framework script is used for evaluation with and without the use of gazetteers [4]. The second version ANERsys2 adopted a different architecture using Part-Of-Speech tags and Base Phrase Chunks improving the performance by 10 points [16]. Another performance improvement achieved by replacing the Maximum Entropy probabilistic model to Conditional Random Fields [20].

Shaalán and Raza [17] introduced Arabic NER called Named Entity Recognition for Arabic (NERA) extracting 10 types of Arabic named entities adopting the rule-based approach using gazetteers, grammar rules, and a filtering mechanism to extract the NE whitelists first and using the grammar rules with the support of gazetteers to extract the NE finally using Blacklist to filter invalid entities. The system results achieved F-measure of 87.7% for person, 85.9% for locations, and 83.15% for organizations [21], [22].

Oudah and Shaalan [6] developed a new hybrid system from NERA that can recognize 11 Arabic named entity types. The machine learning (ML) approach is used to enhance the accuracy of recognising Named Entity. The rule-base module developed using General Architecture for Text Engineering (GATE) and MADA for morphological analysis and POS tagging. The system results achieved (F-measures 94.4% for Person, 90.1% for Location, and 88.2% for Organisation) tested on the ANERcorp corpus.

Several researchers used the GATE environment for developing ANER systems such as [5], [6], [23], [24]. The rule-based system still achieves significant results and it plays important roles when integrated with machine learning approach to for a unified hybrid system [6].

3 Methodology

Building a rule-based NER system from scratch requires five main stages. The first stage is the data collection for providing the needed language resources. The second is to identify and select the development environment and processing resources. The third stage is concerned with processing the linguistic resources. The fourth stage is to develop the recognition rules, and customize the system components. The last stage is to test and evaluate the system's results.

3.1 Linguistic Resources

The linguistic resources are a critical component for implementing the NER system. The collection process of such resources requires reviewing literatures and other related Arabic NER systems, such as [18], [21], [22], [25], in addition to wide internet search for free Arabic language resources. Different methods and techniques were used to extract and refine the data sources using automated and manual approaches. The linguistic resources consist of two main resources the corpus and gazetteers. This study used freely available corpora due to the lack of access to commercial corpora. Zaghouani [18] conducted an intensive survey for freely available Arabic corpora including six Named Entities Corpora freely available.

A Corpus is a very large set of text which is more considered as a methodology used to teach and study aspects of language [26]. Corpus is often used in Computational Linguistics research and it is essential for NE researchers to use corpus in developing, training and evaluating NER systems. A monolingual corpus contains single language texts, while parallel or bilingual corpus consists of two natural languages the multilingual corpus may include more languages [9]. Corpus used in the training and evaluation process should be annotated. Corpora are manually annotated by individuals or team of researchers [27]. Manually annotated corpora commonly named as gold standard corpora. Another approach is the use of rich structured electronic resources (i.e. Wikipedia, Arabic WordNet⁴, YAGO⁵) to automatically generate annotated corpora commonly named as silver-standard corpora [28]. Shaalan [9], pointed to the importance of the balance of corpora particularly in named entities distribution when used for NER in order to be reliable. Meanwhile Al-Thubaity et al. [29], proposed a framework for evaluating standalone Arabic Corpora to assist researcher in testing their corpora (usability, functionality, and performance). Corpus usually collected from Arabic newspaper archives [30], or other digital electronic resources such as RSS Feed [31] which focused on Modern Standard Arabic. With the growing use of social media networks and mobile devices by the Arabic user's researchers start to build corpus from SMS/Chat by translating Arabizi to Arabic [32], and from different dialects to other languages such as from Algerian Arabic to French [33].

⁴ <http://globalwordnet.org/arabic-wordnet/>

⁵ <http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/>

ANERcorp. The main corpora used in training and evaluating this system is the ANERcorp and additional corpus used for development and testing such as Wik-iFANEGold⁶, NewsFANEGold⁷ which developed by Alotaibi & Lee [34]. Benajiba et al. [4], developed Arabic manually annotated corpus (ANERcorp) for the NER task according to the CONLL 2002 conference framework in their work to implement the ANERsys (Arabic Named Entity Recognition System). The corpus consists of 316 news articles from different topics and newspapers, which includes 150,286 tokens were 11% of tokens are Named Entities distributed as (39% Persons, 30.4% location, 20.6% organization and 10% miscellaneous). The ANERcorp has some spelling and annotation errors as notes by researchers [35].

Example for annotation error is “B-PERS” instead of “B-LOC”

3.2 Gazetteers

The study refined and used the ANERgazet⁸ which is a manually built set of gazetteers collected from several internet resources. It is consisted of three gazetteers (Location Gazetteer:1,950 , Person Gazetteer: 1,920, Organizations Gazetteer:262) [4].

In addition, location gazetteers are collected from several resources, such as the WikiLocation⁹ a Multilanguage structured data set collected from Wikipedia database dump, as well as the GeoNames Gazetteer¹⁰.

Example for the GeoNames (alternateNames) file format:

4471603	292231	ar	بافجيرة
1603445	347497	ar	طنطا

WikiFANEGazet¹¹ is a gazetteer that includes 683555 named entities collected from the Wikipedia which consists of 8 main classes (Person, Organization, Location, Geo-Political, Facility, Vehicle, Weapon, and Product) and 55 subclasses [36].

Example of the WikiFANEGazet gazetteer's entities:

```
<PER_Artist>هولنغ /PER_Artist>
<PER_Politician>جوليا دونا /PER_Politician>
<ORG_Educational>جامعة العلوم اسي لة لة لة /ORG_Educational>
<GPE_Nation>كلب سنة ول هرسك /GPE_Nation>
<ORG_Commercial>شركة مصر للطيران /ORG_Commercial>
<FAC_Building-Grounds>المستاد القاهري دولي /FAC_Building-Grounds>
```

JRC-Names¹² is a multilingual named entity gazetteer that consists of 205,000 named entities for Person and Organisation name entities collected over seven years

⁶ <http://www.cs.bham.ac.uk/~fsa081/>

⁷ <http://www.cs.bham.ac.uk/~fsa081/>

⁸ <http://users.dsic.upv.es/~ybenajiba/>

⁹ <http://files.bendodson.com/wikilocation-dumps/index.html>

¹⁰ <http://www.geonames.org/>

¹¹ <https://sourceforge.net/projects/arabic-named-entity-gazetteer/?source=navbar>

¹² <http://optima.jrc.it/data/entities.gzip>

from multilingual news and Wikipedia mining including a number of spelling variants [37]. By extracting, refining and solving some encoding problems we end up with 19464 Arabic Persons and 773 Organization named entities.

Example of the JRC-Names gazetteer's entities:

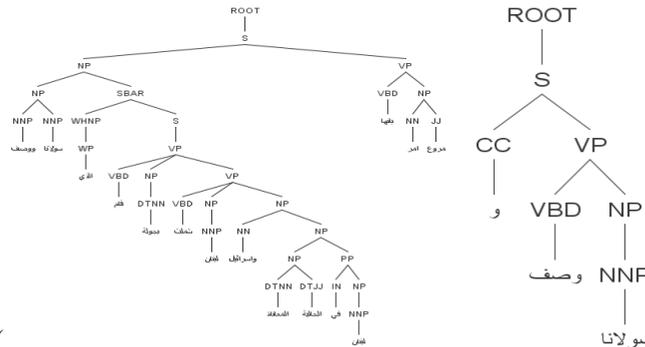
185274	P	ar	عمر بن قيس
185311	P	u	حالين
100044	O	u	ريال بيل دال لويدي

3.3 Arabic Tokenizer

This system uses GATE Arabic language plugin tokenizer¹³, which is customized from the English tokenizer, it is used to split text to the simplest tokens such as punctuation, numbers, and words aiming to increase efficiency and support other process such as grammar rules. The default settings don't recognize the Arabic diacritic and require modification to recognize it as NON_SPACING_MARK [24].

3.4 Part-Of-Speech Tagging

The Part-Of-Speech Tagger reads text and assigns parts of speech for each token. For example, noun -> NN , verb -> VB , adjective-> JJ and noun plural -> NNS). As



shown in (

) the word “Lebanon” in Arabic “لبنان” will be tagged as (NNP: noun proper singular). The proper POS tagging requires clitics segmentation. The lack of this process causes the tool to tag words including clitics as noun. For example “وصفنا” in English “and described” tagged as (NNP), while it should be separated into the proclitic “و” (“And”) and tagged as (CC), and “وصفنا” (“described”) and tagged as (VBD), see Fig. 1.

Monroe et al., argue that although segmentation of clitics improves accuracy on Arabic NLP tasks, it is limited to formal Modern Standard Arabic and has poor performance in dialect Arabic text proposing a single accurate clitic segmentation model for both MSA and informal Arabic [38]. On the other hand, Mohamed & Kübler claims that segmentation is not required for POS tagging and providing higher accuracy proposing a POS tagging approach without any word segmentation described in [39].

¹³ https://gate.ac.uk/gate/doc/plugins.html#Lang_Arabic

Stanford Part-Of-Speech (POS) Tagger. developed originally for English at Stanford University based on the maximum-entropy model [40]. The enhanced version improved the performance and supports other languages (Arabic, Chinese, French, Spanish, and German) [41]. The latest version (v3.5.2¹⁴) includes Arabic model trained on parts 1-3 of the Penn Arabic Treebank (ATB) using the pre-processing described in [42]. Stanford POS tagger requires tokenization and segmentation; although Stanford Word Segmenter¹⁵ supports the Arabic language, unfortunately GATE currently does not support this tool as part of Stanford CoreNLP plugin [43].

Table 2. Stanford POS Tagger example

Arabic	ووصف سولانا التي جرت في لبنان وإسرائيل المصيبة في لبنان بلها أمر مروع
English	Solana who has toured Lebanon and Israel described the current suffering in Lebanon that it is horrible

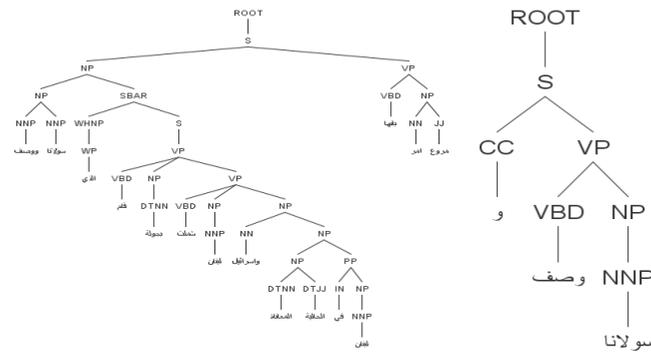


Fig. 1. Stanford POS Tagger\ Parser example

3.5 System development environment

This study used GATE development environment which is a free open-source NLP toolkit with multilingual support including Arabic that provides an infrastructure for developing and deploying software components that process human language [44]. It is widely used by NLP researchers in many communities and used in the development of several Arabic NER systems [9], [45], [46]. A Nearly-New Information Extraction System (ANNIE) included within GATE as a package of reusable processing resources for common natural language processing tasks (tokenizer, sentence splitter, POS tagger, gazetteer, Name matcher), ANNIE depend on finite state algorithms and the JAPE language [47].

¹⁴ <http://nlp.stanford.edu/software/stanford-postagger-full-2015-04-20.zip>

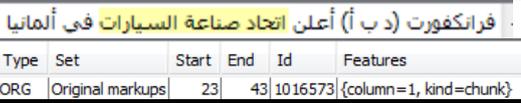
¹⁵ <http://nlp.stanford.edu/software/segmenter.shtml>

3.6 Processing linguistic resources and System Development

GATE used to build the system using the Lang_Arabic plugin components (Gazetteer Collector, Gazetteer, Tokenizer, Sentence splitter and JAPE Transducer) and the Stanford POS Tagger plugin. Corpora divided equally into three parts. The first part is used for the initial development process to implement the grammar rules using JAPE and to test the quality of the gazetteers. The second is the corpora used to test, refine, and implement additional rules and gazetteers to cover other domains such as sports, in addition to tune the rules for optimizing priorities and effectiveness. The last subset used as Evaluation corpora (held-out data) used to evaluate the system.

GATE currently supporting the CoNLL-2002 format which keeps the original marks for named entities words as chunk as example "فلاح اصنع السيارات" originally separated and defined as shown in (Table 3) and will be one chunk annotated as type (ORG).

Table 3. Example for ANERcorp annotation by GATE

Original CoNLL-2002 format	GATE Annotation												
B-ORG تحد	 <table border="1"> <thead> <tr> <th>Type</th> <th>Set</th> <th>Start</th> <th>End</th> <th>Id</th> <th>Features</th> </tr> </thead> <tbody> <tr> <td>ORG</td> <td>Original markups</td> <td>23</td> <td>43</td> <td>1016573</td> <td>{column=1, kind=chunk}</td> </tr> </tbody> </table>	Type	Set	Start	End	Id	Features	ORG	Original markups	23	43	1016573	{column=1, kind=chunk}
Type		Set	Start	End	Id	Features							
ORG		Original markups	23	43	1016573	{column=1, kind=chunk}							
I-ORG اصنع													
O-ORG السيارات													

Gazetteers building, cleansing, and refining require a massive work using several approaches. GATE default Arabic Gazetteer Collector system is customized to automate the extraction of 50 subset named entity Gazetteers from the WikiFANEGold Corpora such as (Sports, Politician, Artist, Educational) incorporated with the WikiFANEGazet¹⁶ [36], and other collected gazetteers. The system used also to extract the (LOC, ORG, PERS, and MISC) from the two ANERcorp corpora used in the development included only in the last stage of testing. In addition to several automated processes for cleansing and refining the gazetteers to remove invalid characters such as “:”, remove extra spaces, delete duplication and trimming leading and trailing whitespaces, the manual process was important to review and group gazetteers such as JRC-Names. The location gazetteers collected from GeoNames¹⁷ Gazetteer (alternate-Names) extracting over 100,000 Arabic location name, which was refined and incorporated with the WikiFANEGazet locations gazetteer to include about 125,000 location, in addition to counters short and long names, other types were collected (landmarks, mountains, water bodies, rivers, continents, and others).

The persons gazetteers set includes different types (Artists, Business, Engineer, Police, Politician, Religious, Scientist, Lawyer and other) collected and refined including the male, female and surname gazetteers from the GATE Arabic gazetteers.

Supporting set of gazetteers created to provide the common words comes before and after the named entities such as titles, nationalities, organizations types and directions. Other gazetteers such as date, time, currency, percent and numbers to support the

¹⁶ <https://sourceforge.net/projects/arabic-named-entity-gazetteer/?source=navbar>

¹⁷ <http://www.geonames.org/>

monetary and numbers expressions rules, in addition to conjunction and stopwords, which collected from several resources.

4 Rule-based Named Entity Extractors

The extraction of named entities approach has two main stages. In the first stage, we extract the identified named entities from gazetteers (whitelist). The second indirectly extracts name entities, through POS features, grammar rules, and implemented grammatical rules using the JAPE Transducer processing resource.

4.1 Locations Named Entities Recognition rules

The extraction of Locations Named Entities implemented in JAPE by identifying the matched patterns in the Left Hand Side (LHS) part of the rule and defining the annotation in the Right Hand Side (RHS) of the rule. The implemented rules can be classified into three types:

Direct rules. using gazetteers whitelist and JAPE macros to annotate tokens and add features for (Country, City, Continent, Waterbody, River), Additional feature (Landmarks and Religious Locations) were annotated (LOCF) as it has not been represented in the ANERcorp.

Indirect rules. which matching Location NE by using keywords gazetteers in a simple form as pre-location keywords (“محافظة” – “City of”) or (“دولة”-“Country of”) and directions, in order to annotate the recognised Locations Named Entities from the whitelist gazetteers and adding attaching features. Example of the location NE annotated by this rule is “جزيرة جاوا” – “Java Island” recognised using the “جزيرة” - “Island” keyword, this named entity cannot be identified by the gazetteers lookup process as the right Arabic spelling is “جاوة”.

Context Relational rules. The contextual relations are a very powerful technique to recognise the NE, the same keyword approach can be integrated with this method as example "مدينتي بيصلك ولحرمك", the relations can be resolved using specific keywords such as “مدينتي” - “the two cites” indicating that the next two following tokens most probable will be the names of two cites.

This rule "LocRule_DirRelation1" will be triggered by this text “لأراضي الجزرية” using keyword “لأراضي” and direction part “لأجزرية” to recognize “ل” as affix + “جزرية” as a location Named Entity.

4.2 Organisations Extractor

Using direct rule to annotate the organisation NE whitelist according to the corpora (ORG), and adding features by researcher for types (Airport, Educational, Commercial, Entertainment, Medical-Science, Media, Non-Government, Religious and Sports). Direct rule is used relying on the quality of the gazetteers. In particular, the recognising sports organisation NE demonstrate its very difficult characteristic to recognise by the direct rule as majority of the sports clubs has cites names and nationality as part of the

NE. Example of the results for rule “OrgSports4” is ” **عبدنجلستربالبريغالي** ” for “Manchester Portuguese player” which uses pre-organization words, Locations gazetteers, Nationality gazetteers to recognise the candidate sports team and utilize the POS tagging to reduce the matching scope for (NNP, “proper noun”) and (NN, “noun”).

4.3 Persons Extractor

This component annotates the Persons NE according to the corpora (PERS) and includes additional features by researcher for persons named entity of the types (Artist, Business, Engineer, Police, Politician, Religious, Scientist, Lawyers, and others). Additional rules implemented to chunk and collect person name sets using gazetteers (e.g., Titles, pre-persons keywords) and POS tags varies techniques. Following some of the implemented rules for recognising the Persons NE using job title and Lexical information using POS tagging. As example, the “**الفنان**” will trigger rule (PERSaffix3) as the first token “**الفنان**” is found in the person pre-words, followed by token tagged as “DTJJ”. Stanford POS tagger follows the Pann Treebank tagset meaning “DT, Determiner” and “JJ, Adjective” matching the second part “**عبدنجلستري**” and looking for a minimum one to the maximum four following tokens with the “noun” types.

Another rule uses both person title and organisation pre-keywords to recognize the person NE the text triggered by this rule “**بن دنجوشولك**” this text formatted as (PRE-PERS + PRE-ORG + B-PERS + I-PERS).

5 Evaluation

The Confusion matrix shown in (**Table 5**) mathematically defines the way of measure the system’s performance “goodness” against a human-annotated “gold standard”. It is used to extract the evaluation metrics (Precision, Recall, and F-Measure) which is widely used in Information Retrieval systems performance Evaluation and became a standard evaluation method [48].

Table 4. Confusion matrix

		Prediction	
		+	-
Original Marks	+	true positive	false negative
	-	false positive	true negative

The following formulas define the calculation of evaluation metrics:

$$Precision = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \quad (1)$$

$$Recall = \frac{\text{true Positive}}{\text{true positive} + \text{false negative}} \quad (2)$$

$$F - \text{measure} = \frac{2 \times \text{true positive}}{(2 \times \text{true positive}) + \text{false positive} + \text{false negative}} \quad (3)$$

The system performance results evaluated using GATE AnnotationDiff tool as shown in (Fig. 2). This tool provides comparison of original marks and the generated annotations, the evaluation results provide statistics of the system performance and the standard measures Recall, Precision and F-Measure.

Start	End	Key	Features	=?	Start	End	Response
49489	49500	شون مكرمهاك	(column=1, kind=chunk)	=	49489	49500	شون مكرمهاك
269877	269892	كريستوفر كولميس	(column=1, kind=chunk)	=	269877	269892	كريستوفر كولميس
167917	167922	سنگري	(column=1, kind=chunk)	=	167917	167922	سنگري
253443	253447	دالي	(column=1, kind=chunk)	=	253443	253447	دالي
269912	269924	فاسكو ديچاما	(column=1, kind=chunk)	=	269912	269924	فاسكو ديچاما
205317	205328	بيتر شيلتون	(column=1, kind=chunk)	=	205317	205328	بيتر شيلتون
163002	163014	سعيد الهاجري	(column=1, kind=chunk)	=	163002	163014	سعيد الهاجري
119884	119898	كوندوليزا رايس	(column=1, kind=chunk)	=	119884	119898	كوندوليزا رايس
103517	103528	عفير بيريس	(column=1, kind=chunk)	=	103517	103528	عفير بيريس
54948	54961	سيفاسميان الوي	(column=1, kind=chunk)	=	54948	54961	سيفاسميان الوي
167937	167943	...	(column=1, kind=chunk)	=	167937	167943	...

Correct:	1354	Recall	Precision	F-measure	
Partially correct:	94	Strict:	0.93	0.71	0.80
Missing:	15	Lenient:	0.99	0.76	0.86
False positives:	454	Average:	0.96	0.74	0.83

Fig. 2. GATE AnnotationDiff tool snapshot for the Arabic Person NE.

The system performance results generated by the AnnotationDiff for the Persons NE is shown in (Table 5). It calculates the measures according to the Confusion matrix approach with the partially correct (lenient) NE system achieved (0.86 F-Measure) and without the partially correct (strict) NE system achieved (0.80 F-Measure) where the average result between strict result and lenient achieved (0.83 F-Measure).

Table 5. Persons NE Results

correct	1354	recall	precision	F-Measure	
partially correct	94	strict	0.93	0.71	0.80
missing	15	lenient	0.99	0.76	0.86
false positives	454	average	0.96	0.74	0.83

The system performance results generated by the AnnotationDiff for the Location NE is shown in (Table 6). It calculates the measures according to the Confusion matrix approach with the partially correct (lenient) NE system achieved (0.93 F-Measure) and without the partially correct (strict) NE system achieved (0.90 F-Measure) where the average result between strict result and lenient achieved (0.92 F-Measure).

Table 6. Location NE Results

correct	1893	recall	precision	F-Measure	
partially correct	78	strict	0.92	0.87	0.90
missing	84	lenient	0.96	0.91	0.93
false positives	193	average	0.94	0.89	0.92

The system performance results generated by the AnnotationDiff for the organisation NE is shown in (Table 7). It calculates the measures according to the Confusion matrix approach with the partially correct (lenient) NE system achieved (0.93 F-Measure) and without the partially correct (strict) NE system achieved (0.86 F-Measure) where the average result between strict result and lenient achieved (0.89 F-Measure).

Table 7. Organization NE Results

correct	1071		recall	precision	F-Measure
partially correct	79	strict	0.88	0.85	0.86
missing	74	lenient	0.94	0.91	0.93
false positives	112	average	0.91	0.88	0.89

Table 8 provides a summary NE system result in three situations (Strict, Lenient, and Average) for persons NE, Location NE, and Organisation NE.

Table 8. NE System results Summary

		Recall	Precision	F-measure
Strict	Persons	0.93	0.71	0.80
	Location	0.92	0.87	0.90
	Organisation	0.88	0.85	0.86
Lenient	Persons	0.99	0.76	0.86
	Location	0.96	0.91	0.93
	Organisation	0.94	0.91	0.93
Average	Persons	0.96	0.74	0.83
	Location	0.94	0.89	0.92
	Organisation	0.91	0.88	0.89

Through comparing the **Average** result with related work findings, this study achieved a better result in Location NE (92% F-measure) and Organisation NE (89% F-measure) where the other work [21], [22] achieved result in Location NE (85.9% F-measure) and Organisation NE (83.1% F-measure) and [6] paper achieved result in Location NE (90.1% F-measure) and Organisation NE (88.2% F-measure) but both studies achieved better result in the Persons NE (87.7% F-measure) and (94.4% F-measure) where this study achieved (83% F-measure).

6 Conclusion

The Named Entity Recognition system using the rule-based approach requires intensive efforts to provide high performance and considered more successful when focused on a specific domain. This system biggest limitation is the lack of segmentation and the need for filter process to resolve and remove the incorrect annotation caused by the

different NE recognition process specially the conflict between the location and organisation NE due to embeddings. One of the important facts learned from implementing this system that although most of the recent researches tend to use the online large knowledge systems available on the internet to increase the gazetteers size and scope the price of such approach is very expensive as it reduces the system precision and requires intensive efforts to clean and manage. Building ANER rule-based systems are more of art than science; it requires experiences and skills to develop and judge the right techniques which needs deep knowledge of the Arabic language.

7 Bibliography

1. R. Grishman and B. Sundheim, "Design of the MUC-6 evaluation," in *Proceedings of a workshop on held at Vienna, Virginia: May 6-8, 1996*, 1996, pp. 413–422.
2. D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Linguisticae Investig.*, vol. 30, no. 1, pp. 3–26, 2007.
3. R. Gaizauskas and Y. Wilks, *Information extraction: beyond document retrieval*, vol. 54, no. 1. 1998.
4. Y. Benajiba, P. Rosso, J. BeneditRuiz, and M. Bened, "ANERsys: an Arabic named entity recognition system based on maximum entropy," *Gelbukh, A. CICLing 2007. LNCS*, pp. 143–153, 2007.
5. S. Abdallah, K. Shaalan, and M. Shoaib, "Integrating Rule-Based System with Classification for Arabic Named Entity Recognition," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7182, no. PART 2, pp. 311–322, 2012.
6. M. Oudah and K. Shaalan, "A Pipeline Arabic Named Entity Recognition using a Hybrid Approach," *Coling*, vol. 2, no. December 2012, pp. 2159–2176, 2012.
7. O. Zayed, S. El-Beltagy, and O. Haggag, "A Novel Approach for Detecting Arabic Persons' Names using Limited Resources," *Complement. Proc. 14th Int. Conf. Intell. Text Process. Comput. Linguist. CICLing 2013 (Accepted Present.)*, vol. 70, pp. 81–93, 2013.
8. B. Bengfort, "A Survey of Stochastic and Gazetteer Based Approaches for Named Entity Recognition," no. 7, pp. 1–9, 2014.
9. K. Shaalan, "A Survey of Arabic Named Entity Recognition and Classification," *Comput. Linguist.*, vol. 40, no. 2, pp. 469–510, Jun. 2014.
10. A. Satpreet, "Named Entity Recognition Literature Survey," pp. 1–47, 2012.
11. V. Gupta, "A survey of Named Entity Recognition in English and other Indian Languages," *IJCSI Int. J. Comput. Sci. Issues*, vol. 7, no. 6, pp. 239–245, 2010.
12. H. J. Mahanta, "A STUDY ON THE APPROACHES OF DEVELOPING A NAMED ENTITY RECOGNITION TOOL," *IJRET Int. J. Res. Eng. Technol.*, pp. 2319–2322, 2013.
13. A. Farghaly and K. Shaalan, "Arabic Natural Language Processing : Challenges and Solutions," *ACM Trans. Asian Lang. Inf. Process.*, vol. 8, no. 4, pp. 1–22, 2009.
14. M. Maamouri, A. Bies, S. Kulick, and M. Ciul, "Developing an Egyptian Arabic Treebank : Impact of Dialectal Morphology on Annotation and Tool Development," in *Proc. of LREC*, 2010, pp. 2348–2354.
15. B. Farber, D. Freitag, N. Habash, and O. Rambow, "Improving NER in Arabic Using a Morphological Tagger," in *Proceedings of the Sixth International Conference on Language*

- Resources and Evaluation (LREC'08)*, 2008, pp. 2509–2514.
16. Y. Benajiba and P. Rosso, “ANERSys 2.0: Conquering the NER Task for the Arabic Language by Combining the Maximum Entropy with POS-tag Information.,” in *3rd Indian International Conference on Artificial Intelligence (IICAI-07)*, 2007, pp. 1814–1823.
 17. K. Shaalan and H. Raza, “Person Name Entity Recognition for Arabic,” *Comput. Linguist.*, no. June, pp. 17–24, 2007.
 18. W. Zaghouni, “Critical Survey of the Freely Available Arabic Corpora Current situation of the freely available,” in *Proceedings of the Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools Workshop Programme*, 2014, p. 1.
 19. J. Maloney and M. Niv, “Tagarab: a fast, accurate arabic name recognizer using high-precision morphological analysis,” *Proc. Work. Comput. Approaches to Semit. Lang.*, pp. 8–15, 1998.
 20. Y. Benajiba and P. Rosso, “Arabic Named Entity Recognition using Conditional Random Fields,” in *Proc. of Workshop on HLT & NLP within the Arabic World*, 2008, pp. 143–153.
 21. K. Shaalan and H. Raza, “Arabic named entity recognition from diverse text types,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5221 LNAI, pp. 440–451, 2008.
 22. K. Shaalan and H. Raza, “NERA: Named entity recognition for Arabic,” *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, no. 8, pp. 1652–1663, 2009.
 23. a Elsebai, F. Meziane, and F. Belkredim, “A Rule Based Persons Names Arabic Extraction System,” *Commun. IBIMA*, vol. 11, no. August, pp. 53–59, 2009.
 24. A. Alfaries, M. Albahlal, M. Almazrua, and A. Almazrua, “A Rule Based Annotation system to extract Tajweed Rules from Quran,” in *Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences*, 2013.
 25. K. Shaalan and M. Oudah, “A hybrid approach to Arabic named entity recognition,” *J. Inf. Sci.*, vol. 40, no. 1, pp. 67–87, 2013.
 26. L. Al-Sulaiti and E. S. Atwell, “The design of a corpus of Contemporary Arabic,” *Int. J. Corpus Linguist.*, vol. 11, no. 2, pp. 135–171, 2006.
 27. D. Farwell *et al.*, “Interlingual annotation of multilingual text corpora,” in *Proceedings of The North American Chapter of the Association for Computational Linguistics Workshop on Frontiers in Corpus Annotation*, 2004, pp. 55–62.
 28. J. Nothman, N. Ringland, W. Radford, T. Murphy, and J. R. Curran, “Learning multilingual named entity recognition from Wikipedia,” *Artif. Intell.*, vol. 194, pp. 151–175, 2013.
 29. A. Al-Thubaity, H. Al-Khalifa, R. Alqifari, and M. Almazrua, “Proposed Framework for the Evaluation of Standalone Corpora Processing Systems: An Application to Arabic Corpora,” *Sci. World J.*, vol. 2014, no. August 2015, pp. 1–10, 2014.
 30. A. M. Saif and M. J. A. Aziz, “An Automatic Collocation Extraction from Arabic Corpus,” *J. Comput. Sci.*, vol. 7, no. 1, pp. 6–11, 2011.
 31. S. Khoja, “An RSS Feed Analysis Application and Corpus Builder,” *Interface J. Educ. Community Values*, vol. 9, no. 3, pp. 115–118, 2009.
 32. A. Bies *et al.*, “Transliteration of Arabizi into Arabic Orthography: Developing a Parallel Annotated Arabizi-Arabic Script SMS / Chat Corpus,” *ANLP 2014*, vol. 93, 2014.
 33. R. Cotterell, A. Renduchintala, N. Saphra, and C. Callison-burch, “An Algerian Arabic-French Code-Switched Corpus,” in *LREC-2014 Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools*, 2014, p. 34.
 34. F. Alotaibi and M. Lee, “A Hybrid Approach to Features Representation for Fine-grained Arabic

- Named Entity Recognition,” in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 23–29.
35. M. M. Oudah, “Integrating Rule-based Approach and Machine learning Approach for Arabic Named Entity Recognition,” The British University in Dubai, 2012.
 36. F. Alotaibi and M. Lee, “Automatically Developing a Fine-grained Arabic Named Entity Corpus and Gazetteer by utilizing Wikipedia,” in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013, no. October, pp. 392–400.
 37. R. Steinberger, B. Pouliquen, and M. Kabadjov, “JRC-NAMES: A Freely Available, Highly Multilingual Named Entity Resource.,” *arXiv Prepr. arXiv*, no. September, pp. 104–110, 2013.
 38. W. Monroe, S. Green, and C. D. Manning, “Word Segmentation of Informal Arabic with Domain Adaptation,” *Proc. 52nd Annu. Meet. Assoc. Comput. Linguist. (Volume 2 Short Pap.)*, pp. 206–211, 2014.
 39. E. Mohamed and S. Kübler, “Is Arabic Part of Speech Tagging Feasible Without Word Segmentation?,” in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010, no. 1, pp. 705–708.
 40. K. Toutanova and C. D. Manning, “Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger,” in *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 2000, pp. 63–70.
 41. K. Toutanova, D. Klein, and C. D. Manning, “Feature-rich part-of-speech tagging with a cyclic dependency network,” *Proc. 2003 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Vol. 1 (NAACL '03)*, pp. 252–259, 2003.
 42. S. Green, M. Galley, and C. D. Manning, “Improved Models of Distortion Cost for Statistical Machine Translation,” *Hum. Lang. Technol. 2010 Annu. Conf. North Am. Chapter Assoc. Comput. Linguist.*, vol. 5, pp. 867–875, 2010.
 43. GATE.ac.uk, “GATE.ac.uk - Stanford CoreNLP,” 2015. .
 44. H. Cunningham *et al.*, “Developing Language Processing Components with GATE Version 8,” 2014.
 45. D. Maynard *et al.*, “A Survey of Uses of GATE,” 2000.
 46. S. Zaidi, M. T. Laskri, and A. Abdelali, “Arabic collocations extraction using gate,” in *2010 International Conference on Machine and Web Intelligence, ICMWI 2010 - Proceedings*, 2010, no. August 2015, pp. 473–475.
 47. K. Bontcheva, D. Maynard, V. Tablan, and H. Cunningham, “GATE: A Unicode-based infrastructure supporting multilingual information extraction,” *Proc. Work. Inf. Extr. Slavon. other Cent. East. Eur. Lang. (IESL03), Borovets*, 2003.
 48. A. De Sitter, T. Calders, and W. Daelemans, “A Formal Framework for Evaluation of Information Extraction,” 2004.

An Unsupervised Statistical Moving Shadow Detection Method for Video Analysis

Hang Shi and Chengjun Liu *

New Jersey Institute of Technology
Department of Computer Science
Newark, NJ 07102 USA

Abstract. In video analysis, people are interested in the active foreground objects. But the cast shadows are always extracted together with the foreground objects. This can have a negative impact on some video analysis results. Many methods based on deep learning are proposed to detect and remove the shadows. However, these methods require huge amount of labeled training data, which is hard to obtain. In this paper, an unsupervised statistical moving shadow detection method is proposed to detect and remove the cast shadows from the foreground detection result, which does not require any labeled training data. First, some candidate shadow pixels are detected using the luminance property that the shadow is darker than the background. Second, the candidate shadow pixels are filtered by an innovative heuristic shadow region detection method, which can define some shadow regions based on the property that the shadow pixels should be continuous. Third, a statistical shadow model is learnt online with the refined shadow pixels and is used to classify the foreground pixels into the object class and the shadow class. Experimental results using the New Jersey Department of Transportation (NJDOT) traffic video sequences show the feasibility of the proposed method.

Keywords: Video analysis, unsupervised shadow detection, statistical shadow model.

1 Introduction

In recent years, many algorithms have been proposed to detect foreground objects [1], [2], [3], [4]. However, one disadvantage of these foreground detection methods is that the cast shadows of the foreground objects are always detected together with the objects [5]. We can see from Fig. 1, the cast shadows are detected as foreground as they are moving together with the vehicles. These cast shadows always defect the video analysis performance. In order to get over this advantage, we present an unsupervised statistical moving shadow detection method to detect and remove the shadows from the foreground.

In our shadow detection method, we first use the foreground detection method proposed in [4] to get a foreground mask with shadows. Then we apply our novel shadow detection method to detect and remove the shadows from this foreground mask.

* Hang Shi and Chengjun Liu are with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, 07102 USA e-mail: hs328@njit.edu, cliu@njit.edu



Fig. 1. (a) A video frame from an NJDOT traffic video. (b) The background derived using the method in [4]. (c) The foreground (with shadow) detected using the method in [4].

There are three major contributions of our novel cast shadow detection method, that are summarized below. First, we present a set of new illuminating criteria to detect the candidate shadow pixels in the HSV color space. We use the HSV color space for shadow detection due to its property of separating the chromaticity from intensity [6]. Second, we present an innovative heuristic shadow region detection method to cut each foreground area into a shadow region and an object region. As we know, the cast shadow of an object should form a continuous region. The shadow pixels cannot be in a non-shadow region. Based on this property, the candidate shadow pixels that fall in the shadow regions are treated as more reliable shadow pixels. We can refine the candidate shadow pixels by filtering them using the shadow regions. Third, we present a statistical shadow modeling and classification method, which uses a single Gaussian distribution to model the shadow, and classifies the foreground pixels into the object class or the shadow class. In our statistical shadow modeling method, the refined shadow pixels are used to estimate the shadow probability density function. Therefore, our statistical shadow detection method does not need any labeled training data, and our shadow model can be updated online to adapt to changes in the environment.

We implement experiments using the New Jersey Department of Transportation (NJDOT) traffic video sequences to show the feasibility of the proposed method.

2 Related Work

Many methods have been published for moving cast shadow detection [5], [7], [8]. As color often provides useful information for shadow detection, some methods apply color information to detect shadows [6], [9]. Many shadow detection methods assume that the shadow areas are darker in intensity but relatively invariant in chromaticity [7], [10]. The color spaces that separate chromaticity from intensity are thus often used for shadow detection. Some example such color spaces are the HSV color space [6], the c1c2c3 color space [11], and the YUV color space [9]. Some popular methods apply a set of chromatic criteria by assuming that the cast shadows have similar hue to the background, but a lower saturation and a lower value than the background [7], [10].

Statistical shadow modeling is applied for shadow detection as well [12]. The major assumption of these methods is that the light source is pure white and the attenuation of

the illumination is linear. Generally speaking, these statistical shadow modeling methods are able to predict color changes of the shadow pixels better than the color based methods, but the shadow detection accuracy in outdoor scenes tends to deteriorate.

There are methods that use the shape, size, and orientation information for shadow detection [13]. These methods are designed to deal with some objects that have specific shapes. The advantage of these methods is that they do not need to estimate the background color of the shadow, but the disadvantage is that they have difficulty in dealing with multiple types of objects in complex scenes.

There are also methods that utilize texture for shadow detection, such as classifying a region into the shadow region or the object region based on the texture correlation between the foreground and the background [14], [15], [16]. These methods extract the texture information in different sizes of the regions. The advantage of these methods is that they are more robust to illumination changes than the color based methods, but the disadvantage is that the computation efficiency of matching the texture features is low.

Recently, there are methods that use machine learning techniques for shadow detection: a paired region based shadow detection algorithm is presented in [15], a kernel least-squares SVM method for separating shadow and non-shadow regions is proposed in [17], and some shadow detection algorithms using the deep neural network are also presented in [18], [19], [20]. Most of these learning based shadow detection methods need huge amount of labeled training data, which is hard to obtain. Additionally, the computational complexity of these methods tends to be very high, so that they cannot perform real time analysis. More ever, the performance of these methods are always relied on the training data, the generality in different data sets is not good.

3 A Novel Statistical Moving Shadow Detection Method

In this paper, we present a novel unsupervised statistical moving shadow detection method, which includes three hierarchical steps. First, we use a set of new illuminating criteria to classify the dark pixels as candidate shadow pixels. Then we use a shadow region detection method to filter the candidate shadow pixels and retrieve some refined shadow pixels. In the end, we use those refined shadow pixels estimate the shadow model and classify the foreground pixels into the object class or the shadow class.

3.1 The New Illuminating Criteria for Shadow Pixel Detection

Many algorithms use chromatic criteria to identify shadow pixels. Some color spaces that separate chromaticity from intensity are applied to detect shadows, such as the HSV color space [6], the c1c2c3 color space [11], and the YUV color space [9]. Among those color spaces, the HSV color space is the most widely used one. The HSV color space is composed of H (Hue), S (Saturation), and V (Value) components. The H component represents the color of a pixel, the S component represents the colorfulness of a pixel, and the V component represents the brightness of a pixel. Some of the algorithms assume that the shadow pixels keep the same chromatic information as the background, and have lower illuminating level [6], [11], [9], [7], [10]. So they use the criteria which

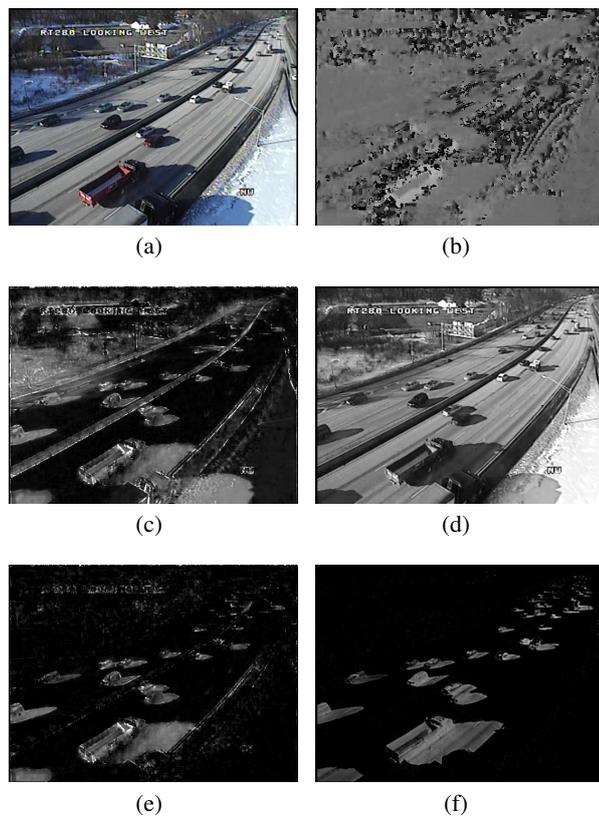


Fig. 2. (a) A video frame from an NJDOT traffic video. (b) The H (hue) component of a video frame. (c) The S (saturation) component of the video frame. (d) The V (value) component of the video frame. (e) The difference of the S component between the frame and the background. (f) The difference of the V component between the background and the frame.

assume that the cast shadows have similar H to the background, but a lower S and a lower V than the background [7]. However, we recognize that is not always the case.

Fig. 2 (a) shows a color video frame, Fig. 2 (b)-(d) display the H (hue), S (saturation), and V (value) components in the HSV color space, Fig. 2 (e) shows the difference of the S component between the foreground and the background, and Fig. 2 (f) shows the difference of the V component between the background and the frame. We can see from Fig. 2 (b) that the H values of the cast shadows are not similar to the background. As a result, in our new illuminating criteria the H values are excluded as they vary a lot especially in some low quality videos. From Fig. 2 (e) and (f) we can see that, for the shadow pixels, the difference between the frame and the background of the S and V components is relatively fixed. Hence we propose a set of illuminating criteria to better detect the shadow pixels by only involving the S and V components in the HSV color space.

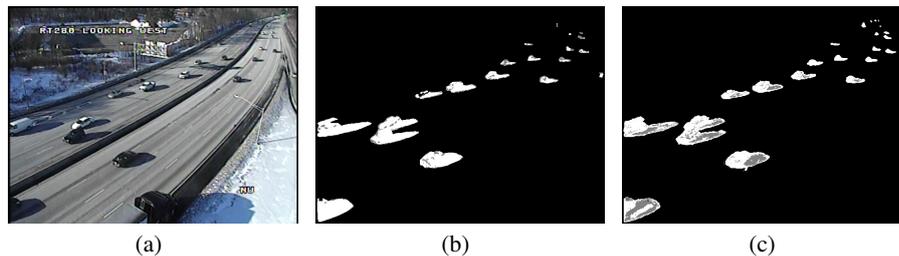


Fig. 3. (a) A video frame from an NJDOT traffic video. (b) The shadow detection results (shadow pixels are represented using gray scale value of 128) using the chromatic criteria in [7], [10] (c) The shadow detection results using our new illuminating criteria.

Let S_f and V_f be the S and V components of a pixel in the foreground region, respectively, and S_b and V_b be the S and V components of the same pixel in the background, respectively. Our new illuminating criteria are defined as follows:

$$\begin{cases} \tau_{sl} < S_f - S_b < \tau_{sh} \\ \tau_{vl} < V_b - V_f < \tau_{vh} \end{cases} \quad (1)$$

where τ_{sl} , τ_{sh} , τ_{vl} , and τ_{vh} represent the thresholds. If a pixel in the foreground region satisfies these illuminating criteria, it is classified as a candidate shadow pixel.

Fig. 3 shows the shadow detection results using our new illuminating criteria and the criteria in [7], [10]. Specifically, Fig. 3 (a) shows a video frame from an NJDOT traffic video, Fig. 3 (b) displays the shadow detection results using the chromatic criteria in [7], [10], and Fig. 3 (c) shows the shadow detection results using our new illuminating criteria. Note that the shadow pixels are represented using gray scale value of 128. We can see from Fig. 3 (b) and (c) that our proposed method using the new illuminating criteria is able to detect the shadow pixels more reliably.

3.2 An Innovative Heuristic Shadow Region Detection Method

As a prior knowledge, the shadow pixels of one object should be in a continuous region, and the shadows of all the objects will be casted on the same side under the sun. Based on this heuristic, we present a shadow region detection method, which can separate each foreground region into the shadow region and the object region. By using the shadow region, we can filter out the candidate shadow pixels outside that region, thus enhancing the shadow detection result.

As the candidate shadow pixels inside each foreground region are detected using the new illuminating criteria, the remaining pixels are the object pixels. For each foreground region B , we can find the centroid of the candidate shadow pixels $Cent_S(B)$ and the centroid of the object pixels $Cent_O(B)$. And a unit vector \vec{V}_B pointing from $Cent_O(B)$ to $Cent_S(B)$ can be calculated. When we deal with the outdoor videos, the light source is the sun. All the shadows should be cast in one direction. So we can use the majority of the unit vectors $\vec{V}_B \in \mathbb{S}_B$ estimate the direction of the light, where \mathbb{S}_B

is the foreground region set. After we get the direction of light \vec{V}_l , a separating line, which is perpendicular to the direction of the light is used to cut each foreground region into two regions: the shadow region and the foreground object region. We assume the percentage of the shadow in each foreground region keeps the same as we get from the new illuminating criteria, we classify that percentage of region as candidate shadow region in each foreground region on \vec{V}_l .

3.3 A New Statistical Shadow Modeling and Classification Method

In traffic videos, most of the shadows are cast on the ground with relatively uniform color. As a result, the low illuminative shadow pixels on the ground should also have a similar color. We can estimate a single Gaussian distribution of shadow pixels as the shadow model.

We first achieve some candidate shadow pixels by using the illuminating criteria. Then we can get some refined shadow pixels by filtering out the candidate pixels out of the shadow area. Those refined shadow pixels tend to be more reliable shadow pixels, so we can apply these shadow pixels to estimate the Gaussian distribution for the shadow class. To enhance the discriminatory power of the pixels [21], we integrate the horizontal and vertical Haar wavelet features with the red, green, and blue color components to form the feature vector \mathbf{x} for each pixel. Let \mathbb{S}_r be the refined shadow pixel set, for each pixel belongs to \mathbb{S}_r , we will build the feature vector \mathbf{x} and use \mathbf{x} to update the Gaussian distribution $N_s(\mathbf{M}, \Sigma)$ as follows:

$$\mathbf{M}' = \mathbf{M} - \alpha(\mathbf{M} - \mathbf{x}) \quad (2)$$

$$\Sigma' = \Sigma + \alpha((\mathbf{M} - \mathbf{x})(\mathbf{M} - \mathbf{x})^t - \Sigma) \quad (3)$$

where \mathbf{M} and Σ are the mean vector and the covariance matrix of the shadow Gaussian distribution, respectively. α is a small number which influences the model updating speed.

For shadow pixel classification, we apply the following discriminant function for each pixel $\mathbf{x} \in \mathbb{R}^d$ in the foreground:

$$s(v_i) = (\mu_i - v_i)^2 - p\sigma_i \quad i \in \{1, 2, \dots, d\} \quad (4)$$

where v_i is the i -th element of the input vector \mathbf{x} , μ_i is the i -th element of the mean vector \mathbf{M} , σ_i is the i -th diagonal element of the covariance matrix Σ , and p is the parameter which determines the threshold. If $s(v_i)$ is greater than zero for any $i \in \{1, 2, \dots, d\}$, we classify \mathbf{x} into the foreground object class. Otherwise we classify it as a shadow pixel. Our new statistical shadow modeling and classification method thus detects the shadow pixels.

Fig. 4 (a) displays a video frame from an NJDOT traffic video, Fig. 4 (b) shows the candidate shadow pixels detected by the new illuminating criteria introduced in Sec. 3.1, Fig. 4 (c) shows the shadow regions detected by the shadow region detection method introduced in Sec. 3.2, Fig. 4 (d) shows the refined shadow pixels, and Fig. 4 (e) shows the shadow detection result using our new statistical shadow modeling and classification method.

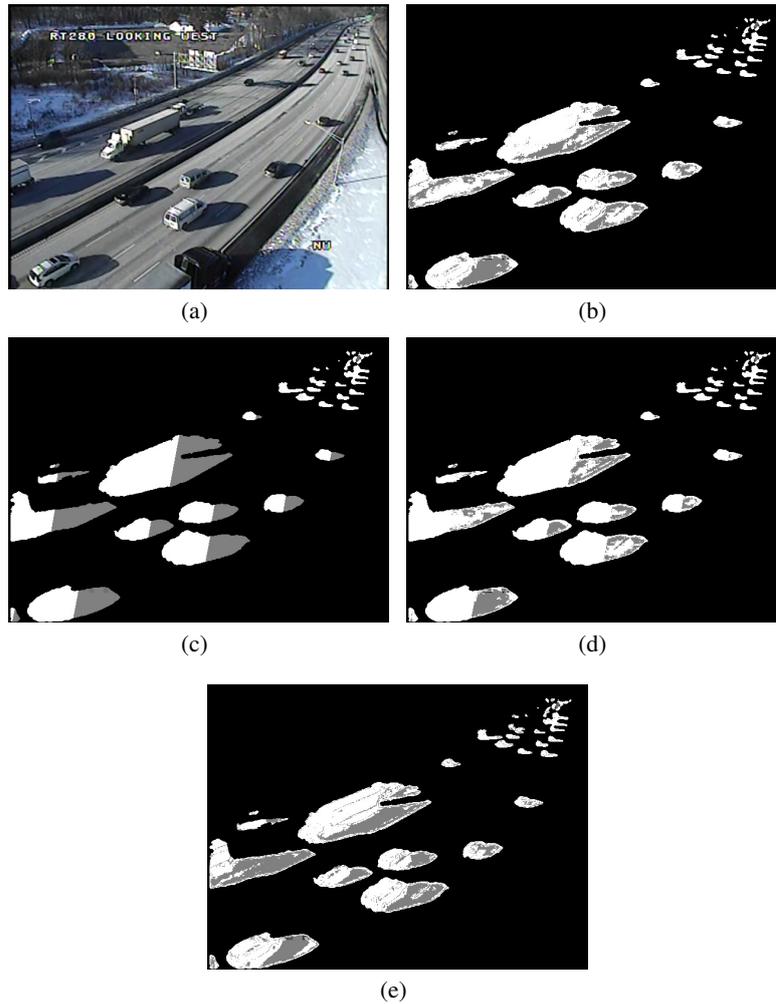


Fig. 4. (a) A video frame from an NJDOT traffic video. (b) The candidate shadow pixels detected using our new illuminating criteria. (c) The shadow regions detected using our shadow region detection method. (d) The refined shadow pixels. (e) The shadow detection result using our new statistical shadow modeling and classification method. Note that both the shadow pixels and the shadow regions are shown using gray scale value of 128.

4 Experiments

We use the New Jersey Department of Transportation (NJDOT) traffic video sequences to evaluate our proposed method quantitative and qualitatively. Specifically, we apply four NJDOT traffic videos, each of which is 15 minutes with a frame rate of 15 frames per second or fps. This dataset is built from real-world traffic surveillance video cameras

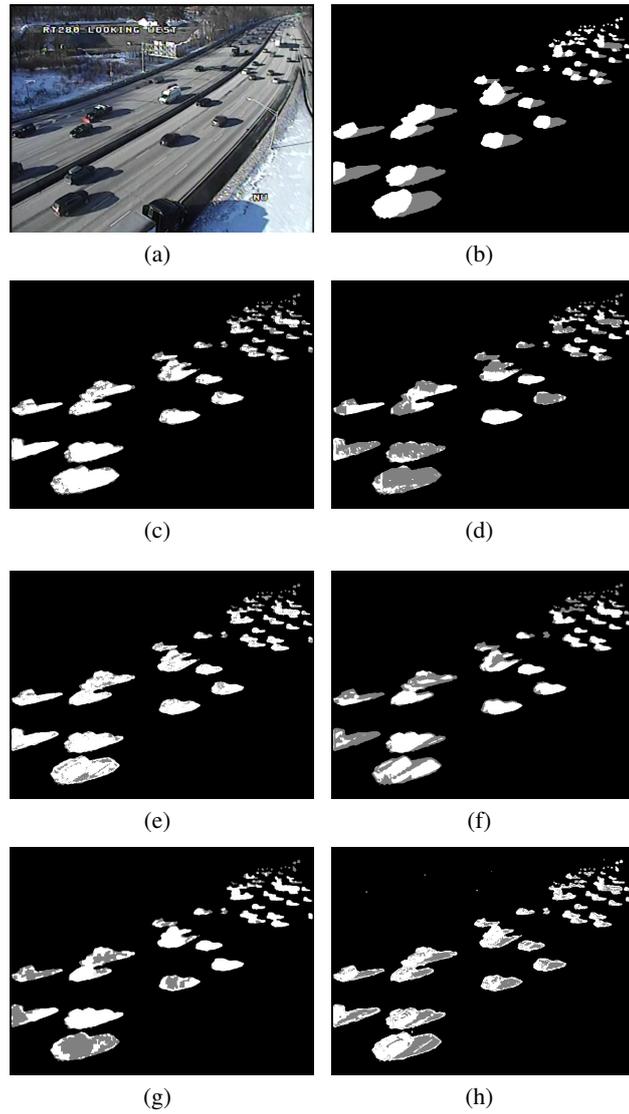


Fig. 5. The foreground masks obtained by different methods. (a). One video frame of the NJDOT video. (b). The ground truth of foreground mask. The white parts are the foreground objects. The gray parts are the cast shadows. (c) The shadow detection result of Cucchiara et al.'s method.[6] (d)The shadow detection result of Huang and Chen's method.[12] (e)The shadow detection result of Hsieh et al.'s method.[13] (f)The shadow detection result of Leone and Distanti's method.[22] (g)The shadow detection result of Sanin et al.'s method.[14] (h)The shadow detection result of our proposed method.



Fig. 6. The comparison of vehicle tracking performance using a frame from the NJDOT traffic videos. Left: the vehicle tracking results without shadow detection. Right: the vehicle tracking results with shadow detection using our proposed shadow detection method.

and has lower resolution and bitrate than other open source datasets. We demonstrate the improvement for the vehicle tracking performance by using our proposed shadow detection method with these videos.

The computer we use is a DELL XPS 8900 PC with a 3.4 GHz processor and 16 GB RAM. The NJDOT videos have the spatial resolution of 640×482 , and it takes 39ms to process each frame using our method. As a result, our proposed shadow detection method is able to perform real time analysis of these videos. The thresholds τ_{sl} , τ_{sh} , τ_{vl} and τ_{vh} in Eq. 1 are defined by the saturation and value components of some manually selected shadow pixels in the first frame of the video.

The videos in the NJDOT dataset have stronger cast shadows and lower video quality. Many shadow detection methods fail to detect shadows in these videos, but our proposed method is able to achieve good shadow detection performance on these videos as shown in Fig. 5. We can see from Fig. 5 that our proposed shadow detection method achieves better shadow detection and removal results than the other popular shadow detection methods.

The shadow detection rate η , the shadow discrimination rate ξ , and the F-measure are popular metrics used to evaluate shadow detection performance quantitatively [23], which are defined as follows:

$$\eta = \frac{TP_s}{TP_s + FN_s} \quad (5)$$

$$\xi = \frac{TP_o}{TP_o + FN_o} \quad (6)$$

$$F - measure = \frac{2\eta\xi}{\eta + \xi} \quad (7)$$

where TP_s and FN_s represent the number of true positive and false negative shadow pixels, respectively, and TP_o and FN_o stand for the number of true positive and false negative foreground object pixels, respectively.

Table. 1 shows the comparative shadow detection performance of our proposed method and some popular shadow detection methods. In particular, our proposed method

achieves the highest F-measure score of 71.7%, compared with the 16.1%, 50.0%, 12.3%, 49.9%, 34.5% F-measure scores by the Cucchiara et al.[6] shadow detection method, Huang and Chen [12] shadow detection method, Hsieh et al. [13] shadow detection method, Leone and Distanto [22] shadow detection method, and Sanin et al. [14] shadow detection method, respectively.

Table 1. The comparative shadow detection performance of our proposed method and some popular unsupervised shadow detection methods.

Methods	η	ξ	$F - measure$
Cucchiara et al.[6]	8.9%	82.5%	16.1%
Huang and Chen [12]	46.5%	54.1%	50.0%
Hsieh et al. [13]	6.6%	89.5%	12.3%
Leone and Distanto [22]	38.6%	70.6%	49.9%
Sanin et al. [14]	22.1%	78.3%	34.5%
Our proposed method	61.0%	86.9%	71.7%

The significance of shadow detection in these videos is to improve the performance of video analysis tasks such as tracking and object detection. In particular, Fig. 6 shows comparatively the vehicle tracking performance using the NJDOT traffic videos: the vehicle tracking results without shadow detection and the vehicle tracking results with shadow detection using our proposed shadow detection method. We can see in the left figure that two vehicles are connected together by their cast shadows and fall into one tracking block when no shadow detection algorithm is applied. After applying our shadow detection algorithm, these two vehicles are separated into two tracking blocks. As a result, the tracking performance is more accurate.

5 Conclusion

We have presented in this paper an unsupervised moving shadow detection method for video analysis. The major contributions of our proposed method are three-fold.

First, we propose a set of new illuminating criteria for shadow pixels' differentiation. Second, we use an innovative heuristic shadow region detection method to detect the continuous shadow regions, and filter the candidate shadow pixels. Third, we build a statistical shadow model to model and classify the shadow pixels with a single Gaussian distribution. The model keeps learning and updating to adapt to the changes of the environment.

The experimental results using the NJDOT video sequences have shown that (i) our proposed method achieves better shadow detection performance than other popular unsupervised shadow detection methods, (ii) our proposed method is able to detect cast shadows in low quality videos, such as the NJDOT videos, while in comparison other methods fail to detect the shadows, and (iii) our proposed method is able to improve the performance of video analysis tasks, e.g. vehicle tracking.

Acknowledgments: This paper is partially supported by the NSF grant 1647170.

References

1. Bouwmans, T., Sobral, A., Javed, S., Jung, S.K., Zahzah, E.: Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset. *Computer Science Review* **23** (2017) 1–71
2. Bouwmans, T., Silva, C., Marghes, C., Zitouni, M.S., Bhaskar, H., Frelicot, C.: On the role and the importance of features for background modeling and foreground detection. *Computer Science Review* **28** (2018) 26–91
3. Shi, H., Liu, C.: A new foreground segmentation method for video analysis in different color spaces. In: 2018 24th International Conference on Pattern Recognition (ICPR), IEEE (2018) 2899–2904
4. Shi, H., Liu, C.: A new global foreground modeling and local background modeling method for video analysis. In: International Conference on Machine Learning and Data Mining in Pattern Recognition, Springer (2018) 49–63
5. Prati, A., Mikic, I., Trivedi, M.M., Cucchiara, R.: Detecting moving shadows: algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(7) (2003) 918–923
6. Cucchiara, R., Grana, C., Piccardi, M., Prati, A.: Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2003)
7. Sanin, A., Sanderson, C., Lovell, B.C.: Shadow detection: A survey and comparative evaluation of recent methods. *Pattern recognition* **45**(4) (2012) 1684–1695
8. Mahajan, R., Bajpayee, A.: A survey on shadow detection and removal based on single light source. In: 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), IEEE (2015) 1–5
9. Chen, C.T., Su, C.Y., Kao, W.C.: An enhanced segmentation on vision-based shadow removal for vehicle detection. In: 2010 International Conference on Green Circuits and Systems (ICGCS), IEEE (2010) 679–682
10. Gomes, V., Barcellos, P., Scharcanski, J.: Stochastic shadow detection using a hypergraph partitioning approach. *Pattern Recognition* **63** (2017) 30–44
11. Salvador, E., Cavallaro, A., Ebrahimi, T.: Cast shadow segmentation using invariant color features. *Computer vision and image understanding* **95**(2) (2004) 238–259
12. Huang, J.B., Chen, C.S.: Moving cast shadow detection using physics-based features. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, IEEE (2009) 2310–2317
13. Hsieh, J.W., Hu, W.F., Chang, C.J., Chen, Y.S.: Shadow elimination for effective moving object detection by gaussian shadow modeling. *Image and Vision Computing* **21**(6) (2003) 505–516
14. Sanin, A., Sanderson, C., Lovell, B.C.: Improved shadow removal for robust person tracking in surveillance scenarios. In: 20th International Conference on Pattern Recognition (ICPR), 2010, IEEE (2010) 141–144
15. Guo, R., Dai, Q., Hoiem, D.: Paired regions for shadow detection and removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(12) (2013) 2956–2967
16. Vicente, T.F.Y., Hoai, M., Samaras, D.: Leave-one-out kernel optimization for shadow detection and removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40**(3) (2018) 682–695
17. Vicente, Y., Tomas, F., Hoai, M., Samaras, D.: Leave-one-out kernel optimization for shadow detection. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 3388–3396

18. Shen, L., Wee Chua, T., Leman, K.: Shadow optimization from structured deep edge detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 2067–2074
19. Khan, S.H., Bennamoun, M., Sohel, F., Togneri, R.: Automatic shadow detection and removal from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **3** (2016) 431–446
20. Qu, L., Tian, J., He, S., Tang, Y., Lau, R.W.: Deshadownet: A multi-context embedding deep network for shadow removal. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). Volume 1. (2017) 3
21. Liu, C., ed.: *Recent Advances in Intelligent Image Search and Video Retrieval*. Springer (2017)
22. Leone, A., Distanti, C.: Shadow detection for moving objects based on texture analysis. *Pattern Recognition* **40**(4) (2007) 1222–1233
23. Ji, W., Zhao, Y.: Moving cast shadow detection using joint color and texture features based on direction and distance. In: 2016 2nd IEEE International Conference on Computer and Communications (ICCC), IEEE (2016) 439–444

Social Activism Analysis: An Application of Machine Learning in the World Values Survey

Francielle M. Nascimento¹, Dante A. C. Barone¹, and Henrique Carlos de Castro²

¹ Institute of Informatics, UFRGS, Porto Alegre - RS, Brazil
{francielle.nascimento,barone}@inf.ufrgs.br

² Political Science, UFRGS, Porto Alegre - RS, Brazil
Principal Investigator of the WVS and National Director for Brazil
henrique@ufrgs.br

Abstract. The study of social sciences is essential to understand different dimensions of human society. Different researches are done to understand human development and its relationships in the community. Given this, in this paper, we have developed a methodology to use typical social science metrics and resources in conjunction with Artificial Intelligence techniques. The goal is to collaborate with research and visualize patterns to help explain human behavior. In this way, we use the World Values Survey's fifth wave data to apply self-learning methods and contribute to the advancement of social science research. We use algorithms to perform classifications, such as Random Forest, Stochastic Gradient Descent and Support Vector Machine in data collected in 58 countries, to verify if there are social patterns that can explain political participation. Thus, we identified that there is a stronger relationship in the results found in the so-called advanced democracies (USA and Europe) compared to those in other societies. From this, we can consider that eventual adjustments in the theory underlying the WVS research or in the instruments of data collection could be made and that more studies are needed to analyze other dimensions.

Keywords: Social Sciences · WVS · political participation · Artificial Intelligence.

1 Introduction

With the advent of globalization, world society is undergoing a process of internationalization and thought reformulation. This evolution has triggered a globalized economy with new standards, practices and cultures, and a worldwide concern for solutions to the problems faced in global proportions. [8]. In this way, mapping and understanding these problems has become increasingly necessary.

Thus, Inglehart [12] developed a method of application of questionnaires in several countries to understand the changes of values and their implications over time. It was called World Values Survey (WVS), forming, then, a database for

studies of social scientists. In this sense, the possibility of applying Artificial Intelligence techniques [9] can generate information and assist in the understanding of world society. Also, it can help to understand the course of value changes regarding economic, cultural and democratic development, according to Inglehart [12] (2003). Knowing this, the use of machine learning to obtain a system to mine data, extract information and standards, from WVS data is proposed.

In this work, we proposed to use machine learning techniques to complement social science studies in the WVS database. This paper is organized as follows. Section 2 discusses the purpose of the WVS for the study of social sciences. Section 3 defines the research pipeline, from the data preparation to the evaluation of machine learning models applied to WVS data. Section 4 shows the analysis of results and the last section contains the conclusions.

2 World Values Survey

Since the advent of globalization, world society has undergone several transformations in the process of internationalization and reformulation of thoughts and actions in a global order. This evolution has triggered the emergence of a globalized economy, new patterns, practices, cultures, political processes, social hierarchies, and global governance [8].

Concurrently, existing problems have taken on global proportions, with new formations such as inequality, hunger, disease, wars, and terrorism among others. Thus, it becomes necessary to understand and map these problems, to seek solutions at global levels to be applied.

As a result, social scientists began to study the change in values and the impact on social and political life from a World Values Survey Association (WVS)[4] initiative to verify the hypothesis that economic and technological changes transform societies values. In 1981, this research became well known. Its main researcher Ronald Inglehart [4], had conducted a survey in many different countries, which became an important instrument in forming the World Values Survey(WVS). In Europe, the data are collected in collaboration with the European Values Study (EVS).

The WVS is a survey conducted in more than 100 countries with complex questions that map out societal characteristics such as economic development, democratization, religion, gender equality, and others. In this way, a WVS database was built and made available by the organization with the aim of scientists developing studies on world values. This database currently has questionnaires and six wave responses (1981-1984, 1990-1994, 1995-1998, 1999-2004, 2005-2009, 2010-2014), and wave 7 is in progress.

3 Data Preparation and Model Definition

3.1 WVS Database

The WVS Database is composed of questions and answers to various aspects to map the behavioral change of society as a whole over the years. For each

application of the questionnaire, there are questions of the economic, social and cultural core, applied using probabilistic population sampling stratified in each of the participating countries. The average population sample of the participating countries is 1500 respondents to the questionnaire, depending on population size.

The Database consists of variables (corresponding to the questionnaire questions) and the value labels for each issue (which correspond to the answers). For each instance, we have a set of questions and their answers.

In this work, we have chosen Wave 5 for our studies. In Wave 5, there are 58 participant countries, and 258 questions [1]. The issues between 4 and 233 for building the models, were selected, because these questions belong to the central core and don't allow bias. Also, we chose them with the objective of understanding human behavior in regards to political activism.

To complete this study effectively, we decided to start our analysis at wave five, because it had more participation from countries of different continents enabling an overall summary. Contained within this study, were a total of fifty-eight participating countries from around the world and an extensive survey of over two hundred and fifty questions related to individual opinions on different topics. For example, the importance of friends, politics, and child obedience, or issues related to the fitness of politicians who don't believe in God to serve in public office. For this study, we decided to select a limited number of questions that don't correspond to demographic items, based on the expectancy of finding some relationship between values and human behavior. The questionnaire can be found at the website of WVS ¹. To satisfy the needs of all participants involved in this study, WVS systematically translated each survey question according to their specific language (if applicable). Due to the intense nature of the study, specific guidelines and a code of ethics have been instilled so that the WVS survey teams reduce any bias and further limitations throughout the questioning process.

To ensure an accurate national sampling, WVS has relied upon the stratified sampling method because it allows dividing into groups mutually exclusive and frequent, allows discriminating different behaviors within the population. Thus, the sampling can reflect the style of the general population of different places, sexes, genders, and ages among other things. It is important to note that the WVS database is public with free access and available for researchers to carry out studies on this basis.

3.2 Data Processing

In this step, some methods were used to organize the data and build the architecture of the model. We re-defined some questions of WVS for binary classification. We have chosen issues related to our principal problem about social activism. Hence, we considered nine questions from WVS, regarding the active participation of the interviewee in voluntary organizations from a list (see Figure 1). The belonging to the class was considered true when at least one of the questions

¹ <http://www.worldvaluessurvey.org/WVSDocumentationWV5.jsp>

about active membership was answered positively, and false, when none of the questions was answered positively.

Now I am going to read off a list of voluntary organizations. For each one, could you tell me whether you are an active member, an inactive member or not a member of that type of organization? (*Read out and code one answer for each organization*):

	Active member	Inactive member	Don't belong
V24. Church or religious organization	2	1	0
V25. Sport or recreational organization	2	1	0
V26. Art, music or educational organization	2	1	0
V27. Labor Union	2	1	0
V28. Political party	2	1	0
V29. Environmental organization	2	1	0
V30. Professional association	2	1	0
V31. Humanitarian or charitable organization	2	1	0
V32. Consumer organization	2	1	0
V33. Any other (<i>write in</i>): _____	2	1	0

Fig. 1. Questions of Wave five correspondents the variables for defining the class, with the list of voluntary organizations [1].

In Table 1 we can observe the results of this criterion used to obtain the classes for the task. The characteristics available for each country show us that there are countries that have a balanced data set of a positive and negative target, as South Africa, Rwanda, and Brazil, while most of the other countries are unbalanced. This way, we expect that the countries that have few samples in one of the classes, it will to present a low performance in the metrics of evaluate.

Distribution by Class				Distribution by Class				Distribution by Class			
Country	Positive	Negative	Total	Country	Positive	Negative	Total	Country	Positive	Negative	Total
Andorra	108	895	1003	Indonesia	758	1257	2015	Serbia	46	1174	1220
Argentina	171	831	1002	Iran	532	2135	2667	Vietnam	89	1406	1495
Australia	226	1195	1421	Italy	92	920	1012	Slovenia	129	908	1037
Brazil	772	728	1500	Japan	47	1049	1096	South Africa	1561	1427	2988
Bulgaria	18	983	1001	Jordan	37	1163	1200	Spain	109	1091	1200
Canada	615	1549	2164	South Korea	228	972	1200	Sweden	67	936	1003
Chile	225	775	1000	Malaysia	187	1014	1201	Switzerland	243	998	1241
China	55	1936	1991	Mali	561	973	1534	Thailand	297	1237	1534
Taiwan	91	1136	1227	Mexico	639	921	1560	Trinidad and Tobago	432	570	1002
Colombia	741	2284	3025	Moldova	135	911	1046	Turkey	19	1327	1346
Cyprus	66	984	1050	Morocco	17	1183	1200	Ukraine	53	947	1000
Ethiopia	447	1053	1500	Netherlands	147	903	1050	Egypt	24	3027	3051
Finland	180	834	1014	New Zealand	155	799	954	United Kingdom	189	852	1041
France	45	956	1001	Norway	85	940	1025	United States	466	783	1249
Georgia	47	1453	1500	Peru	378	1122	1500	Burkina Faso	365	1169	1534
Germany	268	1796	2064	Poland	125	875	1000	Uruguay	146	854	1000
Ghana	1105	429	1534	Romania	95	1681	1776	Zambia	932	568	1500
Hungary	72	935	1007	Russia	46	1987	2033				
India	438	1563	2001	Rwanda	795	712	1507				

Table 1. Distribution of the obtained classes by Country

For feature selection, we use the method called recursive feature elimination (RFE), where resources in each interaction according to coefficient obtained from the estimator weights to features, are removed [3]. This approach evaluates the performance of attributes set for making predictions, and how each features influences in a group of sets in the final model. Lastly, the results represent the best collection of features for the model. We expect that the set of selected features could explain some differences between the types of the countries, with respect to the culture, politics, and economy, which we plan to evaluate in our future work.

Also, we removed some variables and cleaned the data. We evaluate different numbers of variables using RFE the criterion to decide them to composing the model, and we choose the best according to tests. This way, for each country 40 variables, were selected. Furthermore, to prevent data leakage [5] the process of data normalization during cross-validation was deployed. Above all, it seems pertinent to remember that this method has been implemented to all the countries studied in Wave 5.

3.3 Models and Evaluation

With the data ready to be used, we elaborate some models of machine learning for building the classifications and evaluate results. Due to our previous knowledge in the area, we have considered that four models are enough for testing. We have chosen the following Machine Learning methods: Support Vector Machine (SVM), Random Forest Classifier (RFC), linear models with Stochastic Gradient Descent (SGD), and a neural network Multi-layer Perceptron (MLP). Significantly, these four models were applied to each country.

The validation of the model has been done using Stratified K-Folds cross-validator, in which ten different training and test sets were separated computing the evaluation metrics of each group to obtain the mean and the standard deviation (STD) of the results.

We also used measures to evaluate the predictive and classification models, from the F1-Score and Matthews correlation coefficient (MCC) [6]. The MCC is a measure of quality, which analyzes (binary) classifications even when there is an imbalance between occurrence and non-occurrence classes. The coefficient assumes values ranging from -1 and +1, where +1 coefficients relate to a perfect prediction, 0 random predictions and -1 imperfect prediction (total disagreement). This measure is relevant in this work since it makes a global analysis of the predictions and indicates the quality of the binary classifications in a context of the confusion matrix.

4 Analysis of Results

In the evaluation phase of the models, have built different perspectives for analyzing the results. For each country, we choose the model that presents the best F1-Score. In this research we have decided to calculate the metrics F1-Score and

MCC, where the F1-Score is the harmonic mean of precision and recall measures and the MCC assumes values that range between $[-1$ and $+1]$, where $+1$ coefficients correspond to a perfect prediction, 0 to random prediction and -1 to imperfect prediction, respectively.

The TreeMap [10] in Figure 2 shows the distribution of countries according to F1-Score. The best result among countries can be seen at the extreme left, and the worst is at the extreme right of the map. We can observe that Australia got the 0.8811 F1-Score representing the best performance, and other countries like the United States, New Zealand, Canada, Netherlands, South Korea, United Kingdom, also presented satisfactory F1-Score [2] in the range of 0.80. The countries that offered the worst performance are Morocco, Turkey, Bulgaria, and Jordan, with F1-Score on average of 0.49.

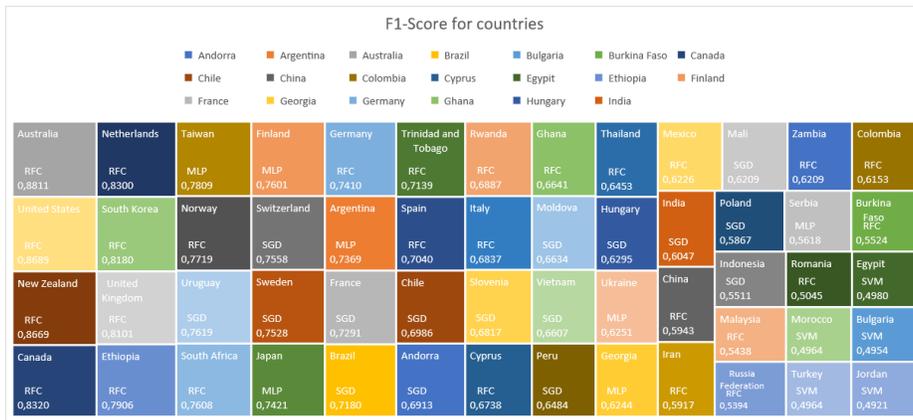


Fig. 2. Tree Map of F1-Score for countries

In this perspective, noting the distribution of TreeMap Figure 3 with hierarchy by continents, we can check that at Wave five, that Europe there is more representative, followed by Asia, Africa, South America, North America and finally Oceania.

We calculate the average and STD for continents, being that Oceania attains the best result with 0.874 of F1-Score and 0.0100 of STD. North America with 0.7594 and 0.1126, and, South America with 0.6965 and 0.0552. The worst performance is at Asia with 0.6265 and 0.1023, and Africa, with 0.6325 and 0.1054, F1-Score and STD, respectively.

The graph of Figure 4 shows the average of the F1-Score and the standard deviation by country. We can note that the vast majority of federations introduce an STD relatively small at the folds of cross-validation, varying between 0.0004 and 0.09. The region that shows the worst STD is Thailand, Japan, Hungary, Cyprus, Ukraine, Argentina, Sweden, with values between 0.10 and 0.15.

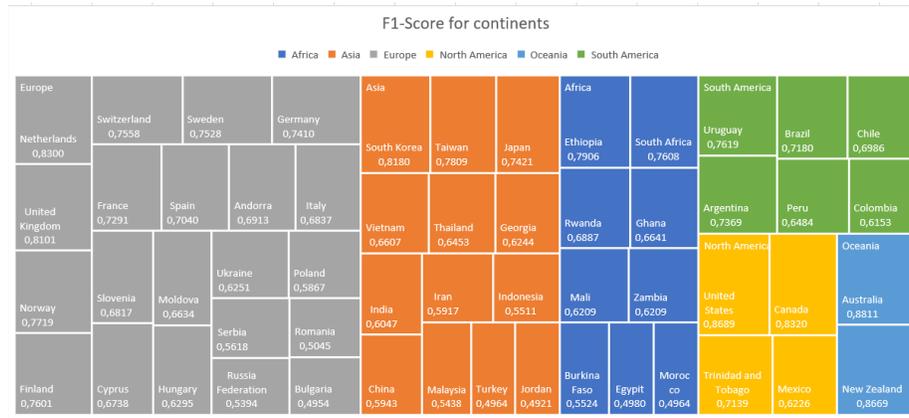


Fig. 3. TreeMap of F1-Score for continents

Analyzing the Pareto diagram in the Figure 5, we can see that 64% of the countries presented F1-Score between 0.5901 and 0.7861, which shows that the models were able to learn a satisfactory pattern during the classifications, in addition, 15% of the countries appeared with an excellent performance, with values varying between 0.7861 and 0.8841, and the remaining 21% with F1-Score of 0.4921 and 0.5901.

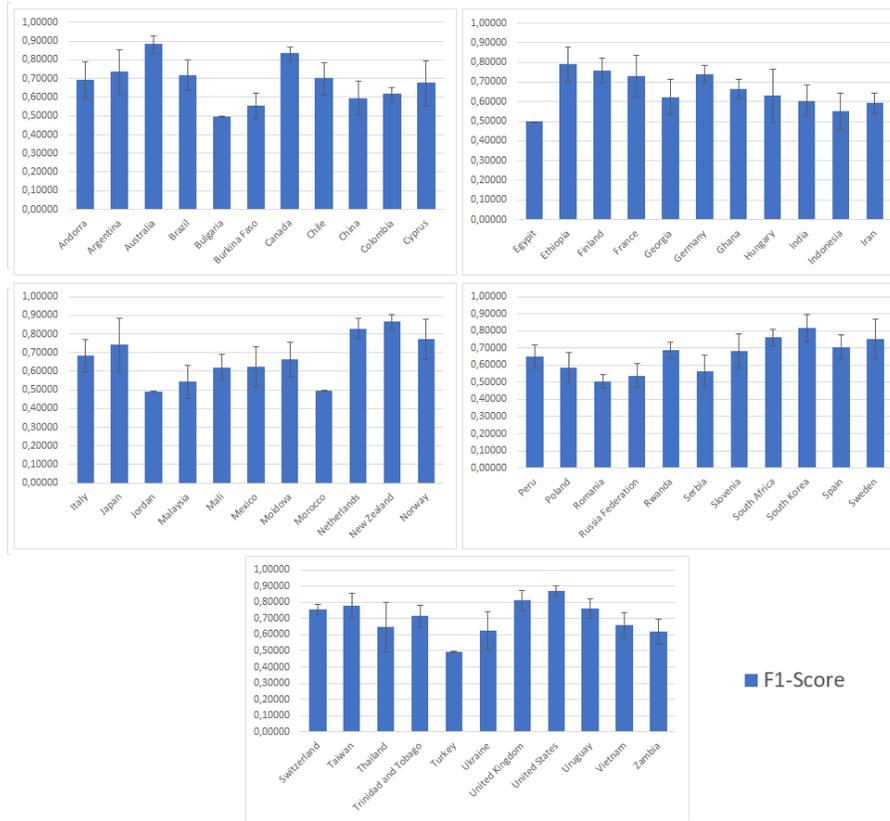


Fig. 4. Mean and Standard deviation by countries

Analyzing Figure 6 we can observe that the RFC Model showed greater variability, besides that according to superior limits the boxplot, most of the countries ranked with this model were better. And the worst model was SVM, with less variability and F1-Score of 0.49 on average. The SGD and MLP behaved results similarly, compared to the RFC. We can understand why the RFC shows the best performance because of the task uses the Wisdom of the Crowd, which significantly improves the results, making the model more accurate and reliable than the other [11]. Also, the data provide a framework that allows generalizations made by the RFC, because it is a survey structure.

Nyman & Ormerod applied machine learning models in research to predict economic recessions. The author used algorithms such as ordinary least squares regression and RFC, in which the latter presented better results. Thus, we can see that, for this task, the RFC is more appropriate and perform well in other problems like this available in the literature. [7]

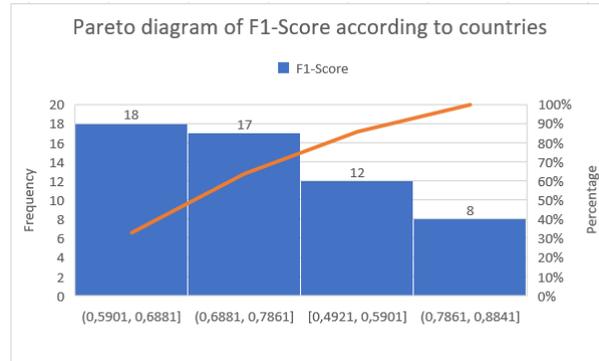


Fig. 5. Country distribution by F1-Score range

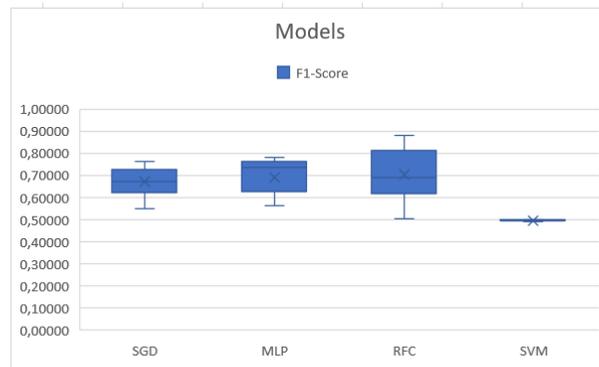


Fig. 6. Comparison between the models

The coefficient of Matthew's can demonstrate like the task of classification behaved in the general aspect at the models and like the predictions were satisfactory for each country. Figure 7 Shows this concept, and we can observe that some countries presented an MCC of zero, like, Morocco, Jordan, Egypt, Bulgaria, and Turkey, that is to say, the predictions were random, which allows us to verify that in these countries the model failed to identify a pattern about the social activism task, for not having expressive differences that allows to separate the classes. On the other hand, the other countries presented positive MCC, of these eighteen federations presented MCC with values above 0.50, and some of them very close to 0.80.

Above all, the aspects analyzed exhibit that the model's performance was satisfactory for this task, and we can perceive that there is a pattern in most countries that can explain why people choose social activism, according to results to F1-Score and MCC. This way, we observed that countries that have zero in MCC are some that have unbalanced data, but despite it, France has few samples in positive class and present a significant performance in F1-Score (0,7291), and

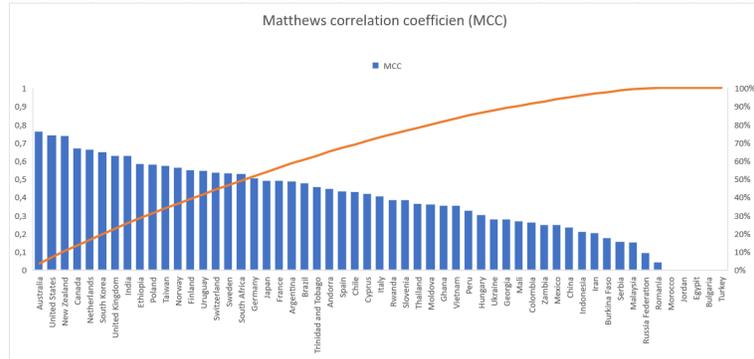


Fig. 7. Matthews correlation coefficient by countries

MCC (0,4939), Japan, Cyprus, Sweden, Norway, Vietnam are some examples with same behavior. Therefore, we can notice the models has clues that could differentiate and realize the classify, even with situations of unbalanced data.

5 Conclusions

The WVS research was built based on theories of the political and social behaviors found in the so-called advanced democracies (Europe and USA). Thus, the variables of the WVS dataset here analyzed seem to fit better to the values existing in those societies (as the results suggested). In societies with different social constructs and histories, the WVS research data may not reflect, at least in relation to the dimension analyzed (political participation), the behavior of the individuals. Given this, the present paper indicates, in a preliminary way, that it may be necessary to review the subjacent theory of certain dimensions of the WVS research, or the data collection instruments (questionnaire). Finally, new data treatments are needed to affirm whether this pattern is found in other dimensions of analysis or whether it is only a characteristic of the political participation dimension.

Acknowledgements: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

References

1. Association, W.V.S., et al.: World values survey. Wave 5, 2005–2008 (2005)
2. Frakes, W.B., Baeza-Yates, R.: Information retrieval: Data structures & algorithms, vol. 331. prentice Hall Englewood Cliffs, NJ (1992)
3. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. *Machine learning* **46**(1-3), 389–422 (2002)
4. Inglehart, Haerpfer, R.C., Moreno, A., Welzel, C., Kizilova, K., Diez-Medrano, J., Lagos, M., Norris, P., Ponarin, E., Puranen, B.: World values survey: Round one - country-pooled datafile (2014), www.worldvaluessurvey.org/WVSDocumentationWV1.jsp.
5. Kaufman, S., Rosset, S., Perlich, C., Stitelman, O.: Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **6**(4), 15 (2012)
6. Matthews, B.: Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure* **405**(2), 442 – 451 (1975). [https://doi.org/https://doi.org/10.1016/0005-2795\(75\)90109-9](https://doi.org/https://doi.org/10.1016/0005-2795(75)90109-9), <http://www.sciencedirect.com/science/article/pii/0005279575901099>
7. Nyman, R., Ormerod, P.: Predicting economic recessions using machine learning algorithms. arXiv preprint arXiv:1701.01428 (2017)
8. Robinson, W.I.: Theories of Globalization, chap. 6, pp. 125–143. Wiley-Blackwell (2008). <https://doi.org/10.1002/9780470691939.ch6>, <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470691939.ch6>
9. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education, 2 edn. (2003)
10. Schreck, T., Keim, D., Mansmann, F.: Regular treemap layouts for visual analysis of hierarchical data. In: Proceedings of the 22Nd Spring Conference on Computer Graphics. pp. 183–190. SCCG '06, ACM, New York, NY, USA (2006). <https://doi.org/10.1145/2602161.2602183>, <http://doi.acm.org/10.1145/2602161.2602183>
11. Wang, L., Michoel, T.: Wisdom of the crowd from unsupervised dimension reduction. arXiv preprint arXiv:1711.11034 (2017)
12. Welzel, C., Inglehart, R., Kligemann, H.D.: The theory of human development: A cross-cultural analysis. *European Journal of Political Research* **42**(3), 341–379 (2003). <https://doi.org/10.1111/1475-6765.00086>, <https://onlinelibrary.wiley.com/doi/abs/10.1111/1475-6765.00086>

Deep Dilated Convolutional Nets for the Automatic Segmentation of Retinal Vessels

Ali Hatamizadeh¹, Hamid Hosseini², Zhengyuan Liu¹, Steven D. Schwartz²,
and Demetri Terzopoulos¹

¹ Computer Science Department, Henry Samueli School of Engineering
² Stein Eye Institute, David Geffen School of Medicine
University of California, Los Angeles, CA 90095, USA

Abstract. The reliable segmentation of retinal vasculature can provide the means to diagnose and monitor the progression of a variety of diseases affecting the blood vessel network, including diabetes and hypertension. We leverage the power of convolutional neural networks to devise a reliable and fully automated method that can accurately detect, segment, and analyze retinal vessels. In particular, we propose a novel, fully convolutional deep neural network with an encoder-decoder architecture that employs dilated spatial pyramid pooling with multiple dilation rates to recover the lost content in the encoder and add multiscale contextual information to the decoder. We also propose a simple yet effective way of quantifying and tracking the widths of retinal vessels through direct use of the segmentation predictions. Unlike previous deep-learning-based approaches to retinal vessel segmentation that mainly rely on patch-wise analysis, our proposed method leverages a whole-image approach during training and inference, resulting in more efficient training and faster inference through the access of global content in the image. We have tested our method on three publicly available datasets, and our state-of-the-art results on both the DRIVE and CHASE-DB1 datasets attest to the effectiveness of our approach.

Keywords: Retina Vessel Segmentation · Width Estimation · Dilated Spatial Pyramid Pooling · Convolutional Neural Networks · Deep Learning.

1 Introduction

The retina and its vasculature are directly visible due to the optically clear media of the human eye. As the only part of the central nervous system that can be rapidly and non-invasively imaged with a variety of modalities in the out-patient setting, the retina provides a window into the human body, thus offering the opportunity to assess changes associated with systemic diseases such as hypertension, diabetes, and neurodegenerative disorders. The sequelae of these conditions, specifically stroke, heart disease, and dementia represent major causes of morbidity and mortality in the developed world. To date, all classification schemes for retinal vascular changes in these conditions, particularly in the early stages

of disease, have been based on qualitative changes based on human assessment. We and others hypothesize that biomarkers of seriously adverse health events exist in the quantitative assessment of retinal vasculature changes associated with early, even asymptomatic, diabetes, hypertension, or neurodegenerations. Specifically, high blood pressure, for example, causes structural changes in the macro- and micro-vasculature of vital organs throughout the body, including the brain, heart, and kidney. The retinal vasculature is similarly impacted but has the advantage of accessibility to multimodal imaging, providing the opportunity to quantitatively assess prognosis, risk, and response to treatment. Narrowing of retinal vessels has been described as an early, classic sign of hypertension. However, this early sign is difficult to use in everyday clinical practice, which usually includes only non-quantitative, subjective visual assessment of the retina by examination, photograph, or even angiography. An automated, quantitative, reliable, reproducible tool that measures changes in the retinal vasculature in response to disease and intervention might augment and disrupt current evaluation and treatment paradigms by allowing physicians to detect disease, predict outcomes, and assess interventions much earlier in the course of disease, thereby opening the potential for improved outcomes in major unmet public health needs.

A critical step in tracking important structural changes of the retinal vasculature is segmentation of the retinal vessels, as it enables locating the veins and arteries and extracting relevant information such as a profile of the width changes of the vessels. Since the manual segmentation of vessels by clinicians is a notoriously laborious and error-prone process, it is important to establish fully automated and reliable segmentation methods that can be leveraged for extracting the aforementioned information with minimal supervision.

Since the advent of deep learning, Convolutional Neural Networks (CNNs) have become popular due to their powerful, nonlinear feature extraction capabilities in many computer vision related applications [5, 7, 14, 2]. Several researchers have applied CNNs to the task of retinal vessel segmentation in fundus images. However, most are patch-wise methods that ignore the global context in the image and are usually inefficient during inference. Melinšćak et al. [11] employed a simple 10-layer CNN architecture based on a patch-wise technique, but their results suffer from low sensitivity in comparison to other techniques. Fu et al. [4] treated the problem of segmentation as a boundary detection problem and combined a CNN with a conditional random field to address the segmentation of retinal vessels, but their method is slower than whole-image CNNs and is outperformed by a number of other proposed methods in different metrics. Zhuang [16] proposed an architecture based on U-Net [13], which utilizes multiple-path networks that leverages a path-wise formulation in segmenting retinal vessels.

In the present paper, we exploit the power of CNNs to create a reliable, fully automated, method that can accurately detect and segment retinal vessels, and we devise an algorithm for the automatic quantification of widths in retinal vessels directly from the segmentation masks, which can be employed toward the creation the aforementioned biomarkers. In particular, we introduce an encoder-decoder CNN architecture that leverages a new dilated spatial pyra-

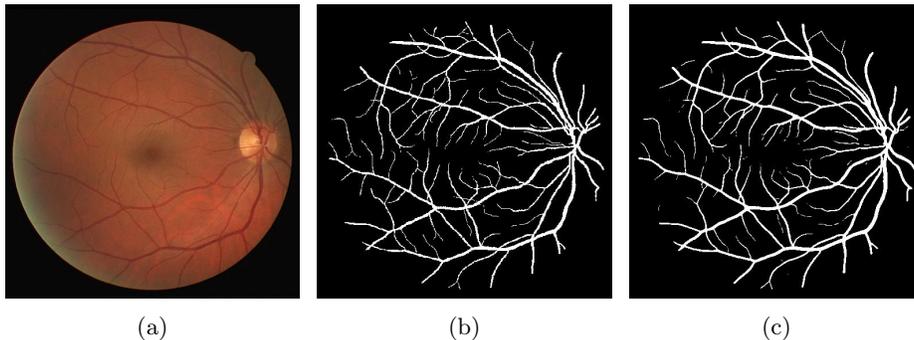


Fig. 1: Segmentation of an image from the DRIVE dataset. (a) Input image. (b) Ground truth. (c) Segmentation prediction output by our network.

mid pooling with multiple dilation rates, which preserves resolution yet adds multiscale information to the decoder. Figure 1 shows an input fundus image with its corresponding ground-truth and the vascular segmentation output by our network.

2 Method

2.1 Vessel Segmentation

We propose a fully convolutional encoder-decoder architecture, as depicted in Figure 2, which leverages dilated residual blocks along with deep supervision at multiple scales for effectively learning the multiresolution details of retinal vessels. Each convolutional layer with kernel W is followed by a rectified linear unit (ReLU) $Re(X) = \max(0, X)$ and a batch normalization $BN_{\gamma, \beta}(X)$ with parameters γ, β that are learned during training. Consequently, every location i in the output of a convolutional layer followed by ReLU and batch normalization can be represented as

$$Y(i) = BN_{\gamma, \beta}(Re(\sum_{j=1} X[i + j \cdot r]W[j])), \quad (1)$$

where r is the dilation rate. We employ both standard and dilated convolutional layers for which the value of r in the former is 1 and in the latter depends on where it is used. In this work, we utilize dilated residual blocks that consist of two consecutive dilated convolutional layers whose outputs are fused with the input.

Our encoder-decoder architecture spans four different resolutions. In the encoder, each path consist of 2 consecutive 3×3 convolutional layers, followed by a dilated residual unit with a dilation rate of 2. Before being fed into the dilated residual unit, the output of these convolutional layers are added with the output

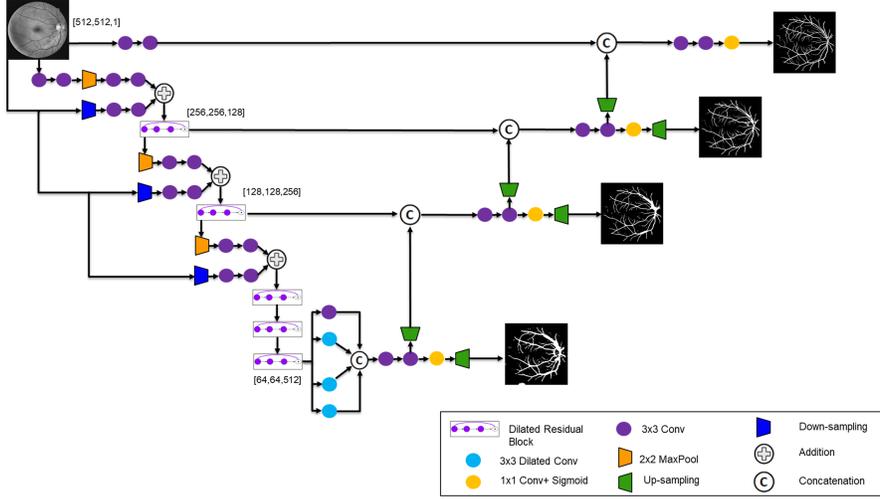


Fig. 2: Our proposed fully convolutional architecture. Dilated spatial pyramid pooling aggregates the outputs of the multiple stages.

feature maps of another 2 consecutive 3×3 convolutional layers that learn additional multiscale information from the re-sized input image in that resolution. At the third stage of our architecture, we utilize a series of 3 consecutive dilated residual blocks with dilation rates of 1, 2, and 4, respectively. Finally, we incorporate a dilated spatial pyramid pooling layer with 4 different dilation rates of 1, 6, 12 and 18 in order to recover the content lost in the learned feature maps during the encoding process.

Subsequently, the decoder in our architecture receives the learned multiscale contextual information of the dilated spatial pyramid pooling layer and is connected to the dilated residual units at each resolution via skip connections. In each path of the decoder, the image is up-sampled and 2 consecutive 3×3 convolutional layers are used before proceeding to the next resolution. Moreover, each scale branches to an additional convolutional layer whose output is resized to the original input image size and is followed by another convolutional layer with sigmoid activation function.

These multiscale prediction maps contribute to the final loss layer. We utilize a soft Srensen-Dice loss function as our basis and aggregate throughout each of the four resolutions:

$$Loss = \sum_{m=1}^4 \left(1 - \sum_{n=1}^N \frac{2G_n P_{n,m}}{G_n + P_{n,m} + \epsilon} \right) + \lambda \|w\|_2^2, \quad (2)$$

where N , $P_{n,m}$, and G_n denote the total number of pixels, the label prediction of pixel n in scale m , and the ground truth label of pixel n , respectively, ϵ is a smoothing constant, and λ is the weight decay regularization hyper-parameter.

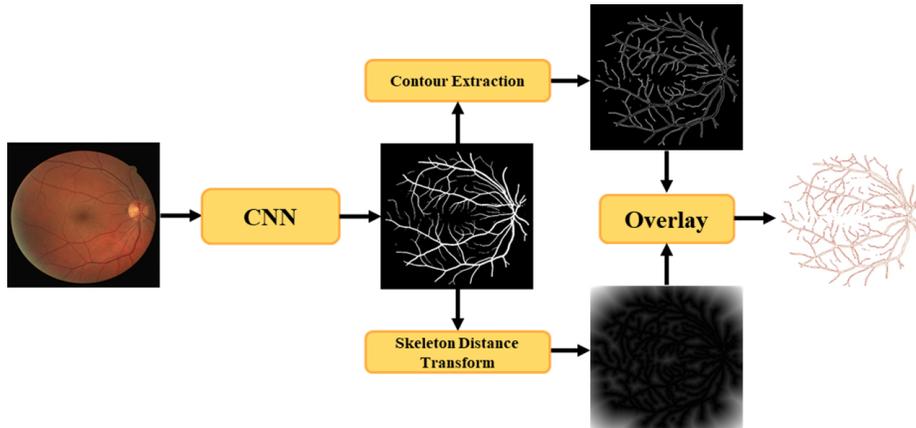


Fig. 3: Estimating the width profile of retinal vessels from segmentation.

2.2 Vessel Width Estimation

We propose a simple, yet effective method for the automatic estimation of vessel width profiles by leveraging the segmentation masks obtained by our CNN. Similar to [15], we first obtain the skeleton of the image by successively identifying the borderline pixels and removing the corresponding pixels that maintain the connectivity of the vessels. This operation approximates the center line of the vessel and represents its topology. We then calculate the distance of each pixel to the derived center-lines by applying an Euclidean distance transform to the generated feature map. Finally, we extract the contour of the original segmentation mask and overlay the generated distance transform onto this map to create the final width map of the retinal vessels. Needless to say, our formulation is only valid in areas where these vessels exist, otherwise the width value is set to zero. Figure 3 illustrates our width estimation algorithm in more detail. Unlike competing methods, our method does not rely on hand-crafted geometric equations nor on user interaction.

3 Experiments

3.1 Implementation Details

We have implemented our CNN in TensorFlow [1]. All the input images are converted to gray-scale, transformed by contrast-limited adaptive histogram equalization, resized to a predefined size of 512×512 , and intensity normalized between 0 and 1. Our model is trained, with a batch size of 2, on an Nvidia Titan XP GPU and an Intel Core i7-7700K CPU @ 4.20GHz. We use the Adam optimization algorithm with an initial learning rate of 0.001 and exponentially decay its rate. The smoothing constant in the loss function and the weight decay hyperparameter are set to 10^{-5} and 0.0008, respectively. Since the number of images

Table 1: Segmentation Evaluations on the DRIVE and CHASE-DB1 datasets.

Method	DRIVE				CHASE-DB1			
	SE	SP	Acc	F1	SE	SP	Acc	F1
Melinščak et al. [11]	0.7276	0.9785	0.9466	-	-	-	-	-
Li et al. [9]	0.7569	0.9816	0.9527	-	0.7507	0.9793	0.9581	-
Liskowski et al. [10]	0.7520	0.9806	0.9515	-	-	-	-	-
Fu et al. [4]	0.7603	-	0.9523	-	0.7130	-	0.9489	-
Oliveira et al. [12]	0.8039	0.9804	0.9576	-	0.7779	0.9864	0.9653	-
M2U-Net [8]	-	-	0.9630	0.8091	-	-	0.9703	0.8006
U-Net [3]	0.7537	0.9820	0.9531	0.8142	0.8288	0.9701	0.9578	0.7783
Recurrent U-Net [3]	0.7751	0.9816	0.9556	0.8155	0.7459	0.9836	0.9622	0.7810
R2U-Net [3]	0.7792	0.9816	0.9556	0.8171	0.7756	0.9820	0.9634	0.7928
LadderNet [16]	0.7856	0.9810	0.9561	0.8202	0.7978	0.9818	0.9656	0.8031
DUNet [6]	0.7894	0.9870	0.9697	0.8203	0.8229	0.9821	0.9724	0.7853
Ours	0.8197	0.9819	0.9686	0.8223	0.8300	0.9848	0.9750	0.8073

is limited, we perform common data augmentation techniques such as rotating, flipping horizontally and vertically, and transposing the image.

3.2 Datasets

We have tested our model on two publicly available retinal vessel segmentation datasets—DRIVE and CHASE-DB1. The DRIVE dataset consists of 40 two-dimensional RGB images with each image having a resolution of 565×584 pixels, divided into a training set and test set, each comprising 20 images. The CHASE-DB1 dataset includes 28 images, collected from both eyes of 14 children. Each image has a resolution of 999×960 pixels. We divided the CHASE-DB1 dataset into a training set of 20 images and a testing set of 8 images.

3.3 Results

We used the following metrics to measure the performance of our model: With TP , TN , FP , FN denoting true positive, true negative, false positive, and false negative, respectively, the sensitivity and specificity are given as

$$SE = \frac{TP}{TP + FN}, \quad SP = \frac{TN}{TN + FP}, \quad (3)$$

the accuracy and precision as

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}, \quad Precision = \frac{TP}{TP + FP}, \quad (4)$$

and the recall as

$$Recall = \frac{TP}{TP + FN}. \quad (5)$$

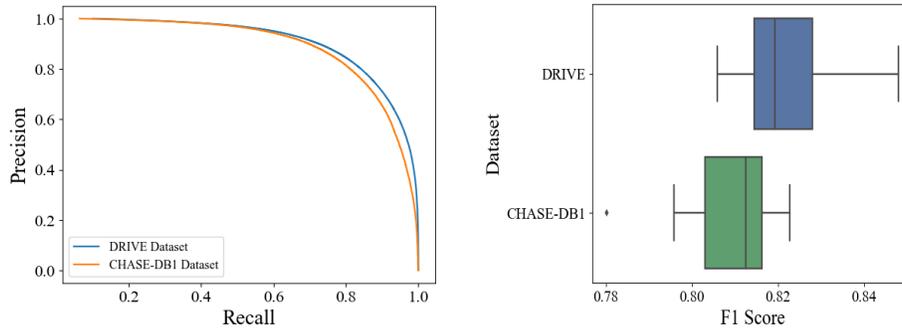


Fig. 4: Evaluation of our model’s performance on the DRIVE and CHASE-DB1 datasets.

The score

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

is the same as the Dice coefficient.

Table 1 compares to competing methods the performance of our model on the DRIVE and CHASE-DB1 datasets. On the DRIVE dataset, our model exceeds the state-of-the-art performance on the F1 score and sensitivity while being very competitive to [6] in specificity and accuracy. Figure 4 presents the precision-recall curve as well as the box and whisker plot of our model’s performance on the DRIVE dataset. The F1 scores are between 0.80 to 0.84 with no outliers falling outside the interquartile range.

In addition, for the CHASE-DB1 dataset our model exceeds the state-of-the-art performance for all metrics except specificity, for which [12] performs slightly better. The F1 scores are between 0.78 to 0.82. The precision-recall curve in Figure 4 demonstrates a similar performance on both the DRIVE and CHASE-DB1 datasets with a marginal lead for the latter dataset.

For 4 images from the DRIVE dataset and 2 images from the CHASE-DB1 dataset, Figure 5 demonstrates the final segmentation of our model compared to the masks that were manually created by clinicians as well as the generated width profiles. Clearly, our model successfully captures the intricate arteries and veins without the presence of any additional false positives as are present in the outputs of the competing methods.

4 Discussion

Our substantially improved state-of-the-art results on two publicly available datasets, DRIVE and CHASE-DB1, confirm the effectiveness of our model. Unlike the competing patch-wise approaches, our method operates on the entire image. This has proven to be beneficial as our model is significantly faster than

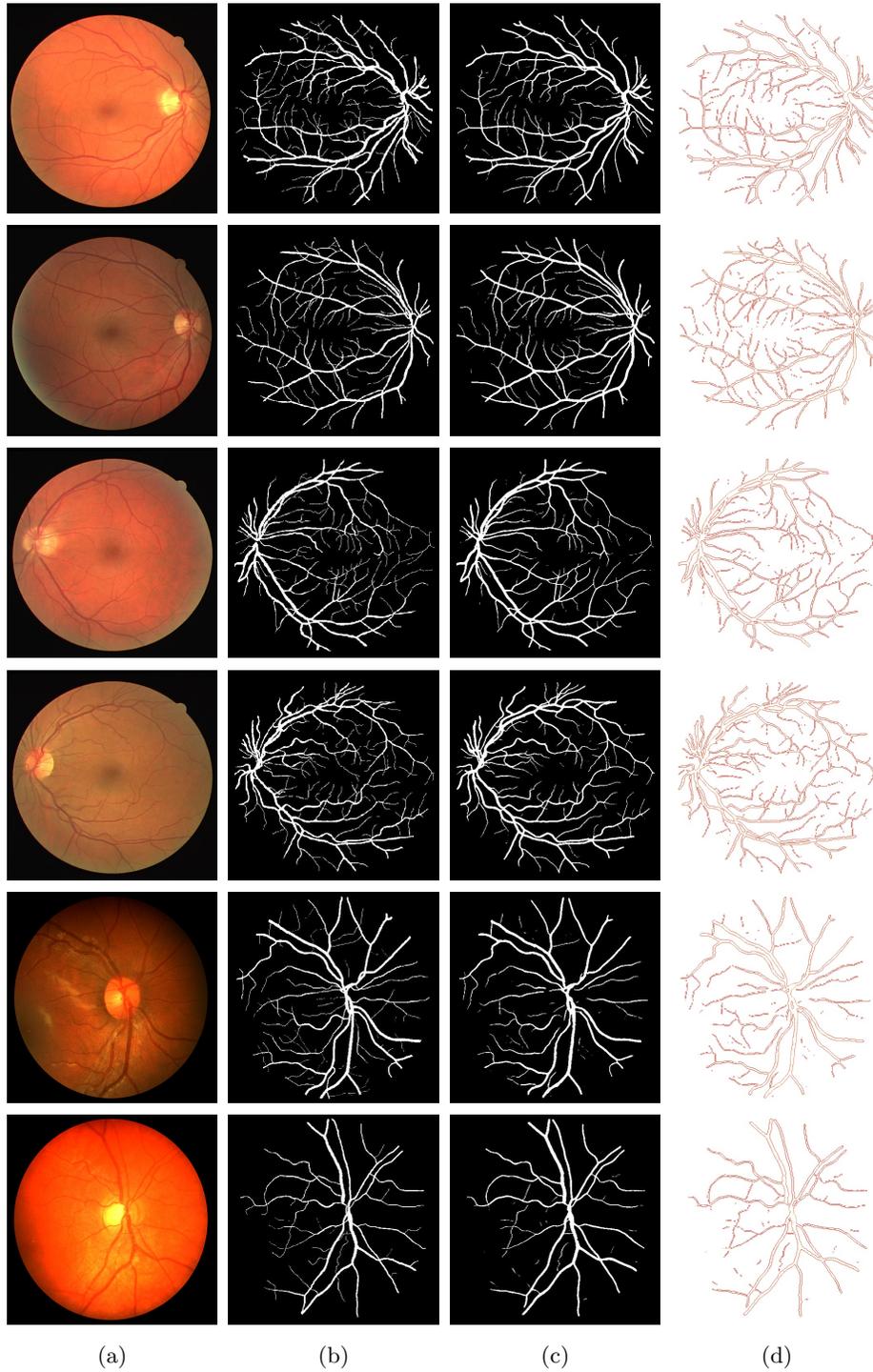


Fig. 5: (a) Input images (rows 1–4 from DRIVE; rows 5–6 from CHASE-DB1), (b) labels, (c) segmentations, (d) width estimation profiles.

the patch-wise approaches that must slide moving windows of multiple sizes over the image. Additionally, our model produces more natural and continuous segmentation masks and is able to capture finer details because it benefits from a dilated spatial pyramid pooling layer that recovers the content lost during encoding by leveraging different dilation rates and aggregating the multiscale feature maps into the decoder. Furthermore, introducing the input image at multiple scales throughout our architecture and introducing supervision at each of these scales helps our model to effectively aggregate the outputs of different stages.

Our technique for estimating the width profiles of retinal vessels by leveraging the generated segmentation masks has also proven to be effective. Its accuracy promises to help in quantifying new relevant biomarkers that correlate with the narrowing and structural changes of vessels.

5 Conclusion

We have presented a novel, fully automated method for retinal vessel segmentation and width estimation. Our deep CNN employs spatial dilated pyramid pooling and introduces the input image at multiple scales with supervision to segment retinal vessels in order to capture the smallest structural details. Our method was tested on two publicly available datasets. It has achieved better than state-of-the-art results in sensitivity and accuracy while being comparable in F1 score on the DRIVE dataset. It also achieves competitive results on the CHASE-DB1 dataset. In addition, we have introduced a method that employs the vessel segmentation maps to estimate the width profiles of retinal vessels. Such information may be very helpful to clinicians as they explore novel biomarkers and in the quantitative assessment of retinal vasculature changes associated with diseases such as diabetes and hypertension.

References

- [1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: OSDI. vol. 16, pp. 265–283 (2016) 5
- [2] Akula, A.R., Zhu, S.C.: Visual discourse parsing. arXiv preprint arXiv:1903.02252 (2019) 2
- [3] Alom, M.Z., Hasan, M., Yakopcic, C., Taha, T.M., Asari, V.K.: Recurrent residual convolutional neural network based on U-Net (R2U-Net) for medical image segmentation. arXiv preprint arXiv:1802.06955 (2018) 6
- [4] Fu, H., Xu, Y., Lin, S., Wong, D.W.K., Liu, J.: Deepvessel: Retinal vessel segmentation via deep learning and conditional random field. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 132–139. Springer (2016) 2, 6
- [5] Imran, A., Hatamizadeh, A., Ananth, S.P., Ding, X., Terzopoulos, D., Tajbakhsh, N.: Automatic segmentation of pulmonary lobes using a progressive dense V-network. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, pp. 282–290. Springer (2018), Proc. Fourth MICCAI

- International Workshop on Deep Learning in Medical Image Analysis (DLMIA 18) 2
- [6] Jin, Q., Meng, Z., Pham, T.D., Chen, Q., Wei, L., Su, R.: Dunet: A deformable network for retinal vessel segmentation. *Knowledge-Based Systems* (2019) 6, 7
 - [7] Kachuee, M., Darabi, S., Moatamed, B., Sarrafzadeh, M.: Dynamic feature acquisition using denoising autoencoders. *IEEE Transactions on Neural Networks and Learning Systems* pp. 1–11 (2018) 2
 - [8] Laibacher, T., Weyde, T., Jalali, S.: M2U-Net: Effective and efficient retinal vessel segmentation for resource-constrained environments. *arXiv preprint arXiv:1811.07738* (2018) 6
 - [9] Li, Q., Feng, B., Xie, L., Liang, P., Zhang, H., Wang, T.: A cross-modality learning approach for vessel segmentation in retinal images. *IEEE Transactions on Medical Imaging* 35(1), 109–118 (2016) 6
 - [10] Liskowski, P., Krawiec, K.: Segmenting retinal blood vessels with deep neural networks. *IEEE Transactions on Medical Imaging* 35(11), 2369–2380 (2016) 6
 - [11] Melinščak, M., Prentašić, P., Lončarić, S.: Retinal vessel segmentation using deep neural networks. In: *VISAPP 2015 (10th International Conference on Computer Vision Theory and Applications)* (2015) 2, 6
 - [12] Oliveira, A.F.M., Pereira, S.R.M., Silva, C.A.B.: Retinal vessel segmentation based on fully convolutional neural networks. *Expert Systems with Applications* (2018) 6, 7
 - [13] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: *Proc. of MICCAI*. pp. 234–241. Springer (2015) 2
 - [14] Xie, J., Zheng, Z., Gao, R., Wang, W., Zhu, S.C., Nian Wu, Y.: Learning descriptor networks for 3D shape synthesis and analysis. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018) 2
 - [15] Zhang, T.Y., Suen, C.Y.: A fast parallel algorithm for thinning digital patterns. *Communications of the ACM* 27(3), 236–239 (Mar 1984) 5
 - [16] Zhuang, J.: LadderNet: Multi-path networks based on U-Net for medical image segmentation. *arXiv preprint arXiv:1810.07810* (2018) 2, 6

Clonal Selection Algorithm Applied to Object Recognition in Mobile Robots

Jose Guillermo Guarnizo¹ and Luis Fernando Nino²

¹ Faculty of Electronic Engineering, Universidad Santo Tomas
Carrera 9 #51-11, Bogotá, D.C. Colombia.
jose.guarnizo@usantotomas.edu.co

² Intelligent Systems Research Laboratory (LISI), Universidad Nacional de Colombia
Carrera 30 No. 45-03, Edificio 453, Oficina 101, Bogotá, D.C., Colombia.
lfninov@unal.edu.co

Abstract. This paper proposes an algorithm for outline recognition in mobile robots, based on Clonal Selection algorithm, a machine learning technique of Artificial Immune Systems. The model is defined from the industrial process chain point of view, where a robot should recognize object outlines and transport them based on their geometric forms. This detection process should be also rotation and translation invariant. The robot is equipped with color sensors and laser sensors for classifying contours. The robot should have the capacity to recognize new objects, classifying those different from the existing ones. The results showed that the Clonal Selection Algorithm in a mobile robot generated antibodies for the correct classification of objects, regardless of their geometrical shape, either defined or undefined geometrical shapes, or even irregular shapes, and including objects with modified contours due to wear and tear, and white noise in the sensors. Therefore, whenever new objects were introduced to the chain process, the robot was successful trained to correctly classify them.

Keywords: Outline Recognition, Artificial Immune Systems, Clonal Selection Algorithm, Mobile Robots

1 Introduction

The application of object recognition for mobile robots increased in industrial environment, such as navigation [1], object detection or building [2]. In this way, the need for low-cost sensors has led to use lasers or sonar sensors to obtain characteristics from the object to be recognized, for example, edge detection for classifying objects [3]- [4]. In mobile robots, recognition systems need ability to adaption, robustness, and invariance to rotation and translation. In this way, evolutionary algorithms are commonly used in recognition systems, for example genetic algorithms for object recognition system in a dynamical environment [5], evolutionary algorithms applied to image segmentation [6], or emotion recognition [7]. Within the class of evolutionary algorithms, Artificial Immune Systems (AIS) present a kind of bio-inspired model

that adapts the capability of the human immune response in order to adapt the immunological response when there are different pathogens [8].

Artificial Immune Systems have been used in different industrial tasks, such as pattern recognition [9], cognitive models of behavior [10], or navigation [11]. Within the Artificial Immune Systems, Clonal Selection Theory establishes the idea that cells can recognize antigens, which are selected to proliferate [12]. In this context, different applications of Clonal Selection Algorithm have been applied in robotics, for example in UAV cooperation [13], trajectory planning in a robotic manipulator [14], coordination in swarm systems [15], or other applications data mining, medical application or classification [16]. One advantage of Artificial Immune Systems is their capacity to adapt to new conditions and their ability to learn online [17].

This paper proposes an Artificial Immune System model based on Clonal Selection Algorithm, applied to object recognition in mobile robots. For the recognition process, outline object information and Hu invariant moments are used, the latter are employed because they provide invariance in relation to the translation and the rotation of the object to be recognized. This immune model involves innate and acquired response; this model belongs to cognitive model where a multi-agent system must identify, classify, transport, and store objects. Robots have similar characteristics and can collaborate to the classification process. The model is validated in a 2D simulation environment.

This article is organized as follows. Section 2 presents the Clonal Selection Algorithm. Section 3 explains the stage conditions. Section 4 analyzes the immune model. Section 5 shows the experiment and. Finally, section 6 provides conclusions and future works.

2 Clonal Selection Algorithm

Understanding the AIS (Artificial Immune System) models involves knowing about the immune theories that inspired the AIS. In the case of the Natural Immune System, it has comprises two principal immune responses: the innate response and the acquired response.

The innate response is an innate immunity static barrier. This barrier is activated when a foreign agent (pathogen) enters the body. The innate response distinguishes between self cells and non-self cells in the body and attacks non-self cells. When non-self cells are detected, the innate system liberates proteins, called innate cells, that produce inflammation and fever [18].

As some pathogens have the capacity to avoid the innate response, vertebrates developed the acquired system. The acquired response anticipates mutations in the pathogen and then different mechanisms are activated to neutralize the pathogen attack. The antigens are molecules (usually proteins), on the surface of pathogens, and each antigen is composed of divisions called epitopes. The antigen is used to recognize the pathogen; for example, when phagocytic cells neutralize the pathogen, the phagocytes present the antigen of the specialized cells of the acquired system. B cells are specialized cells in the antigen presentation process; B cells interact with T cells, T cells

confirm the antigen, and B cells immediately begin a mutation and clonal selection process. When an antigen enters the body, it can be recognized by the immunoglobulins of B lymphocytes. If the antigen is recognized, the confirmation of T lymphocytes is necessary, and then the humoral response is activated. At this point, B cells are cloned and differentiated (cloned B cells with high affinity are cloned and reproduced again; B cells with low affinity are suppressed). T cells produce memory cells and plasma cells that generate immunoglobulins [19].

The clonal selection algorithm proposes that antigens and antibodies are presented as vector spaces. The affinity between antigens and antibodies are measured by a metric that provides a set of candidate solutions by selecting the best affinities in order to generate a new set of clones looking for antibodies with best affinities [20]. Figure 1 shows this process.

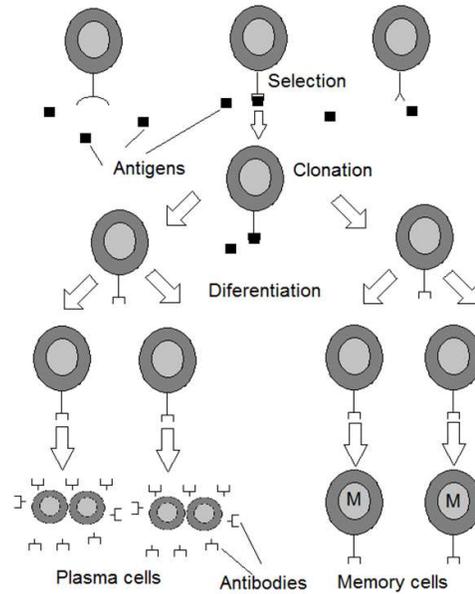


Fig. 1. Clonal selection theory.

3 Stage Conditions

Given a 2D environment, some geometrical objects must be recognized and classified by mobile robots for future transportation. For the experiment, Player Project is employed, a client-server platform, using TCP IP protocols, which provides an interface among sensors, actuators, and control response for robot control. Player Project includes a 2D simulator: Stage. This simulator provides drivers for sensors and robots, and it has the capacity to build different stages for simulations. This simulator sup-

ports C++ language. The stage is simulated in an industrial environment. Figure 2 shows an initial set of objects; these objects are called submarine, invader, phantom, circle, squarer, and triangle, respectively. They must be classified by a mobile robot.

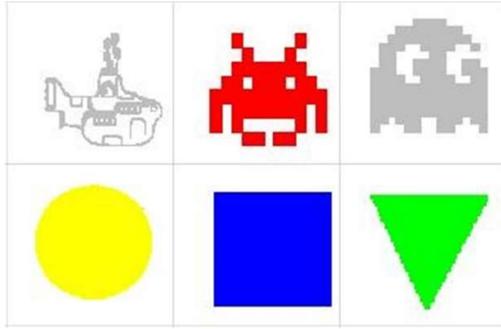


Fig. 2. Objects to be classified.

The robots have laser sensors and color sensors in their front. The laser sweeps the area, with 180 degrees around the robot, with a frequency of 180 samples per second; the range of the laser sensor is 8 meters. The laser returns a signal value corresponding to the distance of the object. For example, if there is an object two meters away, the laser returns a value of two. The clonal selection algorithm is used by the pattern recognition algorithm. Figure 3 shows the experiment stage. In this scenario, the robot looks for objects to be classified, so it must differentiate between the object (non-self objects) and walls or other robots (obstacles of the environment).

4 Object Recognition Based on Immune Model

4.1 Innate and Acquired Immune Model

The robot moves through the environment while looking for objects to be classified; it uses laser sensors and color sensors. First, the robot should distinguish between self objects (walls and other robots) and non-self objects (objects to be classified). For these works, an Artificial Immune System is proposed.

According to the theory of Innate Response, the innate system is activated when non-self pathogen is detected. When the robot detects an object using its laser sensors at a distance of 1.3 meters ahead, it will determine if that object must be identified or not, using the innate response. In this industrial stage, some environment conditions can be controlled. Consequently, self objects such as walls, obstacles, and other robots are black. Objects to be identified have different colors; these objects are considered non-self objects. The classification is made by color detection: if the color is black, the object will be considered a self object and then the robot continues the search.

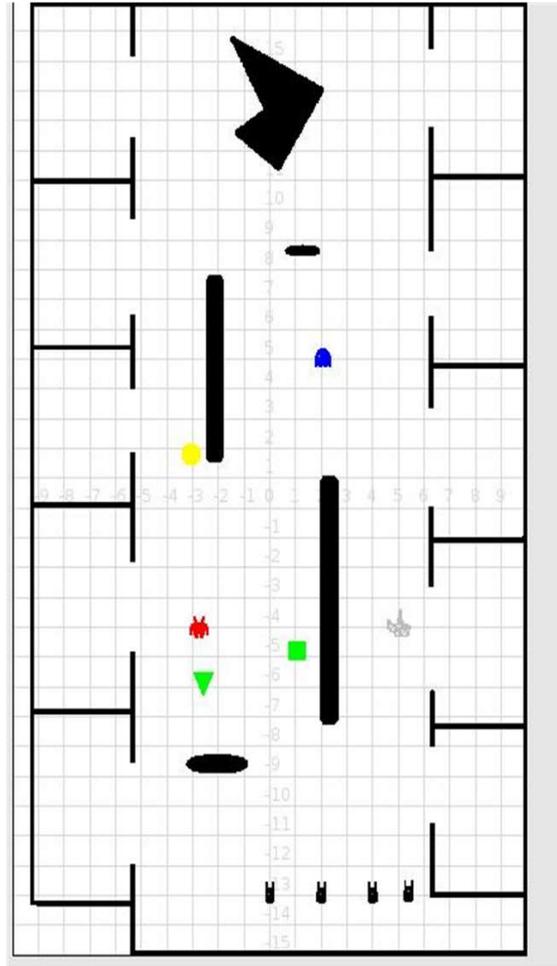


Fig. 3. Stage.

In the innate systems, macrophages can phagocytose the pathogens, applying the metaphor in the innate model. The macrophage corresponds to the color sensor and the phagocytosis process corresponds to the transportation of the non-self object without any classification process different from color recognition. Section 2 has explained that macrophages can activate the acquired response through the antigen presentation process. Based on this concept, the macrophage shows the antigen to the acquired systems, and the acquired response attacks the pathogen identified. Subsequently, the laser sensor works as antigen presenting-cell. When the color sensor detects the object as a non-self object (color different from black), the robot follows the following steps:

- Robot goes into distance of 0.7 meters in front the object.
- Robot locates in front of the object using laser information of the right and left of center.
- Robot rotates 90 degrees in clockwise direction.
- Robot turns around the object. In this act, the left laser sensor that is located to 180 degrees of the X axis of the robot, takes information of the outline of the robot.

Figure 4 shows this process.

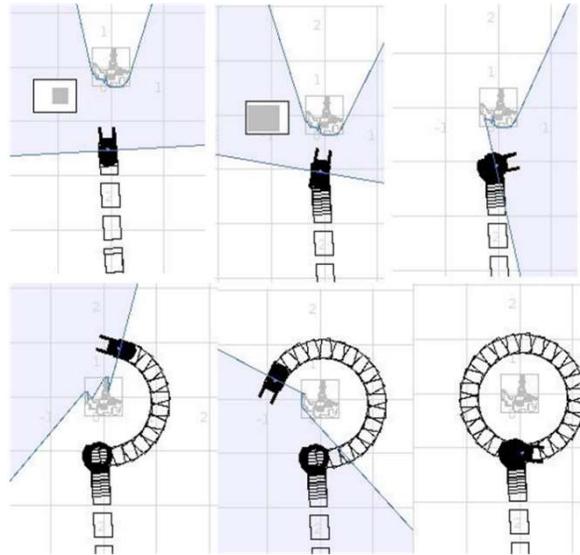


Fig. 4. Data acquisition for the classification process.

To acquire the antigen of the pathogen (non-self object to be transported), Hu invariant moments are used to obtain outline information on the object. The data obtained in the last process correspond to distance between the robot and the object. A set of the information contains between 150 and 200 different information vectors. The information on the object is translated to the outline of the object in a plane (x,y), by (1) in the axis X, and (2) in the axis Y.

$$X_c = l * \cos\theta - X_r \quad (1)$$

$$Y_c = l * \sin\theta - Y_r \quad (2)$$

X_c and Y_c are the coordinates of the outline of the object, l is the distance between the object and the robot, θ is the angle between the front of the robot and the X axis, and X_r and Y_r are the robot's coordinates. Figure 5 shows an example of the outlines obtained; these are triangle, invader, submarine, phantom, circle, and square, respectively.

The antigen is the vector that contains the Hu moments on the object to be identified. Each component of this vector corresponds to the epitope. The antibodies correspond to sets of Hu moments of the different objects previously obtained; these are used for the classification process. Each moment that composes each antibody is the paratope, in immunology paratope is defined as the region of the antibody that it used to recognize and grab the antigen [15]. The antigen is compared by a set of the antibodies stored in the robot by using equation (3).

$$Aa = K1|Mp1-Me1| + K2|Mp2-Me2| + \dots + K7|Mp7-Me7| \quad (3)$$

Aa corresponds to error between the antigen and the antibody; this error represents the difference between antigens and antibodies. Mpn is the n paratope of Hu moments stored in the robot, Men is the n epitope of the antigen, and Kn is a n scaling factor. If Aa is lower than the $U1$ threshold, the object will be considered classified. If Aa is greater than $U1$ but lower than $U2$, the object will be considered indeterminate. Finally, if Aa is greater than $U2$, the object will be considered unknown.

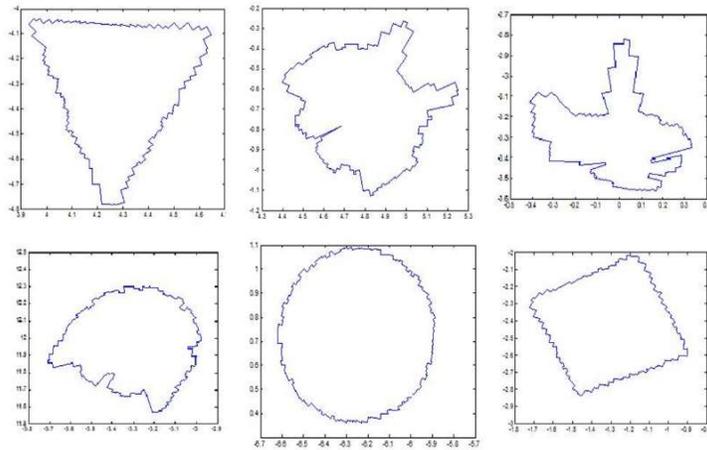


Fig. 5. Object outlines.

A set of antibodies is generated using the clonal selection algorithm. In case the error between the antibodies and the antigens is greater than $U2$, the pathogen will be considered unknown and the mechanism of generation of the set of antibodies using clonal selection algorithm will be activated. Table 1 shows the immunological metaphor. The set of antibodies stored inside the robot's memory corresponds to the immunological memory.

Table 1. Immunological metaphor.

Robotic System	Immune Inspiration
Object to Recognize	Pathogen
Hu Invariant Moments	Vector Antigen
Individual Moments	Epitope
Invariant Moments Stored in Robot	Antibodies
Individual Moments Stored in Robot	Paratope
New Object to Recognize	Antibodies Generation
Antibodies Stored in the Robot	Immunology Memory
Color Recognition	Innate Immune System
Outline Recognition	Acquired Immune System

4.2 Clonal Selection Algorithm

In order to identify an object, it is necessary to generate a set of antibodies for each object. To this end, the following Clonal Selection Algorithm is proposed:

1. A set of antigens is acquired for each pathogen (objects to be recognized). These are acquired in different directions of the object. In this process, the robot finds a non-self object in the environment (pathogen). Subsequently, the robot starts the process of antigen recognition; the agent repeats the process explained in Section 3 and shown in Figure 3. The antigen is composed of the vector of Hu invariant moments. On average, eight antigens are taken for most of the cases, rotating 23 degrees.
2. For each antigen acquired, one antibody is exactly cloned and created, which corresponds to a non-matured antibody. Each antibody is composed of the vector of Hu invariant moments.
3. The non-matured antibody is cloned in a determinate number of copies. There are ten clones for each non-matured antibody; these clones are similar to the original antibody.
4. Subsequently, each clone is muted. The mutation consists in randomly mutating each paratope $\pm 25\%$ of its value. Each paratope corresponds to each individual Hu invariant moment.
5. The error between the antibodies non-matured and their own clones mutated is calculated by solving for (3). The clones with error lower than 10% are selected as matured antibodies. These clones are included in the immunological memory of the acquired response.

5 Experiments and Results

5.1 Experiment 1

In the first experiment, ten different trajectories were performed for the six elements in order to extract their antigens by measuring the error with the non-matured antibodies acquired previously. The antibodies will be activated with errors lower than 50, this error is obtained by using equation (3), representing an absolute value of the comparison between antigens and antibodies. Table 2 shows the results obtained for each pathogen; the first column contains the object, the second column contains the numbers of antibodies with affinity lower than 50, and the third column contains the range of the errors. The error was obtained experimentally.

Table 2. Results obtained in experiment 1.

Object	Antibodies activated	Range of errors
Submarine	28 of submarine	From 8 to 42
	16 of triangle	From 18 to 50
Invader	28 of invader	From 8 to 47
	14 of circle	From 30 to 45
	1 of phantom	40
Phantom	36 of phantom	From 8 to 47
	2 of circle	From 46 to 49
	2 of square	From 46 to 48
Circle	20 of circle	From 1 to 10
	26 of invader	From 35 to 50
Square	22 of square	From 8 to 50
	1 of circle	45
Triangle	26 of triangle	From 5 to 46
	4 of submarine	From 30 to 50

In the case of the submarine, some trajectories were strongly critical because the error measured of the antibodies of submarine was similar to the triangle. In other cases of the submarine, antibodies of the triangle were activated, but submarine antibodies were more related. Only in four cases the most related antibody of submarine had an error lower than 20. For the invader, in all cases the best measured error was obtained with antibodies of invader, with an error lower than 20, except for two cases in which the errors measured were greater than 30. In the case of the phantom, in three cases the antibodies had an error lower than 20. For the circle, there is some similarity to the invader, but the error of the antibodies of the circle is strong, while the measures of the invader could be ruled out. The error of the square is convincing for the recognition; only one of other antibodies was selected, belonging to the circle, but the error obtained was not comparable with the error obtained by the square. The last case of

the triangle, the lowest error was not lower than 20 in five experiments, and in one case the error of the submarine was low, so the classification was incorrect.

5.2 Experiment 2

In order to obtain better results, the Clonal Selection Algorithm was activated and a set of matured antibodies was selected and included in the immunological memory of the agents. Once a new set of antibodies matured were trained, the conditions of experiment 1 were repeated, with ten different trajectories for each object. In this case, the antibodies activated had an error lower than 30 according to (3). Table 3 shows the results; the first column contains the object, the second column contains the numbers of antibodies cloned with an error lower than 30, and the third column shows the range of the value of errors. It is worth noting that the error was lower in experiment 2, because the matured antibodies had higher affinity with the antigens.

Table 3. Results obtained in experiment 2.

Objects	Antibodies cloned activated	Range of errors
Submarine	50 of submarine	From 4 to 30
Invader	100 of invader	From 8 to 30
	5 of circle	From 25 to 30
Phantom	40 of phantom	From 3 to 30
Circle	30 of circle	From 1 to 30
	5 of invader	From 27 to 30
Square	32 to square	From 7 to 30
Triangle	47 of triangle	From 2 to 27
	10 of submarine	From 17 to 29

In the experiments of the submarine, the antibodies of the triangle activated in experiment 1 were not activated in the new cases. The error of the antibodies selected in experiment 2 had an error lower than the error obtained in experiment 1. For the case of the invader, the errors of antibodies were lower without any doubt of the classification in all cases. For the phantom, the error values were always conclusive for the classification process; no antibodies for other pathogen were activated. In the case of the circle, although few antibodies of the invader were activated, the errors of the circle's antibodies were lower for all cases. In the experiments of the square, no other antibodies with different pathogen were selected. In all cases antibodies with errors lower than 30 were activated. For the triangle, in all cases the antibodies of triangle had robust performance, in the cases in which errors belonging to the submarine were activated, the lowest error belonging to the triangle was lower than 15, achieving correct classification in all cases.

5.3 Experiment 3

The circle, the square, and the invader were added in the environment, with noise in their outlines, as shown in Figure 6. Noise in the square is significantly higher than in the circle and in the invader. In two cases, antibodies of the square had affinity values around 70; these values were greater than U_2 , and the clonal selection algorithm was activated in order to train new antibodies for the new object. As for the circle and the invader, affinity values were lower than 20 (lower than U_1); they were classified correctly.



Fig. 6. Pathogens mutated.

The mutation in square is higher than in circle and invader. Then the antigen of square mutated enough to be considered a different pathogen. Biologically, there are some viruses that can mutate their epitopes, so the antibodies cannot detect and neutralize the pathogen [21].

5.4 Experiment 4

Finally, Kohonen maps were developed for comparison. For the experiments the six objects considered in experiments 1 and 2 were used. For the training of the Kohonen map, the Hu invariant moments obtained in previous experiments were used; those correspond to the inputs of the network. The Kohonen Network was trained with 48 networks. Once the Kohonen Map was trained, each neuron was assigned to one object. To this end, equation (4), based on (3), was used.

$$A_a = K_1|M_{par1}-M_{nk1}| + K_2|M_{par2}-M_{nk2}| + \dots + K_7|M_{par7}-M_{nk7}| \quad (4)$$

A_a corresponds to error or affinity between the neuron and the Immune System, the set M_{par} corresponds to Hu moments used by the Immune System (antibodies), and the set M_{nk} corresponds to the weights of each neuron; constants K are similar to (3). Each antibody was compared with each neuron by evaluating their affinity based on (4). If A_a is lower than 30, the object associated to this antibody will be assigned to the respective neuron. In case the A_a values between the same neuron and different antibodies associated to different objects are lower than 30, the neuron will be considered not classifiable. A similar case is when all the A_a values of one neuron are greater than 30.

Finally, the neurons were assigned as follows: to submarine, 10 neurons; to invader, 8 neurons; to phantom, 5 neurons; to square, 13 neurons; to triangle, 7 neurons; to circle, 1 neuron; and finally, 4 neurons are unclassifiable. In order to compare the Kohonen network with the Immune System, robots were used in order to classify the six objects, with similar conditions as explained in experiment 1, using both classifiers. Ten experiments per classifier were conducted by each object; Table 4 shows the results.

Table 4. Results obtained in experiment 4.

Objects	Antibodies activated.	Neurons activated.
Submarine	10 of submarine	9 of submarine 1 of triangle
Invader	10 of invader	5 of invader 3 of phantoms 1 of submarine 1 non-classifiable
Phantom	10 of phantom	6 of phantom 2 of square 2 non-classifiable
Circle	10 of circle	10 of circle
Square	10 of square	5 of square 3 of phantom 1 of circle 1 non-classifiable
Triangle	10 of triangle	6 of triangle 2 of submarine 2 non-classifiable

6 Conclusions and Future Works

An Artificial Immune System was designed for classifying objects with a mobile robot. This Immune System involved an innate response and acquired response. The innate response involved color detection, but not geometrical classification, which is used for acquired response.

For the acquired response, a set of antibodies was trained by a proposed Clonal Selection Algorithm. The antibodies showed robustness in case of noise in the classification. The noise was modeled as superficial deformation. The AIS was adaptable to new pathogens and their new antibodies generated were stored in the immunology memory of the robot.

Clonal Selection Algorithm was used for the maturation of antibodies by mutation and cloning. The performance of the acquired system classification, after maturing the clones, improved the affinity values of. A lower threshold for activation could also be assigned. When there were untrained antigens in the AIS, the immune system was

able to detect them as unknown pathogens, and a new set of antibodies was trained and stored in the immunological memory.

In future works Artificial Immune Systems will be proposed for machine learning applied to 3D object recognition using kinetic and real sense in mobile robots.

Acknowledgements. This work has been funded by “Decimotercera convocatoria interna para el Fomento de la Investigación - FODEIN 2019” at Universidad Santo Tomás, Bogotá Colombia, entitled “Mapeo, Localización Y Planeación De Trayectorias En Ambientes Interiores Aplicado A Robots Móviles Para Las Sedes De La Universidad Santo Tomás”, project code: 1936005.

References

1. Guo, S., Diao, Q., & Xi, F. (2017). Vision Based Navigation for Omni-directional Mobile Industrial Robot. *Procedia Computer Science*, 105, 20-26.
2. Schou, C., Andersen, R., Chrysostomou, D., Bøgh, S., & Madsen, O. (2018). Skill-based instruction of collaborative robots in industrial settings. *Robotics and Computer-Integrated Manufacturing*, 53, 72-80.
3. Llanos Neuta, N., Aponte Vivas, S., Velandia Fajardo, N., Rodriguez Giraldo, O., & Romero Cano, V. (2018). Low-cost recognition and classification system based on LIDAR sensors. *IEEE 2nd Colombian Conference on Robotics and Automation (CCRA)*. Barranquilla, Colombia.
4. Lee, S.-J., Lee, K., & Song, J.-B. (2014). Development of advanced grid map building model based on sonar geometric reliability for indoor mobile robot localization. *11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. Kuala Lumpur, Malaysia.
5. Wei, H., & Tang, X.-S. (2015). A Genetic-Algorithm-Based Explicit Description of Object Contour and its Ability to Facilitate Recognition. *IEEE Transactions on Cybernetics*, 15(11), 2558-2571.
6. Chouhan, S., Kaul, A., & Singh, U. (2019). Image Segmentation Using Computational Intelligence Techniques: Review. *Archives of Computational Methods in Engineering*, 26(3). 533–596.
7. Boubenna, H., & Lee, D. (2018). Image-based emotion recognition using evolutionary algorithms. *Biologically Inspired Cognitive Architectures*, 24, 70-76.
8. Nikhil, S., Semwal, T., & Nair, S. (2016). Immuno-inspired behaviour adaptation in Multi-Robot Systems. *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Budapest. Hungary.
9. Wang, W., Gao, S., & Tang, Z. (2008). A Complex Artificial Immune System. *2008 Fourth International Conference on Natural Computation*. Jinan, China.
10. Akram, M., & Raza, A. (2018). Towards the development of robot immune system: A combined approach involving innate immune cells and T-lymphocytes. *Biosystems*, 172, 52-67.
11. He, T., Zhang, Y., Sun, F., & Shi, X. (2016). Immune optimization based multi-objective six-DOF trajectory planning for industrial robot manipulators. *12th World Congress on Intelligent Control and Automation (WCICA)*. Guilin, China.

12. de Castro, L., & Von Zuben, F. (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3), 239-251.
13. Wang, Y., Zhang, W., & Li, Y. (2016). An efficient clonal selection algorithm to solve dynamic weapon-target assignment game model in UAV cooperative aerial combat. 35th Chinese Control Conference (CCC). Chengdu, China.
14. Jeronimo, D., Borges, Y., & Coelho, L. (2010). Clonal Selection Algorithm with Oppositional Approach Applied to Trajectory Planning of a Robotic Manipulator. Eleventh Brazilian Symposium on Neural Networks. Sao Paulo Brazil.
15. Weng, L., Liu, Q., Xia, M., & Song, Y. (2014). Immune network-based swarm intelligence and its application to unmanned aerial vehicle (UAV) swarm coordination. *Neurocomputing*, 125, 134-141.
16. Luo, W., & Lin, X. (2017). Recent advances in clonal selection algorithms and applications. *IEEE Symposium Series on Computational Intelligence (SSCI)*. Honolulu, HI, USA.
17. Tan, Y., Mi, G., Zhu, Y., & Deng, C. (2013). Artificial immune system based methods for spam filtering. *IEEE International Symposium on Circuits and Systems (ISCAS2013)*. Beijing, China.
18. Sakai, R., Kitano, E., Maeda, A., Lo, P.-c., Eguchi, H., Watanabe, M., Nagashima, H., Okuyama, H., Miyagawa, S. (2017). Studies of innate immune systems against human cells. *Transplant Immunology*, 40, 66-71.
19. Nino Vasquez, L., Munoz Mopan, F., Prieto Salazar, C., & Guarnizo, J. (2009). Applications of Artificial Immune Systems in Agents. In *Handbook of Research on Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies* (pages 99-122). IGI Global.
20. Pang, W., & Coghill, G. (2007). Modified clonal selection algorithm for learning qualitative compartmental models of metabolic systems. *GECCO '07 Proceedings of the 9th annual conference companion on Genetic and evolutionary computation*. London, United Kingdom.
21. Carr, A., & Penny, R. (1997). Human immunodeficiency virus infection and acquired immunodeficiency syndrome. In *Clinical immunology* (pages 28-58). Oxford, New York: Oxford University Press.

Customized use of Expectation Maximization for Glioma Identification from Brain MRIs

Nidhi Gupta¹, Pushpraj Bhatele², and Pritee Khanna¹

¹ Indian Institute of Information Technology, Design and Manufacturing, Jabalpur, India

² NSCB Medical College, Jabalpur, India

{nidhi.gupta, pkhanna}@iiitdmj.ac.in, drprbhatele@gmail.com

Abstract. This work is aimed to provide an automated and accurate computer diagnosis system to assist radiologists. The key idea is to combine output of statistical model with the local structure of pixels by using centralized patterns of run length on brain tumor region. The customization is applied by combining local characteristics of tumors on the segmented output obtained from the use of expectation maximization (EM) theorem. The incorporation of this idea resulted in efficient and effective outputs when testing is performed on two datasets. The proposed system observes 99.12% accuracy for tumor classification on a dataset named as JMCD which is collected from a prestigious local hospital. The observed accuracy on the publically available Brain Tumor Segmentation Challenge (BRATS) dataset is 100%. Four frequently found gliomas are targeted and identified through the proposed approach. The observed accuracies for glioma identification lies in the range of 95.41% to 100% for these datasets. The robustness is provided by 10-fold cross validation and results are validated by domain experts as well through statistical testing.

Keywords: Brain Tumor · Image Segmentation · Expectation Maximization · Glioma Identification.

1 Introduction

Survey reports reveal that brain tumors are placed second among death reasons [1] [2]. It was expected that 78,980 new cases of primary malignant and non-malignant brain, and other central nervous system (CNS) tumors would be diagnosed in the United States in 2018 [1]. Also, it was expected that 16,616 deaths will be attributed to primary malignant brain and other CNS tumors at the end of 2018 in US. In another expected scenario, 3,560 new cases of brain and other CNS tumors would be diagnosed in childhood. Therefore, efforts are required to detect brain tumors at an early stage.

Radiologists and clinicians examine patient health history, symptoms, and brain scans for diagnosis. MRI scans are found more suitable for classification

and identification of brain tumors [3]. MR scans hold different kind of image intensities in different image pulse sequences. This helps to visualize tumor characteristics under different parameters and conditions. Tumors frequently found in glue (supportive) cells are known as gliomas [4]. Identification of gliomas based on their pathological characteristics is a challenging task. The appearance of each glioma is quite similar, but a few inherent characteristics differentiate them individually. Among these, astrocytoma circumscribed in group and tends to respect anatomic strong boundaries. Other gliomas such as ependymoma, oligodendroglioma, and glioblastoma multiformes are found more infiltrating because of their tendency to invade. Ependymomas are found mostly at the central location of brain and have well-defined margins. Oligodendrogliomas are insidious, slowly growing, and most commonly arise in the cerebral hemispheres of middle aged adults. They traverse by a delicate capillary network and possess a tendency to calcify, that is helpful to radiologists for their histological diagnosis [4]. On the examination by naked eyes, glioblastoma multiformes have poorly defined intra axial mass with variegated (multiforme) appearance due to necrosis and hemorrhage on brain.

This work proposes an automatic and efficient computer aided diagnosis (CAD) system for detection and identification of glioma tumors from T2-weighted pulse sequences of any of the three views of brain MRIs, i.e., sagittal, axial, and coronal. Novelty of this work lies with the customized use of expectation maximization (EM) theorem for brain tumor segmentation and detection. Extracted run length of centralized patterns (RLCP) from these segmented regions are classified through naive Bayes (NB) classifier. Outcome of the proposed system on two datasets are validated by domain experts as well.

The work is organized as follows. Section 2 includes discussion on the existing techniques. The proposed method is described in Section 3 with the detailed description of each step. Results are discussed in Section 4. Comparison with the existing techniques is given in Section 5 and the work is concluded in Section 6.

2 State of the Art

Various statistical, probabilistic, and computational models are proposed for diagnosis of abnormalities through medical scan images. Segmentation of the compulsive regions such as tumor and edema in MRI is a challenging task due to associated uncertainties like proper location, shape, size, and texture properties of tumor. These characteristics are not static and hence, result in increased difficulties for radiologists and clinicians to diagnose tumor at an early stage. The existing studies are useful, yet an automatic and accurate detection of tumor is a challenging task.

Different techniques are used in the existing systems for preprocessing, segmentation, feature extraction, and classification [3]. The issues such as noise reduction, enhancement, skull stripping, smoothing, and sharpening of boundaries are considered under preprocessing. Clustering algorithms like, watershed [5], k-means, fuzzy c-means [6], region growing, threshold [7], and level-set method

[8] are used for segmentation purpose. Features like discrete wavelet transform (DWT) coefficients [9], Gabor [10] [11], Zernike moments [12], pyramid of histogram of gradients [13], local binary patterns [14], gray level co-occurrence matrix [15], morphological properties [5], and run length of centralized patterns [16] are extracted for further classification. Well-known classifiers like support vector machine (SVM), k-nearest neighbour, NB, and random forest are used in the classification step.

Feature extraction through spatial information and its classification through Markov random field is used by [17], but only 75% similarity index was achieved. Multiscale diffusion filtering scheme for the construction of multiscale imaging and fuzzy c-means for the detection of tumors were used to observe 88% overlap ratio [18]. Zernike moments were explored to identify brain tumors using ANN classifier [12]. Normalized distance of 13.89 was observed for the segmented tumors from the ground truth. Centroid coordinates were used for tumor detection to achieve 92% accuracy [19]. DWT and feed forward back-propagation neural network were explored to classify brain tumor images to achieve 99% accuracy [3]. The anisotropic diffusion and DWT used with SVM classifier resulted in 99.78% accuracy for tumor detection [6]. Mutual information between histograms of two brain hemispheres was compared to detect tumors [11]. Gabor wavelets and statistical features were extracted and classified by SVM to observe 97.40% accuracy. In another work, block based segmentation was used to detect the tumor and classified by threshold to achieve 97.93% accuracy [20]. Also, fuzzy logic was explored for the segmentation of tumors. Various shape and texture features were extracted and classified through NB classifier resulted in 91% accuracy [5]. In a similar work, fuzzy k-means with hybrid self-organizing map for segmentation was used to observe 96.18% accuracy [21]. A technique using superpixels and segmented images by extremely randomized trees reported 91% dice overlap ratio at maximum [22]. In another work, background intensity compensation was applied and localized active contour model was developed to segment tumors with 91.02% dice coefficient [23].

The techniques discussed so far detected or segmented tumors. A work addressed identification of astrocytoma, meningioma, metastatic bronchogenic carcinoma, and sarcoma tumors using GLCM features and feed forward neural network classifier to achieve 97.50% accuracy [15]. Astrocytoma and its types, e.g., low grade, pilocytic, anaplastic, and glioblastoma tumors were classified and 98.67% accuracy was observed [24]. Also, tumor was identified and classified into low grade and high grade astrocytomas in [5]. In another approach, image fusion followed by adaptive threshold was used to segment gliomas and RLCP features were explored. NB classifier was applied to classify images to achieve 97.83% and 96.47% accuracies on JMCD and BRATS datasets, respectively [16].

The study involves many challenges. Dataset acquisition is a challenging task and most of the datasets used are either biased or of small size. Segmentation plays a major role in the identification of tumor location and morphological properties like area, size, and location etc. Sometimes existing techniques are dataset dependent or biased towards the type of input MR sequences. Although

application of textural and shape features proved quite efficient for brain tumor detection, still these are yet to explore for categorization of tumors, e.g., identification of gliomas. Multiclass classification also requires attention to proceed accurately. This study attempts to address these issues by providing improvements over existing techniques. The proposed approach is tested on a larger size dataset as well as a publicly available dataset. Further, different kind of gliomas are identified with the morphological properties. Manual intervention may result in an unavoidable error in terms of accuracy of the analysis, segmentation, classification, and identification of the tumor. The proposed work is aimed to develop an automated and accurate CAD system based on T2-weighted pulse sequences of any of the three views. This generalization is proved quite effective and efficient as compared to other existing systems. EM has proved as an efficient tool for determining incomplete and missing labels of pixels. Employment of EM theorem in iterative manner solves the approximation problem efficiently. The repetition till convergence maximizes the posterior margin of segmented regions. The proposed system is able to classify four types of gliomas from brain MRIs by customizing the output of EM algorithm and classifying RLCP features (earlier proposed by authors for tumor detection) through NB classifier. The system is examined on two datasets and the results obtained with 10-fold cross-validation showcase its robustness. Outcomes of the system are also verified by domain experts.

3 The Proposed Approach

Typical components in the basic pipeline of a CAD system are discussed in the following subsections.

3.1 Dataset Acquisition and Preprocessing

Experiments are performed on two datasets. A dataset referred as JMCD is collected during 2014-2017 from a prestigious local government medical college, namely, Netaji Subhash Chandra Bose Medical College, Jabalpur. It contains MRI scans of 140 patients with 200-250 scanned MR images of each patient. JMCD contains images of four types of glioma tumors; namely astrocytoma, ependymoma, oligodendroglioma, and glioblastoma multiformes, along with non-tumorous images. Another publicly available standard dataset, Brain Tumor Segmentation Challenge (BRATS) [25], used in this work consists two types of gliomas (low grade and high grade) along with non-tumorous images. Both the datasets consist of all four kinds of image pulse sequences (T1-weighted, T2-weighted, T1-post contrast, and Fluid Attenuated Inversion Recovery). Tumorous region possess brighter intensity than normal tissues (hyper-intense) on T2-weighted MR scan images, hence it is most preferred by the radiologists, clinicians, and medical scientists for easy prognosis of brain tumors [26]. Due to this fact, only T2-weighted pulse sequence from all three views are focused in this work.

Preprocessing includes enhancement to accelerate fast and accurate prognosis of tumorous region. Poor contrast and inhomogeneous MRIs are enhanced by combining anisotropic diffusion (AD) with dynamic stochastic resonance (DSR) in DWT domain [27]. AD deals with the sharpening of boundaries and edges present in the image, while DSR increases mean and variance of the image. Prominent parts of the image become more visible on the application of AD. Entropy based threshold is used to select remaining dark areas for enhancement. The iterative application of DSR on LL bands of these regions further enhances intensity values of dark pixels. DSR uses inherent noise of the regions, and modifies mean and variance of the image in order to enhance brightness and contrast of the image, respectively. Iteration stops after meeting the standard criteria of quality indexed parameters [28] [29] [30]. This technique significantly enhances MRIs without introducing any artifacts.

3.2 Segmentation: EM Theorem

Expectation-maximization/maximization of posterior marginals (EM/MPM) is one of the proven statistical models for segmentation in image processing [13]. EM theorem is typically used for clustering task to provide labels to underlying missing data. It measures the expected values using current estimates of different parameters and applies conditions on the variables. Maximization step measures maximum likelihood and determines the label of missing data with minimum error. This kind of traditional statistical model involves latent (unknown) variables and known observations. The maximum likelihood is estimated by calculating derivatives of likelihood function with respect to all unknown values. The zero variance and mean parameters determine convergence of the mixture model. Derivative converges to zero states that maximum value is achieved [13] [31] [32]. Image segmentation task is accomplished by measuring maximum likelihood for the given parameters and analyze it for missing pixel labels. The log-likelihood function deals with continuous and differentiable parameters of image pixels. EM algorithm is then used to estimate the parameters of data model, which retains maximum likelihood. Further, mean and variance are measured from the extracted regions. Through this algorithm, the label of a particular pixel is estimated iteratively in the expectation step. In this way, maximum probability is observed and label is assigned to the misclassified and unlabeled pixel.

Let image pixels are denoted as $x_1, x_2, x_3 \dots x_{MN}$ and MN is the total number of pixels in the image [13] [31] [32]. The statistical model works on two assumptions: (1). Random variables Y_1, Y_2, \dots, Y_N are conditionally independent given the pixel label field X . (2). Conditional probability density function (PDF) of Y_r , given X depends only on the value of X at pixel location r . Using these two conditions, the conditional PDF can be written as

$$f_{Y|X}(y|x, \theta) = \prod_{r=1}^{MN} f_{Y_r|X}(y_r|x_r, \theta) \quad (1)$$

where θ is a non-random vector whose elements are unknown parameters of the conditional PDF of Y given X . Mean and variance of Y_r depend on the class to which pixel r belongs. Thus, all random variables in Y which represent class i , for any $i = 1, 2, \dots, L$ are independent and identically distributed Gaussian random variables with mean and variance which are elements of parameter vector θ . Using this, the conditional PDF of Y given X is

$$f_{Y|X}(y|x, \theta) = \prod_{r=1}^{MN} \frac{1}{\sqrt{2\pi\sigma_{x_r}^2}} \exp - \frac{(y_r - \mu_{x_r})^2}{2\sigma_{x_r}^2} \quad (2)$$

The conditional probability is mass function of X given Y to segment the image. This problem is formulated as an optimization problem. The optimization criteria are used as minimization of expected value of the number of misclassified pixels. It is clear that minimizing the expected value is equivalent to maximizing $P(X_s = k|Y = y)$ over all $k \in 1, 2, \dots, L, \forall s \in S$. In this model, EM and MPM algorithms are combined to get segmented images. Expectation and maximization steps are iteratively performed [13]. Solving both of these steps provides

$$\mu_k(p) = \frac{1}{N_k(p)} \sum_{s=1}^{MN} y_s p_{x_s|Y}(k|y, \theta(p-1)) \quad (3)$$

and

$$\sigma_k^2(p) = \frac{1}{N_k(p)} \sum_{s=1}^{MN} (y_s - \mu_k(p))^2 p_{x_s|Y}(k|y, \theta(p-1)) \quad (4)$$

where

$$N_k(p) = \sum_{s=1}^{MN} p_{x_s|Y}(k|y, \theta(p-1)), k = 1, 2, \dots, L. \quad (5)$$

The minimum mean and variance subjected to one strong segmented region, whose probability is measured in the maximization step, are accomplished. The inter-region margin is maximized and intra-region margin is minimized. The entire algorithm can be summarized into two steps. At first, initial estimates of θ and X are selected. Then, for $p = 1, 2, \dots, P$, stage p of the algorithm consists of two steps: (a). Perform T_p iterations of MPM algorithm using $\theta(p-1)$ as the value of θ . (b). Use EM update equations of θ to obtain $\theta(p)$, using the values as estimates of $p_{x_s|Y}(k|y, \theta(p-1))$. Afterwards, $\theta(P)$ is obtained as final estimate of θ . T_{P+1} iterations of MPM algorithm are performed using $\theta(P)$. The final segmented image is $X(P+1; T_{P+1})$.

Customized Post Processing on the Segmentation Results: Segmentation results obtained with the procedure discussed above are customized to get a better and fast estimation in this context. Brain tissues are segmented into four basic regions, i.e., background pixels, GM, CSF, and WM. Only CSF region is targeted here, as it provides the most accessible index of any abnormality present on brain tissues. It is noted that numerous scattered regions are present in the

CSF region. Sometimes extraction of the largest blob does not necessarily produce the meaningful region. A few inherent characteristics of extracted regions like area, perimeter, orientation, solidity etc. may help to identify meaningful region required for tumor detection. Among these properties, area and perimeter are the most significant and relevant features having high discriminating power as they signify growth and aggressiveness of the tumor, respectively. These inherent parameters enhance the performance of categorization task and help in the identification of tumor. Therefore, two prior conditions are developed with area and perimeter of extracted regions. At first, area of the extracted regions is measured by counting the number of pixels present in it and the region having largest area is identified. Secondly, the perimeter of the largest region is calculated. It is noted that the perimeter of interested region is ranged [100 1000]. The maximum limit of perimeter touches to 10^3 . It is also noted that if perimeter is higher than 10^3 , then most probably it would represent skull region or a normal tissue only. If perimeter of the largest region falls in the range [100 1000], then it is identified as the meaningful region required for further processing. Otherwise, second largest region (area-wise) is considered and its perimeter is checked for the above mentioned range. If its perimeter falls within limit, then the region is considered for further processing. Otherwise the same process is repeated till a meaningful region is identified. In this way, these two post-processing conditions are applied to get meaningful segmented region. The binary image is mapped to its gray level image to extract segmented region. Figure 1 illustrates the extracted regions. Background region is not shown here.

3.3 Feature Extraction and Classification

RLCP features earlier developed by us are extracted from segmented images [16]. Sometimes binary patterns are not sufficient to differentiate normal and abnormal tissues. Run lengths of centralized patterns can be used to differentiate spatial distribution of pixels and their local grid structure. RLCP associates GLRL matrices with LBP patterns to provide efficient texture features.

First of all, LBP codes corresponding to the original image are obtained. Afterwards, LBP code is indexed and eleven GLRL matrices [30] in four principal directions are formed to count occurrences of run lengths for each gray level. Let $C(i, j)$ is a RLCP matrix, which denotes run-length matrix for indexed LBP image. $C(i, j)$ defines the number of runs with pixels of gray level i and run length j [33]. For this run length matrix $C(i, j)$, let M_R is the number of gray levels, N_R is the maximum run length, n_r is the total number of run lengths, and n_p is the number of pixels in the indexed LBP image. Based on $C(i, j)$, eleven run length matrices, namely short run emphasis, long run emphasis, gray-level non-uniformity, run length non-uniformity, run percentage, low gray-level run emphasis, high gray-level run emphasis, short run low gray-level emphasis, short run high gray-level emphasis, long run low gray-level emphasis, and long run high gray-level emphasis are calculated [30] [33].

Short run emphasis measures the distribution of short runs in the indexed LBP image and expected to be high for fine textures. Long run emphasis (LRE)

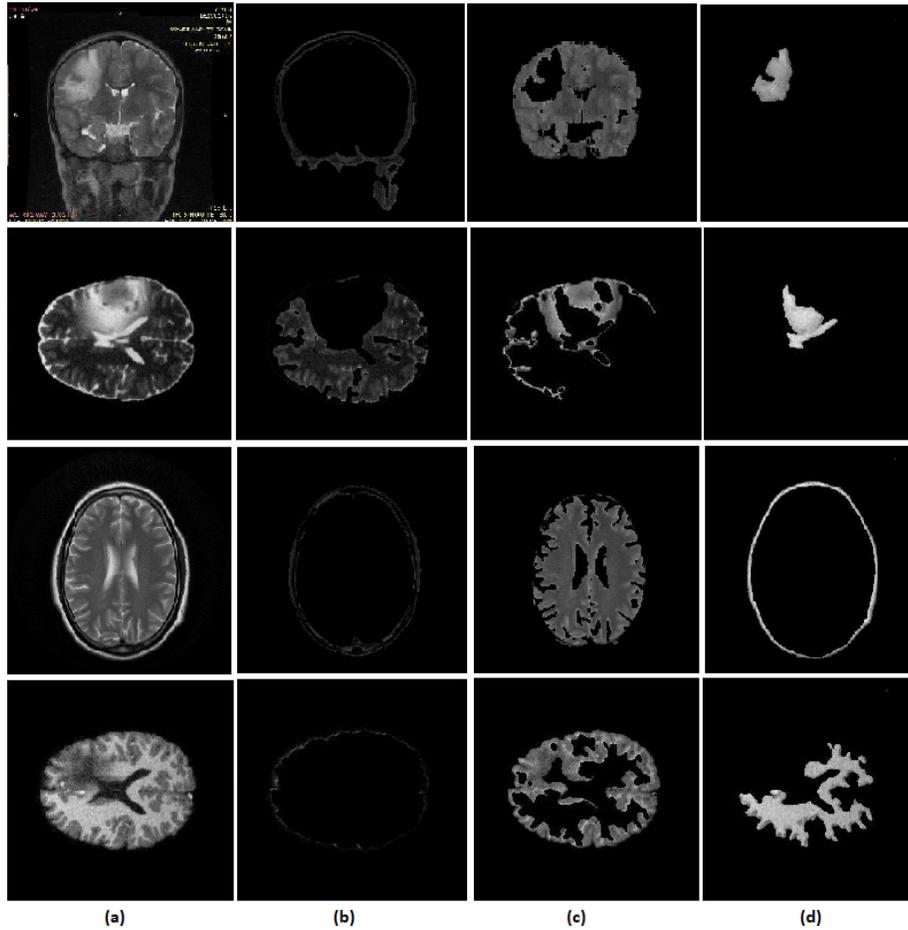


Fig. 1. Results of model based segmentation (first two rows are from JMCD and the last two rows are from BRATS) (a) Original image, (b) WM, (c) GM, and (d) CSF.

measures the distribution of long runs in the indexed LBP image and expected to be high for coarse structural textures. Gray-level non-uniformity measures the similarity of gray level values throughout the indexed LBP image. It is low if the gray levels are alike throughout the image. Run length non-uniformity measures the similarity of the length of runs throughout the indexed LBP image. It is low if the run lengths are alike throughout the image. Run percentage measures the homogeneity and the distribution of runs of indexed LBP image in a specific direction. It is highest when the length of runs is 1 for all gray levels. Low gray-level run emphasis measures the distribution of low gray level values. It increases when the texture is dominated by many runs of low gray values. High gray-level run emphasis measures the distribution of high gray level values. It

increases when the texture is dominated by many runs of high gray values. The next four features measure the joint distribution of run length and gray level. Short run low gray-level emphasis measures the joint distribution of short runs and low gray level values. Short run high gray-level emphasis measures the joint distribution of short runs and high gray level values. Long run low gray-level emphasis measures the joint distribution of long runs and low gray level values. Long run high gray-level emphasis measures the joint distribution of long runs and high gray level values.

The complete process of RLCP feature extraction for LRE feature is shown in Figure 2. A small portion of generated LBP image shown in Figure 2(a) is indexed in Figure 2(b) [19]. A run length matrix $C(i, j)$ in the horizontal direction is generated for this indexed LBP image as shown in Figure 2(c). Finally, LRE feature is calculated using this matrix. Following this approach, eleven RLCP features are obtained in all four principal directions, i.e., 0° , 45° , 90° , and 135° . This gives extracted feature vector of length 44. The reason behind fast extraction of RLCP features lies in its operation on the local centralized gray pattern image instead of its gray values. The emphasis on centralized patterns strengthens the power of differentiation of tissues. Further, the use of a small number of RLCP features also favours faster execution of the proposed CAD system. The extracted features are classified through powerful NB classifier which builds up the model based on the probability and predicts the missing label on maximal basis. The relative advantage of NB classifier over other classifiers is discussed in [16].

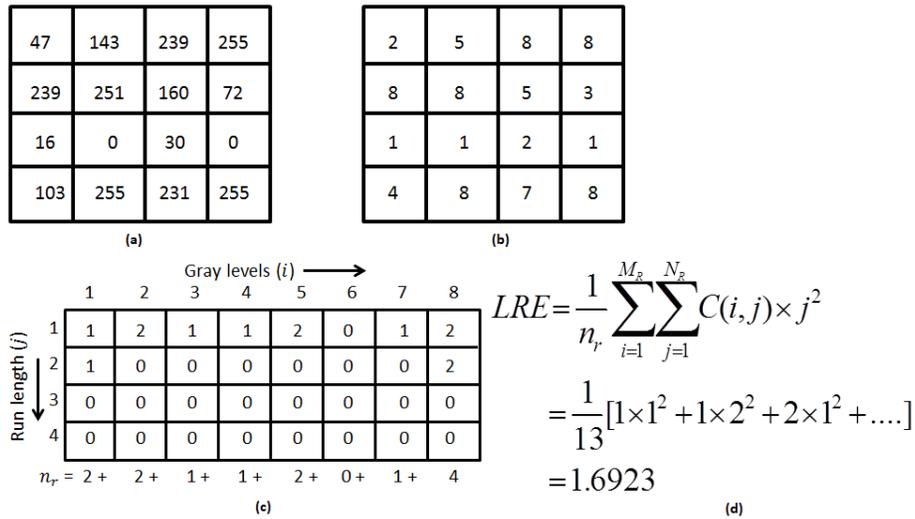


Fig. 2. Extraction of RLCP features (a) LBP image, (b) Indexed LBP image, (c) GLRL matrix in the horizontal direction, and (d) LRE feature extraction [16].

4 Results and Discussion

Experiments are performed in MATLAB[®] R2013a environment on Windows 7 platform with 2.83GHz processor and 4GB RAM. The performance of segmentation is evaluated on well-known parameters and observations are given in Table 1. Classification performance of the proposed CAD system is examined and observations are presented in Table 2. Performance of the proposed system as compared to the existing approaches is summarized in Table 3.

4.1 Segmentation Assessment

Segmentation analysis is done on some well established parameters, namely, peak signal-to-noise ratio (PSNR), mean square error (MSE), Hausdroff distance (HD), Jaccard index (JI), and disc similarity coefficient (DSC) [34][35][21]. PSNR and MSE are quality index measurements of segmented images. HD is the largest distance among of all the distances from a point in the segmented image and ground truth. JI and DSC are measurements of similarity between the segmented images and ground truths. As given in Table 1, higher values for PSNR, JI, and DSC and lower values of MSE and HD for both the datasets reflect significant segmentation performance.

Table 1. Segmentation Assessment.

Datasets / Parameters	PSNR	MSE	HD	JI	DSC
JMCD	58.13	0.10	10.23	0.89	0.91
BRATS	56.36	0.15	13.06	0.92	0.86

4.2 Performance Assessment

Using JMCD and BRATS datasets, performance of the proposed system is examined on various well-established classification rates like, accuracy, precision, sensitivity, and specificity. These are defined as

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (6)$$

$$Precision = TP/(TP + FP) \quad (7)$$

$$Sensitivity = TP/(TP + FN) \quad (8)$$

$$Specificity = TN/(TN + FP) \quad (9)$$

where TP, TN, FP, and FN stands for True Positive, True Negative, False Positive, and False Negative cases respectively. The observed values for glioma identification are given in Table 2. On JMCD, average accuracy achieved for tumorous MRIs is 96.45% and for non-tumorous MRIs it is observed as 96.38%. Among

gliomas, astrocytomas are more accurately identified in comparison to other tumors. As discussed earlier also, it contains anatomically strong boundaries. On BRATS, 97.50% accuracy is observed for tumorous images and 100% is observed for non-tumorous images. The system works equally well in identification of low grade and high grade gliomas. Two screenshots of the proposed system on JMCD dataset are shown in Figure 3.

Table 2. Performance of the proposed CAD system for glioma identification on JMCD and BRATS datasets (in %).

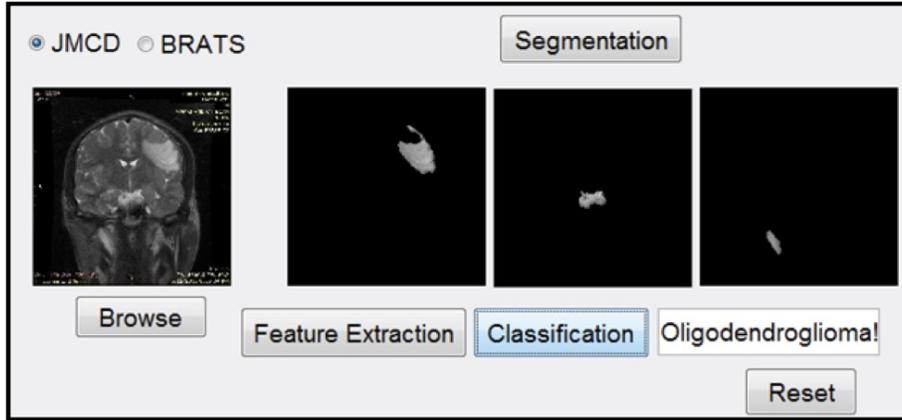
Dataset	Tumor Identification	Accuracy	Precision	Sensitivity	Specificity
JMCD	Astrocytoma	96.97	96.33	95	98.64
	Ependymoma	96.53	100	100	98.54
	Oligodendroglioma	95.41	95	96.31	99.05
	Glioblastoma Multiforme	96.92	99.26	93.15	98.83
	Non Tumorous	96.38	97.02	99.87	96.90
BRATS	Low grade	97.50	100	96.67	100
	High grade	97.50	96	100	96
	Non Tumorous	100	100	100	100

4.3 Performance Validation

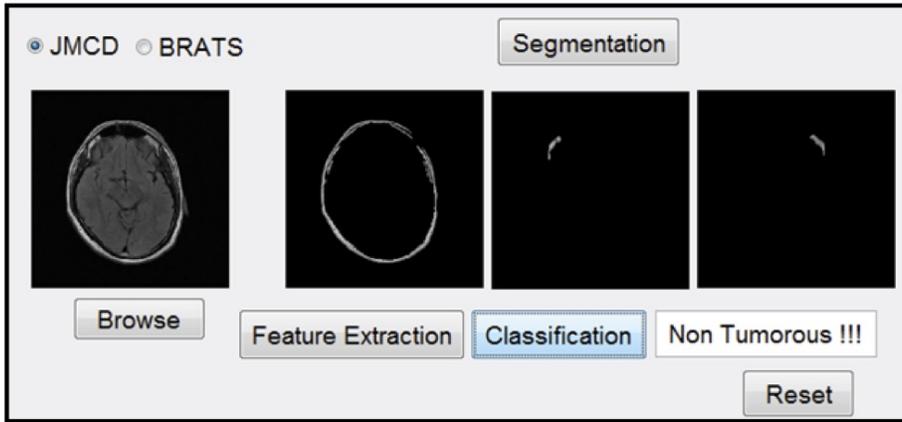
Performance of the proposed system is examined by domain experts as well as validated through statistical testing. The outcomes are verified and validated by domain experts of Netaji Subhash Chandra Bose Medical College, Jabalpur. Five radiologists examined the outcomes and found that execution of the proposed CAD system is quite efficient with successful classification of gliomas for more than 95% among overall results. Statistical testing of the proposed system is performed for 10-fold cross validation. Student’s *t*-test is performed and 95% confidence interval on accuracy is achieved with lower *p* values (< 0.01). This way performance validation provides adaptation and acceptability to the proposed CAD system.

5 Comparisons with the Existing CAD Systems

Some of the existing CAD systems have been implemented and tested on experimental datasets in the uniform environment. As performance of most of these systems is evaluated for tumor classification only, the comparison here is also performed for tumor classification on JMCD and BRATS datasets. Performance of the proposed system along with the existing CAD systems is shown in Table 3 for tumor classification only, and not for glioma identification. Some of the existing systems perform well on BRATS dataset and provide good results in the prediction of missing labels, but on JMCD dataset their performance is slightly degraded due to its size.



(a)



(b)

Fig. 3. Screenshot of the proposed CAD System for JMCD dataset (a) Tumorous image (b) Non-tumorous image.

Table 3. Performance comparison with other existing techniques for tumor detection on accuracy (in %).

Techniques / Datasets	JMCD	BRATS
Dahshan et al. (2010) [9]	96.25	99.99
Gupta and Khanna (2015) [20]	97.30	98.90
Nabizadeh et al. (2015) [11]	96.29	100
Subashini et al. (2016) [5]	94.43	97.80
Gupta et al. (2016) [14]	94.76	97.97
Gupta et al. (2018) [16]	98.14	97.73
Proposed CAD System	99.12	100

DWT coefficients are extracted from raw images in [9]. The LL subband is used with Haar wavelet and reduction is done by principal component analysis (PCA). The results are classified by KNN. DWT holds poor directionality and shift- variance properties, hence its performance is degraded in the classification task. In another work, block based segmentation is performed and mean is observed as a significant feature for extraction [20]. This also could not perform well on JMCD dataset. Numerous statistical features are used and classified by SVM [11]. These features distinguish quite successfully normal and abnormal tissues on BRATS dataset as observed accuracy is 100%, but not on JMCD. In another work, MR images using watershed and fuzzy c-means algorithm are segmented [5]. Various shape and intensity based features are extracted. Among nine statistical features, area is found the most prominent feature for the classification task. Again, performance is not good on both the datasets. The reason behind low performance may be related to the selection of the classifier. In a previous work, LBP is used as a feature vector and selective block based approach for segmentation is followed [14]. The binary patterns are found efficient, but such patterns are found sensitive towards noise. In another approach proposed by us, RLCP as feature is used and classified through NB classifier [16]. At that time numbers of patients were less in JMCD dataset. It is clear from Table 3 that the proposed system produces comparatively better results on both the datasets.

6 Conclusion

The challenges identified through state-of-the-art are addressed to some extent by the proposed system. The proposed CAD system shows efficient and quite accurate results for the detection of brain tumors along with glioma identification on T2-weighted MR images irrespective of scan views. Output of EM theorem is customized through its postprocessing. The experiments are performed on real datasets and collected raw images are preprocessed without any kind of distortion. The proposed system is tested and verified on two datasets. The observed accuracy on JMCD dataset is $(99.12 \pm 0.01)\%$, while the same on BRATS is $(100 \pm 0.00)\%$. Also, statistical testing as well as validation performed by domain experts enhances adaptivity of the proposed system for radiologists and clinicians.

Acknowledgement We thank MP MRI CT Scan & Diagnostic Centre, Netaji Subhash Chandra Bose Medical College, Jabalpur and radiologists working there for providing JMCD dataset and its ground truth. We are also thankful to radiologists for validating the system outputs.

Compliance with ethical standards The authors declare that they have no conflict of interest. For this type of study formal consent is not required.

References

1. Cbtrus fact sheet, Available online at: cbtrus.org/factsheet/factsheet.html. Last accessed July 20 (2016) 1–4.
2. The hindu: Over 2,500 indian kids suffer from brain tumour every year, *The Hindu* (2017) 1–6.
3. E.-S. A. El-Dahshan, H. M. Mohsen, K. Revett, A.-B. M. Salem, Computer-aided diagnosis of human brain tumor through mri: A survey and a new algorithm, *Expert systems with Applications* 41 (11) (2014) 5526–5545.
4. S. R. Hamilton, L. A. Aaltonen, et al., Pathology and genetics of tumours of the digestive system, Vol. 48, IARC press Lyon., 2000.
5. M. M. Subashini, S. K. Sahoo, V. Sunil, S. Easwaran, A non-invasive methodology for the grade identification of astrocytoma using image processing and artificial intelligence techniques, *Expert Systems with Applications* 43 (2016) 186–196.
6. A.-O. Boudraa, S. M. R. Dehak, Y.-M. Zhu, C. Pachai, Y.-G. Bao, J. Grimaud, Automated segmentation of multiple sclerosis lesions in multispectral mr imaging using fuzzy clustering, *Computers in biology and medicine* 30 (1) (2000) 23–40.
7. P. Natarajan, N. Krishnan, N. S. Kenkre, S. Nancy, B. P. Singh, Tumor detection using threshold operation in mri brain images, in: *Computational Intelligence & Computing Research (ICCIC), 2012 IEEE International Conference on, IEEE, 2012*, pp. 1–4.
8. D. Yamamoto, H. Arimura, S. Kakeda, T. Magome, Y. Yamashita, F. Toyofuku, M. Ohki, Y. Higashida, Y. Korogi, Computer-aided detection of multiple sclerosis lesions in brain magnetic resonance images: False positive reduction scheme consisted of rule-based, level set method, and support vector machine, *Computerized Medical Imaging and Graphics* 34 (5) (2010) 404–413.
9. E.-S. A. El-Dahshan, T. Hosny, A.-B. M. Salem, Hybrid intelligent techniques for mri brain images classification, *Digital Signal Processing* 20 (2) (2010) 433–441.
10. A. M. Malviya, A. S. Joshi, Gabor wavelet approach for automatic brain tumor detection, *International journal of emerging technology and advanced engineering*.
11. N. Nabizadeh, M. Kubat, Brain tumors detection and segmentation in mr images: Gabor wavelet vs. statistical features, *Computers & Electrical Engineering* 45 (2015) 286–301.
12. Z. Iscan, Z. Dokur, T. Ölmez, Tumor detection by using zernike moments on segmented magnetic resonance brain images, *Expert Systems with Applications* 37 (3) (2010) 2540–2549.
13. A. Kak, Expectation-maximization algorithm for clustering multidimensional numerical data, *An RVL Tutorial Presentation First Presented: Summer 2012*.
14. N. Gupta, A. Seal, P. Bhatele, P. Khanna, Selective block based approach for neoplasm detection from t2-weighted brain mris, in: *Signal and Image Processing (ICSIP), IEEE International Conference on, IEEE, 2016*, pp. 151–155.
15. N. Zulpe, V. Pawar, Glcm textural features for brain tumor classification, *International Journal of Computer Science Issues (IJCSI)* 9 (3) (2012) 354.
16. N. Gupta, P. Bhatele, P. Khanna, Identification of gliomas from brain mri through adaptive segmentation and run length of centralized patterns, *Journal of Computational Science* 25 (2018) 213–220.
17. R. Khayati, M. Vafadust, F. Towhidkhal, S. M. Nabavi, A novel method for automatic determination of different stages of multiple sclerosis lesions in brain mr flair images, *Computerized Medical Imaging and Graphics* 32 (2) (2008) 124–133.

18. H. Wang, B. Fei, A modified fuzzy c-means classification method using a multiscale diffusion filtering scheme, *Medical image analysis* 13 (2) (2009) 193–202.
19. B. N. Saha, N. Ray, R. Greiner, A. Murtha, H. Zhang, Quick detection of brain tumors and edemas: A bounding box method using symmetry, *Computerized medical imaging and graphics* 36 (2) (2012) 95–107.
20. N. Gupta, P. Khanna, A fast and efficient computer aided diagnostic system to detect tumor from brain magnetic resonance imaging, *International Journal of Imaging Systems and Technology* 25 (2) (2015) 123–130.
21. G. Vishnuvarthanan, M. P. Rajasekaran, P. Subbaraj, A. Vishnuvarthanan, An unsupervised learning method with a clustering approach for tumor identification and tissue segmentation in magnetic resonance brain images, *Applied Soft Computing* 38 (2016) 190–212.
22. M. Soltaninejad, G. Yang, T. Lambrou, N. Allinson, T. L. Jones, T. R. Barrick, F. A. Howe, X. Ye, Automated brain tumour detection and segmentation using superpixel-based extremely randomized trees in flair mri, *International journal of computer assisted radiology and surgery* 12 (2) (2017) 183–203.
23. E. Ilunga-Mbuyamba, J. G. Avina-Cervantes, A. Garcia-Perez, R. de Jesus Romero-Troncoso, H. Aguirre-Ramos, I. Cruz-Aceves, C. Chalopin, Localized active contour model with background intensity compensation applied on automatic mr brain tumor segmentation, *Neurocomputing* 220 (2017) 84–97.
24. M. Sharma, S. Mukherjee, Fuzzy c-means and snake model for segmenting astrocytoma type of brain tumor, *Int J Adv Eng Sci* 3 (3) (2013) 30–35.
25. M. Kistler, S. Bonaretti, M. Pfahrer, R. Niklaus, P. Büchler, The virtual skeleton database: an open access repository for biomedical research and collaboration, *Journal of medical Internet research* 15 (11).
26. A. Mehndiratta, F. L. Giesel, Brain tumour imaging, in: *Diagnostic techniques and surgical management of brain tumors*, InTech, 2011.
27. N. Gupta, R. K. Jha, Enhancement of dark images using dynamic stochastic resonance with anisotropic diffusion, *Journal of Electronic Imaging* 25 (2) (2016) 023017.
28. R. K. Jha, P. Biswas, B. Chatterji, Contrast enhancement of dark images using stochastic resonance, *IET Image Processing* 6 (3) (2012) 230–237.
29. S. Singh, K. Bovis, An evaluation of contrast enhancement techniques for mammographic breast masses, *IEEE Transactions on Information Technology in Biomedicine* 9 (1) (2005) 109–119.
30. H.-H. Loh, J.-G. Leu, R. C. Luo, The analysis of natural textures using run length features, *IEEE Transactions on Industrial Electronics* 35 (2) (1988) 323–328.
31. Y. Weiss, Motion segmentation using em—a short tutorial.
32. E. Weinstein, Maximization algorithm and applications list of concepts, Courant Institute of Mathematical Sciences.
33. X. Tang, Texture information in run-length matrices, *IEEE transactions on image processing* 7 (11) (1998) 1602–1609.
34. D. P. Huttenlocher, G. A. Klanderman, W. J. Rucklidge, Comparing images using the hausdorff distance, *IEEE Transactions on pattern analysis and machine intelligence* 15 (9) (1993) 850–863.
35. V. Thada, V. Jaglan, Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm, *International Journal of Innovations in Engineering and Technology* 2 (4) (2013) 202–205.

A Novel Phonetic Algorithm for Predicting Chinese Names using Chinese PinYin

Hua Zhao¹ and Fairouz Kamareddine²

¹ The school of Mathematical and Computer Sciences, Heriot Watt University, Edinburgh, UK hz103@hw.ac.uk

² The school of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, UK f.d.kamareddine@hw.ac.uk

Abstract. People’s names are initially written in the alphabet of their first language (e.g., Chinese, Russian, Arabic, etc.). However, these names are also romanised (i.e., written in the Roman (Latin) alphabet) for ease of global communication and interaction. The romanisation of names is not unique: the same name can have many different Roman versions. There has been little success at detecting the most accurate original name from a Roman version. This paper concentrates on Chinese names. It provides a new method that is able to find the most suitable Hanzi version (i.e., written in Chinese characters) for a Chinese name written in the PinYin version (i.e., written in the Romanised Chinese version). We develop a new algorithm that uses phonetic prediction for Chinese names in their Romanised Chinese version. We run our algorithm on a variety of data. Our experimental results establish the accuracy and feasibility of our new algorithm to accurately detect the most suitable Hanzi version of a Chinese name that is written in PinYin.

Keywords: Language Identification · NLP · Phonetic Prediction

1 Introduction

Being able to correctly move between an original name (written using the alphabet of the original language, e.g., Chinese) and its Romanised version (the name written using the Roman alphabet) is an essential part of improving Business, Education and Government. Mining fuzzy names that do not accurately represent the person in question is a challenge in text mining. This is especially problematic when original names are written using a different alphabet. E.g., when a Chinese name written using Chinese characters (aka Hanzi Chinese) is transformed into a corresponding version in the Roman alphabet (aka PinYin Chinese³). The transformation from the original alphabet to the Roman alphabet can result in a number of different answers some of which are accurate where others are not. For instance, Chinese people usually use their PinYin Chinese names outside of China and this can lead to problems due to the fact that the

³ PinYin is the convert between roman alphabet letters and lexical tone transcriptions from Chinese characters [8].

mapping between Hanzi Chinese names and PinYin Chinese names is not one-to-one. Different Chinese names can have the same Romanised name after they are converted.

For example, 'yi' is a pronunciation in Mandarin as a Chinese PinYin word. However, this PinYin word does have over 100 Chinese characters [7].

Name variation is a challenge for name detection [1, 2]. Most exact string matching in name variation can bring about the worst results [1]. In current research, several name matching methods exist for analysing people's names [4]. But most of these existing name matching methods only work with European languages, especially in the English language [1]. All these methods use the alphabet as the depiction language [1]. In contrast to these methods based on the alphabet for European languages, other languages use the so-called logograms as a means for representation. The most popular logogram is Hanzi (汉字) for Chinese characters [1].

Logograms are behind the challenge for name matching in Chinese [1]. For Chinese names, people have a small number of Hanzi as their whole names. But Chinese people's names contain tens of thousands of Hanzi [1]. Martschat et al. suggest that downstream tasks can improve the name matching in Chinese languages [6]. On name matching techniques, Peng et al. used their new method and the existing approaches in Chinese name matching for analysing two Chinese strings [1]. Wu et al. present a dictionary method for translating Chinese PinYin to Chinese Hanzi as a google patent [7]. However, there are limited methods that convert from Chinese PinYin to Chinese Hanzi. In this paper, we propose a new algorithm for name matching to convert Chinese PinYin to Chinese Hanzi. We use Chinese phonetic features as the prediction points and build the regular patterns from phonetic features in Chinese names as components of the prediction algorithm that identifies the suitable Chinese Hanzi.

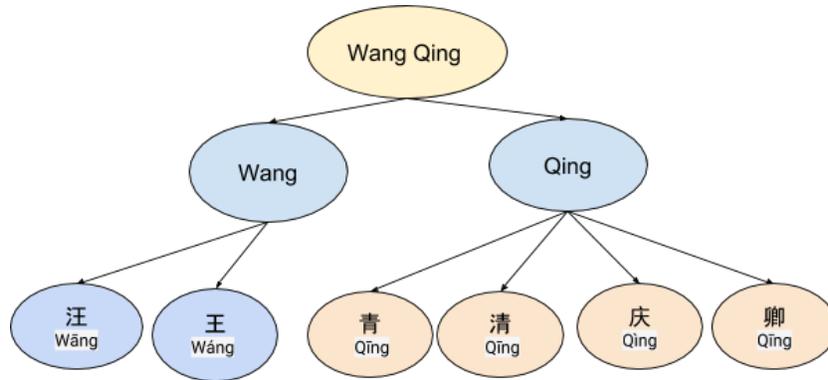
Chinese PinYin has 26 letters just like the Romanisation alphabet [9]. The only difference between the English alphabet and Chinese PinYin is 'ü'. So typically, people use 'v' as 'ü' in the global description [8, 9]. Furthermore, Chinese PinYin, it has four lexical-tone representations [9]. The lexical tone is an important component in PinYin for Chinese phonetic [8]. Four lexical tones are the high level tone, the low rising tone, the falling-rising tone and the high falling tone [2, 8, 9, 13]. Table 1 shows an example of the four lexical tones in PinYin 'wang'. Here, wāng is the high level tone of 'wang', wáng is the low rising tone of 'wang', wǎng is the falling-rising tone of 'wang', wàng as the high falling tone of 'wang'. Using PinYin to predict Chinese Hanzi, we can get a large number of possible options in Hanzi from one word of PinYin. This is a problem for name prediction.

PinYin practice can help to learn Chinese characters [8, 10]. It can illustrate the exact sound of a given Hanzi [8]. On researching Chinese names, we found that all the Chinese names have their regular phonetic patterns. For example, 'Wang Qing' is the PinYin name for a person's Chinese name '王青'. 'Wang' is the Second name, 'Qing' is the first name. This name's phonetic name is 'wáng

Table 1. An example of four lexical tones

Chinese Phonetic	Chinese Hanzi
wāng	汪
wáng	王
wǎng	网; 罔; 往;
wàng	旺; 忘; 望

qīng’. Here, wáng has a low rising tone in four lexical tones and qīng has a high level tone in four lexical tones.

**Fig. 1.** An example of the Chinese PinYin name convert to Hanzi name

Lots of Chinese people use the name as ‘Wang Qing’ in PinYin. Fig 1 displays the possibilities for the Chinese characters from a PinYin name ‘Wang Qing’. We can see that there are two possibilities of Chinese Hanzi for PinYin ‘Wang’. They are the ‘汪’ (wāng) and the ‘王’ (wáng) We also know that there are four possibilities of Hanzi for PinYin ‘Qing’. These are ‘卿’ (qīng), ‘庆’ (qìng), ‘清’ (qīng), ‘青’ (qīng). However, no people use ‘wàng qìng’, ‘wāng qǐng’, ‘wǎng qǐng’ as the Chinese phonetic versions of their Chinese names. That is because the pronunciation of these pairs of Chinese phonetic names is typically mouthful.

Depending on these regular patterns in Chinese phonetic names, we made two training datasets to promote our new algorithm development. In this paper, we propose a new algorithm that reduces the number of possibilities of the results of name prediction from Chinese PinYin names to Chinese Hanzi names. We also compare the accuracy of our new algorithm with google translator to show that our new algorithm has a good accuracy on name prediction from Chinese PinYin names to Chinese Hanzi names.

Our contributions are:

1. A new algorithm for name prediction from Chinese PinYin names to Chinese Hanzi names.

2. Good accuracy on name prediction for Chinese names.
3. Efficient reduction of the number of possibilities for name prediction of Chinese names using PinYin names.

In section 2, we describe the related works. In section 3, we introduce our new algorithm for predicting Chinese PinYin names into Chinese Hanzi names. In section 4, we describe the training datasets in detail. In section 5, we outline the experiments' results of testing the algorithm and show the comparison of the accuracy of our algorithm with google translator. In section 6, we conclude and give some future work.

2 Related Work

Our goal is to develop an algorithm that can work for name prediction of Chinese names using PinYin names. In Natural Language Processing (NLP), some researchers concentrate on name matching and name prediction. Choudhury et al. display a word model for the texting language [12]. This model can understand and translate SMS texts into proper alphabet texts. For example, in SMS, people usually use 'u' instead of 'you'. This model can translate 'u' to 'you' for people to understand SMS texts. This is achieved by making a dictionary for storing all the frequencies of each SMS text word, and then using Hidden Markov Model (HMM) modelling to process all the text words. As an extension of this method, Wang et al. used an HMM model for processing name matching in the case of Chinese names [2]. This latter paper used a left-to-right word model to find out the Chinese names using people's aliases [2]. Performing name matching in Chinese names, Peng et al. gave a method for processing the two Chinese strings for name matching [1]. This latter method can do name matching between two Hanzi names [1]. For name matching in different languages, Freeman et al. enhanced an algorithm for matching English names with Arabic names [5]. However, there are limited methods and algorithms for name prediction of Chinese names using PinYin names with a good accuracy. In our previous research [14], we used a decision tree method on fuzzy name identification for Chinese names using PinYin names. However, that method did not accurately detect the suitable names and instead, it resulted in a large number of possibilities for one PinYin name. Therefore, in this paper, we propose a new algorithm that can reduce the number of possible results, and also can get a good accuracy for name prediction from Chinese PinYin names to Chinese Hanzi names.

In the next section, we will introduce our new phonetic prediction algorithm.

3 New Phonetic Prediction Algorithm

In this section, we describe our new phonetic prediction algorithm which predicts Chinese names using PinYin names. Fig 2 shows a simple example of the idea on our algorithm doing name prediction with a PinYin name, 'Wang Qing Hua'. Here, we can see that the example displays the algorithm using a left to right

process to make a phonetic prediction for the PinYin name. It does the prediction work from 'Wang' to 'Qing' to 'Hua'. Finally, the Hanzi name is predicted as '王青骅'.

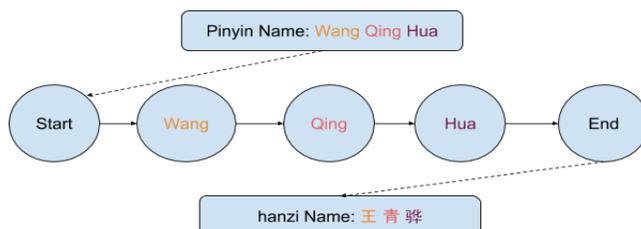


Fig. 2. An example of the left to right processing in Chinese phonetic prediction

In Chinese, there are only these two kinds of first names in a full name for Han people⁴. The two kinds are either one single first name or a pair of first names. In our algorithm, we process the PinYin names that can be classified under these two kinds. Recall that our phonetic prediction algorithm needs to use four lexical tones, which we refer to by the numbers '1', '2', '3', '4'. Table 2 shows an example of our algorithm using four lexical tones. To define four lexical tones in our algorithm, we use '1' as the high level tone, '2' as the low rising tone, '3' as the falling-rising tone and '4' as the high falling tone. In table 2, we can see wāng as 'wang1' for PinYin 'wang' in the high level tone, wáng as 'wang2' for PinYin 'wang' in the low rising tone, wǎng as 'wang3' for PinYin 'wang' in the falling-rising tone and wàng as 'wang4' for PinYin 'wang' in the high falling tone.

Table 2. An example of four lexical tones

PinYin name	Phonetic name in our algorithm
wāng	wang1
wáng	wang2
wǎng	wang3
wàng	wang4

Let's determine a Chinese PinYin name as N . Below, we use g_1 as the second name, g_2^n as the first name.

$$N = (g_1, g_2^n) \quad (1)$$

We define the number of possible Chinese characters for N as p_i , so from a number of possibilities of the Chinese characters, we can get the phonetic names

⁴ Han people are the largest ethnic group in China (93%) and in the world (20%) [15].

for the second name as below,

$$\mathit{argMax}(p_i|g_1) = C_i \quad (2)$$

Here, C_i is a list of possibilities Hanzi names of the second name. Typically, i is equal to 1, a maximum number of i is 3.

From C_i , we can get its phonetic names, the formula below shows that S_i is a list of phonetic names from its Hanzi names. It does have i number of phonetic names.

$$C_i = S_i \quad (3)$$

We use S_i to predict the phonetic names for First names, x_i^1 . The formula is as follows:

$$k = \sum(P(x_i^1)) = \sum(p_i|S_i|g_2^1) = 1 \quad (4)$$

Here, $P(x_i^1)$ is the possibilities of the phonetic names of the first name.

If there are two first names from the PinYin name, $n = 2$, we do process the phonetic names of second first name's formula with x_i^1 as follows:

$$k = \sum(P(x_i^2)) = \sum(p_i|x_i^1|g_2^2) = 1 \quad (5)$$

For $P(x_i^n)$, we define k as the possibilities of the phonetic names:

$$k = (P(x_1^n) + P(x_2^n) + P(x_3^n) + P(x_4^n)) \quad (6)$$

Here, we also have that:

$$k = \begin{cases} 1, & \text{if phonetic names are available} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

In order to select the possible phonetic names for first names, we use Standard Deviation (SD) using the following formula:

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (P(x_i^n) - \bar{k})^2} \quad (8)$$

Here, in table 3, we made a claim area for the SD results for selecting the phonetic names for first names,

Table 3. SD Range in our algorithm

SD Range	$P(x_i^n)$
$s \geq 0.3$	$\mathit{Max}(P(x_i^n))$
$0.0 < s < 0.3$	$0.35 \leq P(x_i^n) \leq 0.6$

From the selected phonetic names for first names, we get the Hanzi names as ' f^n '. Therefore, the final Hanzi name as the (C_i, f^n) .

Fig 3 shows the process of the algorithm during the phonetic prediction using one first name for a full Chinese name.

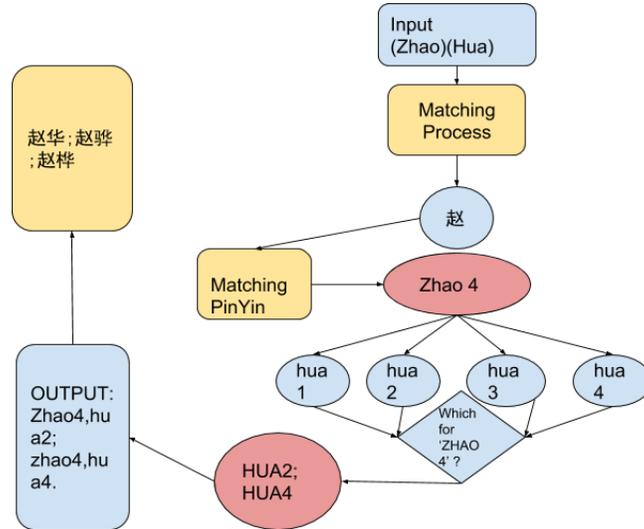


Fig. 3. An example of the phonetic algorithm using one first name

From fig 3, we can see that the PinYin name is 'zhao hua', the family name is 'zhao' and the first name is 'hua'. Firstly, our algorithm does process its family name to get a Hanzi family name, '赵'. Then, from this Hanzi, the algorithm gets the phonetic name zhào. In the algorithm, we define four lexical-tone representations as the number '1', '2', '3', '4'. Here, '1' as the high level tone, '2' as the low rising tone, '3' as the falling-rising tone, '4' as the high falling tone. Therefore, we get the phonetic name from Hanzi '赵' to be 'zhao4', which is the same as zhào. After we get the phonetic name of the family name, we use it to predict the first name 'hua'. We classify the phonetic names in first name as 'hua1', 'hua2', 'hua3', 'hua4'. Then, our algorithm does process all the phonetic names in first names to get the possible phonetic names for the first name. In this figure, we can see that the possible phonetic names for its first name are 'hua2' and 'hua4'. From this condition, we can get the final possible Hanzi full name from the two possible phonetic names 'zhao4,hua2', 'zhao4,hua4' as the '赵华', '赵骅', '赵桦'.

Fig 4 displays an example of our algorithm on processing the PinYin name using two first names. The example PinYin name is 'wang fu chun', 'wang' is the family name, 'fu' is the first of the first name, 'chun' is the second first name. From this figure, we can see that after our algorithm predicts the phonetic names

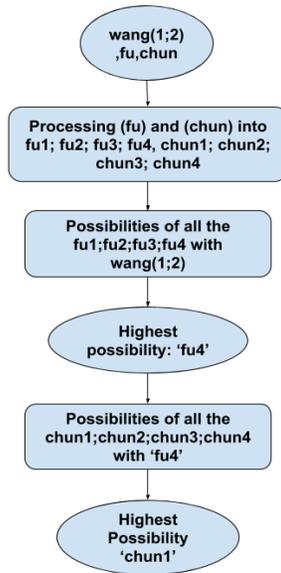


Fig. 4. An example of the phonetic algorithm using two first names

from the PinYin name 'wang', we can get two possible phonetic names for this family name. They are wāng as 'wang1' and wáng as 'wang2'. After this, our algorithm uses these two possible phonetic names in family names to predict the first one of the first names, 'fu'. The algorithm compares four phonetic names fū , fú, fǔ and fù to get the possible phonetic names of 'fu' which is fù. The next step is that our algorithm uses the possible phonetic name fù to predict the second first name, 'chun'. Our algorithm compares the phonetic name 'fu4' with 'chun1', 'chun2', 'chun3' and 'chun4' to predict the possible phonetic name of the second first name which is chūn. Therefore, the final possibilities of the phonetic name of the full name are 'wang1 fu4 chun1' and 'wang2 fu4 chun1'. So the results of this example using our algorithm are the '汪福春', '王福春'.

In the next section, we will describe the datasets that we use for evaluating our algorithm.

4 Data Processing

In this section, we talk about the datasets that we use in our phonetic prediction algorithm. In our algorithm, we use two training datasets for predicting Hanzi names using PinYin names. We use 1018047 data to create two training datasets for our algorithms [11].

Fig 5 shows an example of our first training dataset. We define the first training dataset to compare the Chinese Hanzi names, PinYin names and each

name's phonetic name. Therefore, our algorithm can use this training dataset to get the Hanzi names from phonetic names, and also use it to process the possible phonetic names.

```

姜,zhen,0,41,22,zhen1
适,shi,0,4,0,shi4
遨,ao,0,8,0,ao2
蓄,xu,0,4,1,xu4
币,bi,0,1,0,bi4
邈,yao,0,141,29,yao2
韩,wei,0,1,0,wei2
希,xi,0,1003,235,xi1
蓉,rong,0,1077,1157,rong2
遑,yuan,0,10,0,yuan3
遣,qian,0,1,0,qian3
琴,chen,0,3,0,chen1
帖,tie,14,1,0,tie1
汛,xun,0,20,3,xun4

```

Fig. 5. An example of the first training dataset

This training dataset has six conditions. The first one is for Chinese Hanzi names; the second condition is for the PinYin name of the Hanzi name, the third one is the frequency of the family names for each Hanzi name. The fourth condition is the frequency of the first name in males for each name, and the fifth one is the frequency of the first name in females on each name. The last condition is the phonetic name for each Hanzi name. For making the first training dataset, we processed the collected data from Hanzi to PinYin and then classified all the data into family names and first names. This data is the labeled data, so we classify all the first names with gender. In Chinese names, people do not have gender for their family names. After this, we list all of the processed words with phonetic names.

Table 4. An example of the first training dataset

Hanzi	PinYin	Frequency of the Family name	Frequency of the first name(male)	Frequency of the First name(female)	Phonetic name
正	zheng	0	3842	312	zheng4

For example, table 4 displays the conditions of the Hanzi '正' in the first training dataset. 'zheng' is the PinYin for '正'; it has zero frequency as a second name. We also can see that the frequency of this word using it as a first name in male is 3842. And the frequency of it as a female's first name is 312. The phonetic name of '正' is zhèng as 'zheng4' in our algorithm.

The second training dataset is a dataset for analysing the relationship between a family name and a first name in phonetic names. Fig 6 shows an example of the second training dataset.

```

7,wu2,xiao3, long2
8,yu2,yong3,
9,yuan2,man3,
10,hu2,kun1,
11,shi2,,
12,zhu1,wei4,zhun3
13,zhong1,min3,
14,xu2,hai3,feng1
15,wang2,xiao1,li4

```

Fig. 6. An example of the second training dataset

In fig 6, we can see all the phonetic words in this training dataset. In this training dataset, we use a comma to separate the collected names. We processed all the collected names as phonetic names. And then, we classified them as family names, first names. In the case when a full name has two first names, we use a comma to separate them as well. In the end, we made all the full names in phonetic names as the labeled data. The second training dataset is an assistant to help our algorithm to identify the relationships from family names to first names for phonetic prediction.

In the next section, we will introduce the testing results for our algorithm and how we processed the experiments in detail.

5 Experiments

In this section, we will describe the testing algorithm on getting the accuracy from the experiments' results. We will also display the experiments of testing our algorithm and compare it with google translate. For testing the accuracy of our algorithm, we will show the testing results in detail.

5.1 Experiment Settings

For the experiments, we created an algorithm that can analyse the results of the experiments. Fig 7 shows the accuracy of our results from the experiments.

In this figure, we define real data $N = (g_1, g_2^n)$ as the full name in Chinese character. Here, g_1 is the family name in Hanzi and g_2^n is the first name in Hanzi. In N , n can be equal to one or equal to two. For the testing results, we made each result as a list. In each list, we add '!' behind each family name to distinguish with the first names. If there are two first names in a full name, we add ';' behind the first one of the two first names. In fig 7, we can see that there is an example to show a testing result of a full name as follows:

$$[g_1!, g_1!, g_2', g_2', g_2'] \quad (9)$$

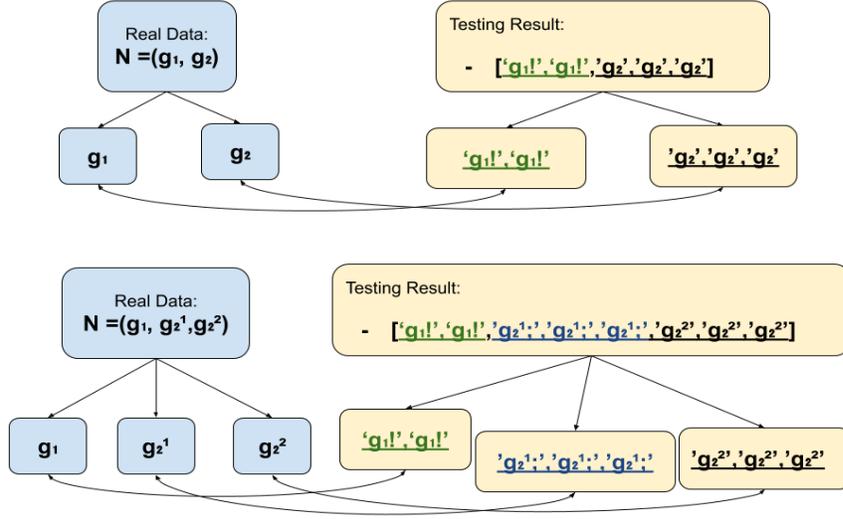


Fig. 7. The process of getting the accuracy from the experiments

In the list, two ' $g_1!$ ' are the prediction results for the family name, three ' g_2' ' are the prediction results for the first name. Therefore, the algorithm can find the accuracy of comparing the family name between ' $g_1!$ ' in real data and two ' $g_1!$ ' in testing results. Similarly, the algorithm can find the accuracy of comparing the first name between ' g_2' ' in real data and three ' g_2' ' in testing results.

On processing the accuracy in two first names of a full name, fig 7 shows the testing result for a full name as below,

$$[g_1!, g_1!, g_2^1!, g_2^1!, g_2^1!, g_2^2!, g_2^2!, g_2^2!] \quad (10)$$

In the list, two ' $g_1!$ ' are the prediction results for the family name, three ' $g_2^1!$ ' are the prediction results for the first one in two first names. And three ' $g_2^2!$ ' are the results for the second one in two first names. Therefore, the algorithm can find out the accuracy of comparing the family name between ' $g_1!$ ' in real data and two ' $g_1!$ ' in testing results, comparing the first one in two names between ' $g_2^1!$ ' in real data and three ' $g_2^1!$ ' in testing results. At last, comparing the second name in two first names between ' $g_2^2!$ ' in real data and three ' $g_2^2!$ ' in testing results.

5.2 Results and Discussion

In experiments, we used 48935 data to test our algorithm. On testing our algorithm, we compared it with google translator. In the experiments, we also

separated the testing of the accuracy of results in full name, family name, two first names of the full name, one first name of the full name and first names.

Table 5 shows the accuracy of our algorithm on testing the separate part using our algorithm. Fig 8 displays the accuracy in a bar chart.

Table 5. Testing results on each parts of the name prediction using our algorithm

Testing parts	Accuracy(%)
Second names	93%
First name(One first name)	60%
First name One (Two first names)	73%
First name Two (Two first names)	71%

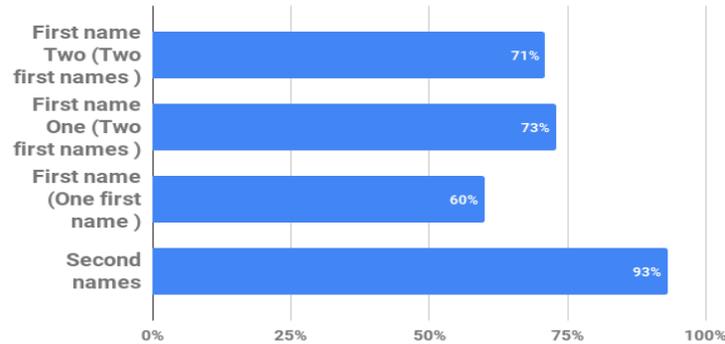


Fig. 8. The accuracy among each part of the full name with our algorithm

When testing the accuracy of the family name for name prediction, we used 45697 family names. Our algorithm finds out 45697 family names among 48036 names. We used 32685 names as the “two first names” for testing our algorithm. And we also used 16250 names for one “first names” when testing our algorithm. When testing our algorithm for two first names of the full name, we found that 29529 names can be processed.

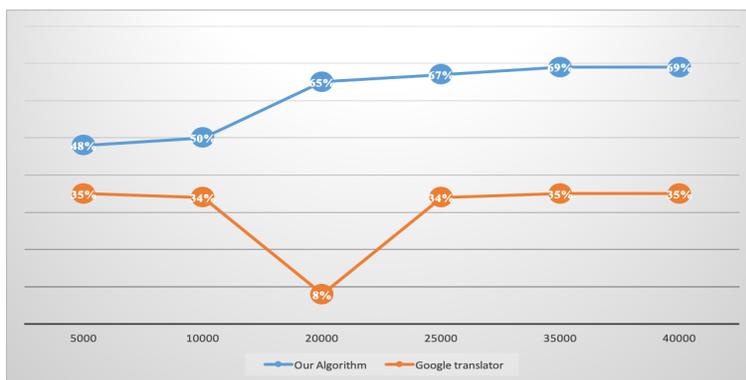
In fig 8, we can see that the highest accuracy of our algorithm on predicting family names is 93 %. Furthermore, we obtained the accuracy of 73% and 71% on testing the two first names of the full name with our algorithm. Moreover, the accuracy on testing the first names for one first name of a full name is 60%.

Table 6 shows the testing results in F-score, Precision and Recall among all parts of the full names using our algorithm.

Table 6. F-Score, Precision, Recall by Testing points

Parts on Testing	F-Score	Precision	Recall
Second Name	0.963	0.446	0.933
First Name (One First Name)	0.742	0.093	0.591
First Name one (Two First Names)	0.843	0.233	0.73
First Name one (Two First Names)	0.823	0.226	0.709

When testing the accuracy of our algorithm, we also compared it with Google translator. Fig 9 shows the testing result between our algorithm and google translator on name prediction from Chinese PinYin names to Chinese Hanzi names. Here, the blue line is defined as the results from our algorithm, the orange line is defined as the results from Google translator. We used 40000 data to do this testing experiment. The result shows that the accuracy of our algorithm is 69% and the accuracy of Google translator is 35%.

**Fig. 9.** Testing results between our algorithm and Google translator

We also carried out experiments on testing the name prediction between two kinds of Chinese names. They are the full names containing one first name, and the full names containing two first names. Fig 10 displays the testing results of this experiment. We used 40000 data for this experiment. In fig 10 , we can see that the accuracy of name prediction with the full names containing two first names is higher than the accuracy of the full names containing one first names. We know that on doing Chinese name prediction with our algorithm, the process does use more times for predicting a full name that has two first names than a full name that has one first name. Therefore, we can see that our algorithm

can be extended to processing text prediction of texts in Chinese Hanzi using PinYin texts.

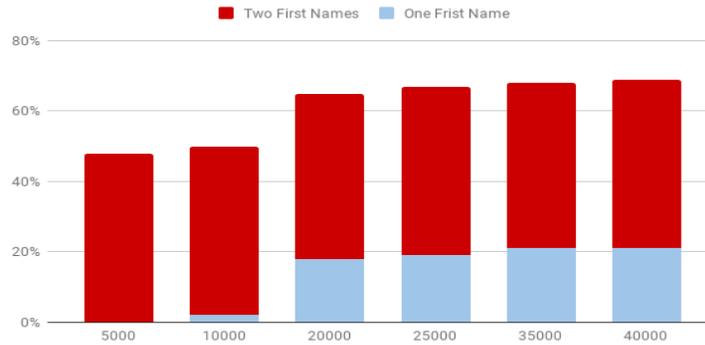


Fig. 10. Testing results between two first names and one first name in a full name in our algorithm

6 Conclusion

In this paper, we have presented a new phonetic prediction algorithm for Chinese name prediction. Our algorithm uses Chinese PinYin names to predict Chinese Hanzi names. We have demonstrated the new method of processing the accuracy for testing experiments. Our new algorithm enjoys good accuracy on name prediction from PinYin names to Hanzi names which were demonstrated by our experiments. Our new method also does help to reduce the number of possibilities in name prediction. We believe that this method is useful for many research areas related to the analysis of data on Chinese names and that it can also be used for name matching in other languages. In the next step, we want to increase the efficiency of our phonetic prediction algorithm. We will also want to extend our algorithm so that it can work on processing text prediction for natural language using PinYin texts.

References

1. Nanyun Peng, Mo Yu, Mark Dredze: An Empirical Study of Chinese Name Matching and Applications. *ACL* (2) : 377-383 (2015).
2. Wei-Chung Wang, Hung-Chen Chen, Zhi-Kai Ji, Hui-I Hsiao, Yu-Shian Chiu, Lun-Wei Ku: Whose Nickname is This? Recognizing Politicians from Their Aliases. *NUT@COLING* : 61-69 (2016).
3. T. El-Shishtawy: A Hybrid Algorithm for Matching Arabic Names. *CoRR* abs/1309.5657 (2013).

4. William W. Cohen, Pradeep Ravikumar, Stephen E. Fienberg: A Comparison of String Distance Metrics for Name-Matching Tasks. *IIWeb* : 73-78 (2003).
5. Andrew Freeman, Sherri L. Condon, Christopher Ackerman: Cross Linguistic Name Matching in English and Arabic. *HLT-NAACL* (2006).
6. Sebastian Martschat, Jie Cai, Samuel Broscheit, Éva Mújdricza-Maydt, Michael Strube: A Multigraph Model for Coreference Resolution. *EMNLP-CoNLL Shared Task* : 100-106(2012).
7. Jun Wu, Hulcan Zhu, Hongjun Zhu: Systems and methods for translating Chinese PinYin to Chinese characters. US Patent 7,478,033 (2009).
8. Lin, D., McBride-Chang, C., Shu, H., Zhang, Y., Li, H., Zhang, J., Aram, D., and Levin, I: Small wins big: Analytic PinYin skills promote Chinese word reading. *Psychological Science*, 21, 1117–1122. doi: 10.1177/0956797610375447 (2010).
9. Institute of Linguistics, Chinese Academy of Social Sciences . *Xinhua Dictionary* (10th Edition). Beijing: The Commercial Press(2004).
10. Chung, K. K. H: Effective use of Hanyu PinYin and English translations as extra stimulus prompts on learning of Chinese characters. *Educational Psychology*, 22 (2): 150 – 164 (2002).
11. Jingchao Hu: ngender0.1.1: GuessgenderforChinesenames. Available at: <https://pypi.python.org/pypi/ngender/0.1.1>, Last accessed: September 2018.
12. Monojit Choudhury, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar, Anupam Basu: Investigation and modeling of the structure of texting language. *IJDAR* 10(3-4): 157-174 (2007).
13. Li Liu, Danling Peng, Guosheng Ding, Zhen Jin, Lei Zhang, Ke Li, Chen Chuan-sheng: Dissociation in the neural basis underlying Chinese tone and vowel production. *Neuroimage*, vol. 29 (pg. 515-23) (2006).
14. Hua Zhao , Fairouz Kamareddine: A Decision Tree Method on Fuzzy Name Identification from Chinese phonemic names to Chinese names. *CSCI 2018*, pg. 227-232. DOI 10.1109/CSCI.2018 (2018).
15. Y.G. Yao, Q.P. Kong, H.J. Bandelt, T. Kivisild, Y.P. Zhang: Phylogeographic differentiation of mitochondrial DNA in Han Chinese. *Am. J. Hum. Genet.*, 70, pp. 635-651 (2002).

Detecting Customer Churn Signals for Telecommunication Industry Through Analyzing Phone Call Transcripts with Recurrent Neural Networks

Junmei Zhong, William Li

Marchex Inc
520 Pike Street, Seattle, WA, USA 98101
jzhong@marchex.com
wli@marchex.com

Abstract. For telecommunication service providers, great efforts have been made to retain the loyalty of existing customers. This can have a great impact on reducing business costs and generating revenue. To ensure the effectiveness and efficiency of the efforts, it is important to understand customer concerns as early as possible to prevent customer churn, a customer action of canceling the subscription and switching to a competitive service provider. In this paper, we analyze the actual customer phone call data by developing recurrent neural networks (RNNs)-based long short-term memory (LSTM) and gated recurrent unit (GRU) predictive models to detect the churn signals from transcripts. Comparative studies show that when sufficient training examples are provided with our company's scalable transcript annotation method, our GRU predictive model performs best and generates exciting performance in churn prediction.

Keywords: Churn Prediction, Deep Learning, RNN, LSTM, GRU.

1 Introduction

For telecommunication companies, great efforts have been made to improve the quality of their customer service to retain existing customers since the cost of acquiring new customers is much higher than that of retaining existing subscribers. However, companies are still losing customers before they take effective actions. To figure out effective solutions, it is important to understand customer concerns as early as possible to prevent the customer churn, which is the customer action of canceling a subscription and switching to a competitive service provider.

When customers cancel their subscriptions, the result is an immediate loss of business. Customers typically choose to switch to other providers when they are no longer satisfied with their existing service provider. In the telecommunications industry, customer churn is not an instant or sudden action, but rather the result of a long interactive process between the customer and the service provider's customer service representatives, through which the customer finally decides to leave after not being able to get a satisfactory solution. During that process, customers typically first make phone calls to the service provider's customer service representatives to express their concerns

and request for a satisfactory solution. It's at this very moment, customers' churn signals appear in phone calls.

If the service providers could accomplish the early detection of these churn signals, and then figure out effective and personalized solutions to address the individual concerns, there is a strong opportunity for the existing service provider to retain the customers who are going to churn, and therefore minimize business loss. Recognizing churn signals in the early stage is critical for any service provider that hopes to take effective steps to re-instill or improve customer satisfaction and retain the customer's business. Previous work on churn prediction has been reported based on social media data, such as microblog data, using either the traditional machine learning algorithms [1, 2] or the convolutional neural network (CNN) [3]. Although shedding light on the investigation of churn prediction, the positive impact of those churn prediction results for telecommunication businesses is still limited. Our analysis reveals that there are two main fundamental factors that make social media data far from being ideal for customer churn prediction in the telecommunication industry. The first factor is that customers usually begin by conducting phone calls to their service provider to have a direct talk rather than indirectly going to the social networks to express their concerns about their subscriptions. This is because, at the current time, service providers mainly use phone calls to answer customers' questions. In the future, online chatting may become a popular customer service tool, and then we can combine the two kinds of data for churn prediction. Because the phone call is the primary channel for customers to get resolutions to their concerns, phone call data is currently more reliable for early detection of churn signals. The second factor is that when the predictive models for the microblog study were trained, there was insufficient training data. For example, the datasets for AT&T, T-Mobile and Verizon, only 2299, 1073, and 2495 golden-truth examples, respectively, were used for training, validating, and testing the predictive models [1, 2, 3]. So, the CNN model for churn prediction [3] with many parameters is not guaranteed to be trained well enough with so few training examples due to the curse of dimensionality.

For our investigation about customer churn prediction, we collaborate with one of the biggest telecommunication companies in the United States, to use their actual customers' phone call data for building a churn prediction model that could, in turn, produce real impact on the telecommunication company's business. For this purpose, we collect audio recordings of actual customer phone calls and apply speech recognition techniques to translate the audio data into text transcripts for deep learning. Given the massive amount of text data of phone call transcripts and the 1 percent rare event of churn signals, the big challenge for building a reliable predictive model is the data labeling work required to provide sufficient training data to train the deep learning model. In the case of churn prediction, since most of the phone calls are non-churny, it is very challenging to pick sufficient churny phone calls. If 5,000 churny phone calls need to be prepared for training the model, we need to label roughly 500,000 phone calls. If the labeling is done by either listening to the individual audio files of the phone calls or by reviewing the individual transcripts of the phone calls, it is both time consuming and expensive, if not cost-prohibitive, because these methods are not scalable. As a result, it is practically infeasible to label data using these methods. It is necessary to use a scalable method to do the labeling work in order to get sufficient churny calls from the millions of phone calls in a fast and cost-affordable way. Our

company has developed a scalable labeling method to label the transcript data on the utterance level. Using this scalable data labeling method, it only takes about one week for four labelers to get the required training examples.

Our churn prediction system consists of four core components. First, we collect the phone call transcript data from the database. We only collect the transcripts from the caller channel since our knowledge of telecommunications customer service indicates that customer churn signals mainly appear in the caller channel, not in the agent channel. The sole use of the caller channel transcript data can significantly reduce the amount of data for both data labeling and data analysis. Second, using our scalable data labeling method, we label the transcript data of phone calls and use them as the training, validation, and testing examples. Third, we apply natural language processing (NLP) algorithms for document tokenization and vector representation for tokens. The fourth and last component is training the text classification model for churn prediction using the RNN-based algorithms. Experiments show that when we can annotate sufficient training examples, our RNN-based GRU model generates state-of-the-art performance for churn prediction measured by the quantitative metrics of F1-score, precision and recall.

Our contributions are demonstrated in the following two aspects:

- To the best of our knowledge, we are the first to conduct the analysis of actual customer service phone calls for customer churn prediction for telecommunication industry. Furthermore, we propose to only use the caller channel for churn prediction. This not only reduces the amount of data for both fast data labeling and data analysis, but also improves prediction performance.
- We develop AI algorithms including NLP and RNN algorithms for churn prediction from text data of transcripts with sufficient training data compared with the previous work, making the deep learning models very efficient.

The rest of the paper is organized as follows: In Section 2, we discuss the research methodology in detail. In Section 3, we present the experimental results. We conclude the paper with some discussions and the direction of future work in Section 4.

2 Research Methodology

There are 3 important components for the research methodology of customer churn prediction. Problem definition and data preparation, embedded vector representation for individual words in the training data of transcripts, and the training of RNN-based deep learning models for churn prediction.

2.1 Problem Definition and Data Preparation

Our phone call data is from daily customer service phone calls. Each phone call consists of two channels: the agent channel and the caller channel. Each agent or caller channel consists of a sequence of utterances. On average, the phone calls are about 45 minutes and the transcripts are long documents. According to our domain knowledge of telecommunications, the customer’s churn information is contained in the customer’s emotion and sentiment information, which is mainly contained in the caller channel,

seldomly present in the agent channel. As a result, we propose to only collect the caller channel transcripts. This not only reduces the amount of data to review for the labeling task, but also has the potential to improve the prediction performance by discarding the irrelevant agent channel. We label the caller channel transcripts and take them as the training data. For ensuring the quality of the training data, we only label the calls that last more than five minutes so there is enough text data for deep learning. Knowing the characteristics of our phone call data, we define the customer churn prediction problem as a binary text classification problem: churn and non-churn. For this purpose, each document is represented as a feature vector through the RNN-based sequence model, and the classification is accomplished with the full connection component in the RNN-architectures.

2.2 Embedded Vector Representation

In the early work on NLP and text mining, after the tokenization for documents, the vector space model is usually used for document representation, in which the values of individual words are estimated using the TF-IDF weighting method, forming the feature vector of each document. Such a bag of words (BOW) representation is not efficient for the classification task because it does not consider the order information of words appearing in the texts let alone the other semantic information in each document. Even if the N-Grams ($N > 1$) are used to get some ordering information about the tokens, their impact on classification is still limited for the following issues. First, the N-Grams ($n > 1$) representation for documents results in much increased features, generating the sparser representation of the documents. The sparse representation of document challenges machine learning algorithms very much. Second, the increased features require many more training examples to be used for training the model so as to reduce the risk of being overfitted. The last one is that the semantic information in the text is still not extracted very much. For example, when some customers use different words or synonyms to express the similar feeling, the transcripts will be similar to each other, but the N-Gram representation still cannot extract such semantic information. Consequently, the N-Grams representation increases the cost for building machine learning models without having a big performance improvement. In this work, we use the word embeddings of word2Vec [4] to represent individual words.

Since the BOW vector representation is not efficient to capture the semantic information from documents with limitations of high dimensionality and sparsity, researchers have investigated different ways to represent documents and words in an embedded low-dimensional dense vector space. The Word2vec algorithm [4] is such a distributed representation learning algorithm to learn the continuous dense vector representation for words in the embedded low dimensional vector space. It consists of 2 related models: the continuous bag-of-words (CBOW) model and the skip-gram model as shown in Fig.1. The CBOW predicts the current word from its surrounding context words in a sentence within a window centered at the current word, while the skip-gram model predicts a given word's surrounding context words in a sentence within a symmetric window.

The Word2vec model can be trained with either the hierarchical softmax or negative sampling method. The hierarchical softmax uses a Huffman tree to reduce the computational complexity by only updating the vectors on the path from the tree root to the leaf node (the current word) while the negative sampling accomplishes this goal and improves the vector quality of low-frequency words by only sampling a small number of the negative samples for updating the vectors in the backpropagation process for which the high-frequency words are down-sampled and the low-frequency words are up-sampled by frequency lifting. These models are the two-layer shallow neural networks. Word2vec takes as its input the high dimensional one-hot vectors of the words in the corpus and produces the dense vector space of several hundred dimensions such that each unique word in the corpus is represented by a dense vector in the vector space. A very salient feature of this kind of word embeddings is that word vectors in the vector space are close to each other when the corresponding words are semantically similar to each other. This offers the benefit that we can infer semantically similar words from the vector space for a word if the word's vector is known and it hence has attracted tremendous attention in text mining, machine translation, and document analysis. But the word2Vec algorithm still has its limitation in representing documents.

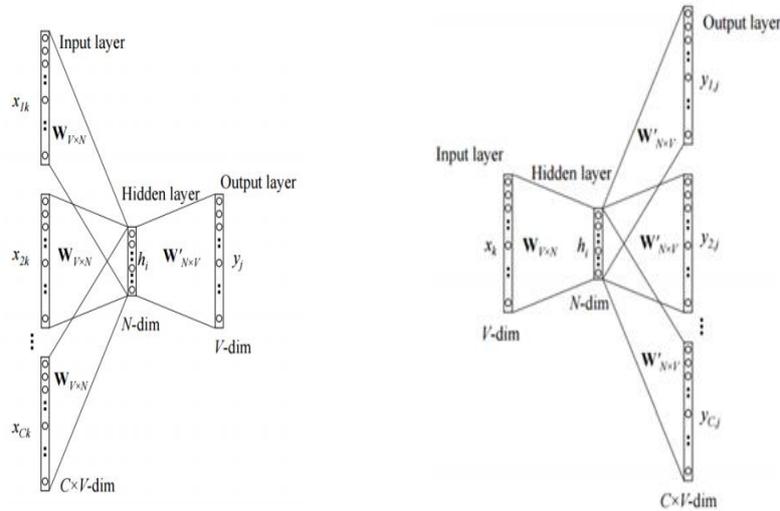


Fig. 1. The CBOW model (left) and Skip-gram(right) model in word2Vec [4]

2.3 RNNs for Churn Prediction

In this paper, we model the individual transcripts as the time series sequence data with some long-term syntactic and semantic dependencies/relationships between the individual words even if they are far from each other, and we believe such long-term dependencies are useful for churn prediction. As a result, we use the RNN-based deep learning algorithms for supervised learning to classify the phone calls into churns and non-churns. We train the RNN-based predictive models and compare them with the

previously published work on churn prediction for telecommunication industry [1, 2, 3].

2.3.1 The Basic RNN Model

CNN and RNN are two mainstreams of deep learning algorithms. RNNs are very suitable for modeling the time series sequence data for prediction and forecasting tasks. As shown in Figure 2, the chain-link RNN architecture consists of a sequence of neural networks (modules) sharing the same parameters. Each module is used to analyze the information at a time in the sequence. At any time, the RNN system takes as inputs the information $x(t)$ at a specific time t and the output $h(t-1)$ of the previous module $c(t-1)$, and then outputs $y(t)$ through the nonlinear activation function \tanh .

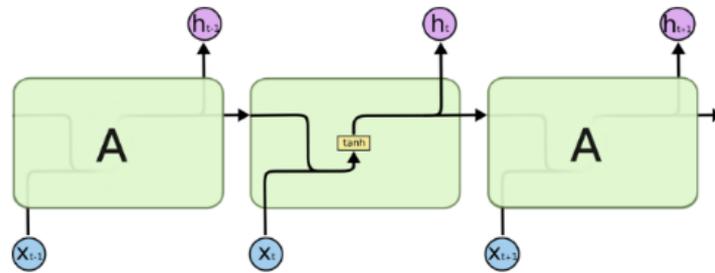


Fig. 2. The standard RNN structure [6]

In the applications of RNN to machine translation (MT), language modeling, and part of speech (POS) tagging, a sentence is taken to be an ordered time series and the typical tasks are about generating the information about the next word given the context words seen so far. Although both MT and POS tagging, the typical applications of RNN, belong to the scope of prediction and forecasting tasks, they are very different from text classification as addressed in this paper. For MT and POS tagging, it usually needs the RNN to produce an out for individual modules as the translation result or tagging information for the input words, but for text classification, it usually only uses the output of the last module for representing the input texts/sentence for classification, and it does not need to produce an output for individual input words. While the RNNs are great for prediction tasks that only need short contexts, they no longer work well for scenarios in which the long-term dependencies need to be remembered due to the intrinsic vanishing gradient problem. As a result, the long-term dependencies cannot be used for learning. Also, it is very time consuming for the RNNs to be convergent. A significantly tweaked version of RNNs, the long short-term memory (LSTM) networks, and the gated recurrent units (GRU) are proposed to cope with the vanishing gradient challenge.

2.3.2 The LSTM Model

The LSTM [7] networks are a special kind of RNNs, targeting to tackle the vanishing gradient problem which prevents from learning the long-term dependencies in the input sequence data, through a gating mechanism. LSTMs have achieved great success in sentiment analysis, text classification, and language translation. As shown in Figure 3,

the LSTM architecture is very similar to that of the RNN. It is a chain of repeating modules, each of which is a modified version of that in RNNs.

By comparing a module at time t in Figure 2 and Figure 3, respectively, we can easily see that the LSTM architecture adds some additional components in each module and these components are called gates with different functionalities to work together so that the long-term dependencies in the data can be considered for learning. The first gate is the forget gate, and it is used to discard some information from the inputs through a sigmoid layer to process $h(t-1)$ and $x(t)$. The output of the forget layer is a vector of values between 0 and 1 to indicate how much the corresponding element in the old cell state $C(t-1)$ can be reserved as the current cell stage. The second gate is the input gate including a sigmoid layer and a tanh layer. The sigmoid layer is used to determine how much information in the inputs can be added to the current cell state $C(t)$, and the tanh layer creates a new vector of values to determine the polarity and proportion for the output of the sigmoid layer to be added to the cell stage. The elementwise product of the outputs of the sigmoid layer and the tanh layer is added to the current cell stage $C(t)$. The third gate is the output gate which includes the filtered cell state $C(t)$ and a sigmoid layer. The output of the sigmoid layer with inputs of $x(t)$ and $h(t-1)$ is used to determine how much we are going to output from the inputs. The filtered cell state through a tanh activation function is to determine the polarity and proportion for each element in the current cell stage for updating the result of the sigmoid layer. Then the result of the filtered cell stage is multiplied with the result of the sigmoid layer in an elementwise way to get the output of the current module as well as the hidden state $h(t)$ for the next module at time $t+1$. Also the current cell stage $C(t)$ is transferred to the next module too.

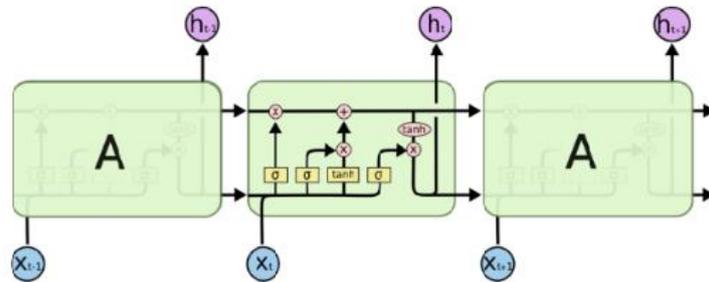


Fig. 3. The LSTM architecture [6]

2.3.3 The GRU Model

The idea of GRUs [8] is like that of the LSTM networks. As outlined by Figure 4, the GRU architecture is a simplified LSTM architecture with only two gates: a reset gate r , and an update gate z . The reset gate determines how to combine the new input with the previous memory, and the upgrade gate determines how much of the information in the previous memory needs to be kept. Compared with the LSTM architecture, the GRU architecture does not have an internal memory $C(t)$ and the output gate. The update gate replaces the input and forget gates in LSTMs, and the reset gate is applied directly to the previous hidden state. The advantage of the GRU over the LSTM is that

it has less number of parameters for learning from the input data in the training process, therefore the learning process is faster.

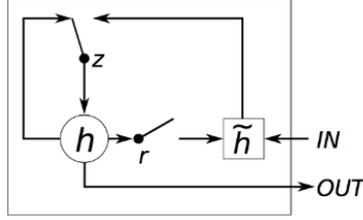


Fig. 4. The GRU architecture [8]

3 Experimental Results and Analysis

We have trained both LSTM and GRU predictive models for churn prediction. To make sure the deep learning models are trained in an optimal way, we annotate 10000 churning and 10000 non-churning phone calls, for which only the caller channels' transcripts are used. These 20000 annotated transcripts are used as the training and validation data, each of which has the equal number of churning and non-churning examples. Furthermore, we have tried different optimization methods, such as the Adam, AdaDelta, and RMSprop. We have found out that the RMSprop optimizer works best for our data set. For GRU, our final settings for the hyperparameters are: batch size 32, hidden size 64, epoch size 50, the dimension of word embeddings 300, dropout 0.5, and L2 regularization lambda 0.7. The LSTM and GRU algorithms are implemented in Python and Tensorflow. Since the training process for both LSTM and GRU is very time consuming for the use of 20,000 training and validation example together with long transcripts, for computational efficiency, we shorten each transcript to be 700 words from the maximum length 7,000 words. This assumes that customers' churn signals are mainly contained in the first 700 words of each transcript. Otherwise the memory requirement is very huge. This assumption is of course not 100% correct, but it greatly helps speed up training the models. In this paper, we use the quantitative metrics of precision, recall, and F1-score to measure the performance of the models, and they are calculated in the following way:

$$Precision = \frac{tp}{tp+fp} * 100 \quad (1)$$

$$Recall = \frac{tp}{tp+fn} * 100 \quad (2)$$

$$F_1_score = 2 * \frac{precision*recall}{precision+recall} * 100 \quad (3)$$

where tp denotes true positives, fp denotes false positives, and fn denotes false negatives. These metrics are multiplied by 100 in this paper for clarity consideration. The higher the metric values, the better the performance of the models.

In the previous work [3], it has been reported that the convolutional neural network (CNN) model [5] with the pre-trained word embeddings outperforms the traditional

machine learning models: the naïve Bayes and SVM [1, 2], for which the bag of word (BOW) based vector space representation is used for document representation. So, in this work, we do not train these traditional machine learning models again and we only train the LSTM and GRU models. Table 1 lists the previous prediction results of the CNN model with 4 different word embeddings for churn prediction using the microblog or Twitter social media data [3]. It is clear to see that the difference of prediction performance between the word embeddings of word2Vec and GloVe is marginal. So, in this paper, we only use the CBOW word embeddings for training our LSTM and GRU models for churn prediction with phone call transcript data. Table 2 lists the previous prediction results based on one of the cutting-edge deep learning technologies: the combination of CNN and different logic rules for distilling the available knowledge into the neural networks [3]. It demonstrates that when the logic rules are added into the CNN model, better prediction performance can be achieved than that without including the logic rules. Table 3 lists our predictive models' performance on 819 testing examples for phone call data analysis. Our LSTM and GRU models use the CBOW word embeddings. By comparing Table 3 with Table 2, it is easy to see that our LSTM predictive model performs favorably in precision and recall with respect to the published best CNN model [3], and our GRU model outperforms the published best CNN model, generating state-of-the-art churn prediction performance. Even though we do not use additional logic rules to distill the knowledge into the neural networks [9], we still get better prediction performance. Our improvement of prediction performance is mainly attributed to the fact that we have much more training examples to train the LSTM and GRU models with the thrust of our scalable data labeling method. Our models are trained with 20,000 training examples, and this greatly reduces the risk of being overfitted for the two models. However, those CNN models in the previous work listed in Table 2 use less than one tenth of our training examples for training. According to the curse of dimensionality, when the CNN model has so many parameters for learning from the training data, it needs a larger number training examples, but the training data is so insufficient, it is hard to say that those CNN models are optimized. This confirms the consensus that for deep learning, the more the data, the better the performance! Furthermore, we have tried to use 80% of the 20000 annotated examples as training and validation examples for training the predictive models, to verify if the 20000 examples are really needed for training. As shown in Table 3, when less training data is used, the models' prediction performance is degraded.

Table 1. The reported churn prediction results of CNN for microblog data analysis using different word embeddings [3]

Input Vectors	F1-Score
Random Embeddings	77.13
CBOW	79.89
Skip-Gram	79.55
GloVE	80.67

Table 2. The reported churn prediction results of CNN + three logic rules for microblog data analysis with/without using word embeddings [3]

Model	F1-Score	Precision	Recall
CNN	77.13	75.36	79.00
CNN+pretrained	80.67	79.28	82.11
CNN+pretrained+“but” rule	81.95	80.84	83.09
CNN+pretrained+“switch from” rule	80.92	79.74	82.14
CNN+pretrained+“switch to” rule	82.60	80.89	84.39
CNN+pretrained+All the 3 rules	83.85	82.56	85.18

Table 3. This paper’s prediction results of the LSTM and GRU models with sufficient training examples of phone call transcripts

Model	F1-Score	Precision	Recall
GRU+pretrained CBOW	86.00	86.00	86.00
LSTM+pretrained CBOW	85.20	85.20	85.20
GRU+pretrained CBOW with 80% of the training examples	0.83	0.83	0.83
LSTM+pretained CBOW with 80% of the training examples	0.84	0.83	0.83

4 Conclusion and Future Work

In this paper, we develop RNN-based deep learning algorithms for the telecommunications industry customer churn prediction through analysis of actual customer phone call data. By comparing with recently published results, our GRU model outperforms the best CNN model in churn prediction. This exciting performance

benefits from much more training examples obtained by using our scalable data labeling method than that used in previous work. This methodology is applicable to all other text classification tasks with deep learning. In the future, we are going to investigate the attention mechanism with bi-directional LSTM and other neural network architectures.

5 Acknowledgment

The authors would like to thank Yu Mao for his assistance in testing some parts of the code of this paper, and Jana Baker's proofreading, which greatly improves the quality of this paper.

References

1. Hadi Amiri and Hal Daume III, Target-dependent churn classification in microblogs, AAAI, pp.2361-2367 (2015).
2. Hadi Amiri and Hal Daume III, Short text representation for detecting churn in microblogs, AAAI, pp.2566-2572 (2016).
3. Mourad Gridach, Churn identification in microblogs using convolutional neural networks with structured logical knowledge, Proceedings of the 3rd workshops on noisy user-generated text, pp. 21-30 (2017).
4. Mikolov Tomas, Chen Kai, Corrado Greg, Dean Jeffrey, Efficient estimation of word representations in vector space, Advances in neural information processing systems, pp. 3111-3119 (2013).
5. Yoon Kim, Convolutional neural networks for sentence classification, Proceedings of the conference on empirical methods in natural language processing, pp. 1746-1751 (2014).
6. Olah C, Understanding LSTM Networks: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> (2015).
7. Hochreiter Sepp, Schmidhuber J., Long short-term memory, Neural computation, Vol.9, No.8, pp. 1735-1780 (1997).
8. Chung, Junyoung, Gulcehre, Cho KyungHyun, Bengio Yoshua. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555v1 (2014).
9. Hu Zhiting, Ma Xuezhe, Liu Zhengzhong, Hovy Eduard, and Xing Eric, Harnessing deep neural networks with logic rules, arXiv preprint arXiv:1603.06318 (2016).

A comparison of linear and machine learning models for the simulation of soil moisture

Milan Cisty^(✉), Frantisek Cyprich and Veronika Soldanova

Faculty of Civil Engineering, Slovak University of Technology in Bratislava,
Bratislava, Slovakia
milan.cisty@stuba.sk

Abstract. This paper focuses on the application of selected machine learning methods and the utilization of remote sensing data sources to simulate soil moisture and obtain a complete time series of its values, assuming the usual situation, i.e., that only data irregularly measured with, for example, weekly or longer time steps between measurements are available. Such a time series is useful for an evaluation of the moisture regime of soil and for decisions regarding building irrigation or drainage structures and for some other practical and research-oriented tasks. The resulting models were verified using data from days on which the field measurements were available. The purpose of this work is to apply the model to data for which the soil moisture measurements were not originally performed. The results show that mainly nonlinear machine learning models are suitable for a solution of the given task. The Extreme Gradient Boosting Machines and Deep Learning Neural Network methods offer the best results. It could be concluded that the method is a suitable alternative for the specification of a time series of soil moisture and for evaluating the water regime of soil.

Keywords: Soil Moisture, Machine Learning, Remote Sensing.

1 Introduction

The term “soil moisture” refers to the water present in the uppermost part of the soil of a field, and its evaluation is necessary for a number of practical applications related to agricultural production, irrigation management, or evaluations of the status of an ecosystem. Soil moisture is influenced by a wide array of ecological, hydrological, geotechnical, and meteorological processes [1]. Along with other variables, such as precipitation, the temperature, solar radiation and evapotranspiration, it plays an important role in the hydrological cycle. The monitoring of soil moisture provides information about the water status of soil and can help in landscape management planning, water resources management, scheduling irrigation, analyses of natural hazards, predictions of agricultural productivity, crop insurance, ecological health, etc. For these reasons, there is a growing need for better soil moisture monitoring techniques.

The measurement of soil moisture by standard methods is carried out by field sampling and its subsequent handling in a pedological laboratory. Soil moisture can be measured by several instruments in the field, e.g., by various types of tensiometers,

neutron probes, time domain reflectometry, capacitance probes or electrical resistance gypsum blocks [2]. Due to time demands, high financial costs, a lack of personnel, and weather fluctuations, such measuring of soil moisture is not usually performed on a daily basis, particularly if it is done in a location where permanent metering equipment cannot be installed.

Therefore, this paper presents an interpolation of soil moisture, i.e., a supplementation of the time series of this variable measured in the field by computed values. The objective is the acquisition of a time series with a daily step between individual values so that the agreement between the calculated and measured data will be as close as possible. This paper focuses on the application of various machine learning methods and the utilization of remote sensing data sources, which helps in obtaining soil moisture data without the presence of measuring personnel in the field.

The sector of the remote sensing of soil moisture has greatly developed in recent years. Soil moisture measurements are based on optical, thermal, and radar satellites, and these methods provide data for convenient spatial coverage [3]. One problem can be that the coarse spatial resolution (pixel size) of satellites usually limits their application to crop field operations. To address such limitations and handle the complex nature of soil moisture dynamics, machine learning tools such as Artificial Neural Networks (ANNs) [4], Support Vector Machines (SVMs) [5], and Random Forest (RF) [6], have been tested in recent years with regard to their ability to estimate spatially-distributed soil moisture.

In this work, a method is proposed that uses various machine learning techniques and different data sources to develop a soil moisture time series, assuming the usual situation, i.e., that only data irregularly measured with, for example, weekly or longer time steps between measurements, are available. In the next part of the paper (“Case Study and Data Description”), the acquisition and preparation of the data is described. The methods applied in this study are then briefly explained. In the “Results and Discussion” section, the settings of the experimental computations are described, and the results are evaluated and discussed. Finally, the “Conclusion” part of the paper summarizes the main achievements of the work.

2 Case Study and Data Description

An area of the Zahorska lowlands was selected for testing the methods described hereinafter. The Zahorska lowlands are located in Slovakia and are situated in the west between the Small Carpathian Mountains and the Morava River. This area has a typical flat relief of a lowland river floodplain. The prevailing soils here are light and medium heavy soils, namely, loamy sands, sandy soils and loam. Three basic data sources were used in this paper:

1. manually measured data using a neutron probe
2. satellite data or data derived from satellite data (evapotranspiration and moisture from the upper level of the soil in locations close to the testing probes – Figure 1),
3. climatic data from an ECA&D dataset

The monitoring of the soil moisture at the selected stations in the Zahorska lowlands was realized using a neutron probe. The locations were selected on the basis of their hydrophysical soil characteristics (soil texture, depth to groundwater level). The monitoring of the soil moisture was carried out approximately two times a month during the growing season (April to October) and once a month in the non-vegetation period (November to March in Slovakia). The method of measuring soil moisture by a neutron probe is an indirect measurement method and is suitable for repeated measurements at the same site. The soil moisture data measured was used for the calibration and testing of the machine learning models that were used in this paper.

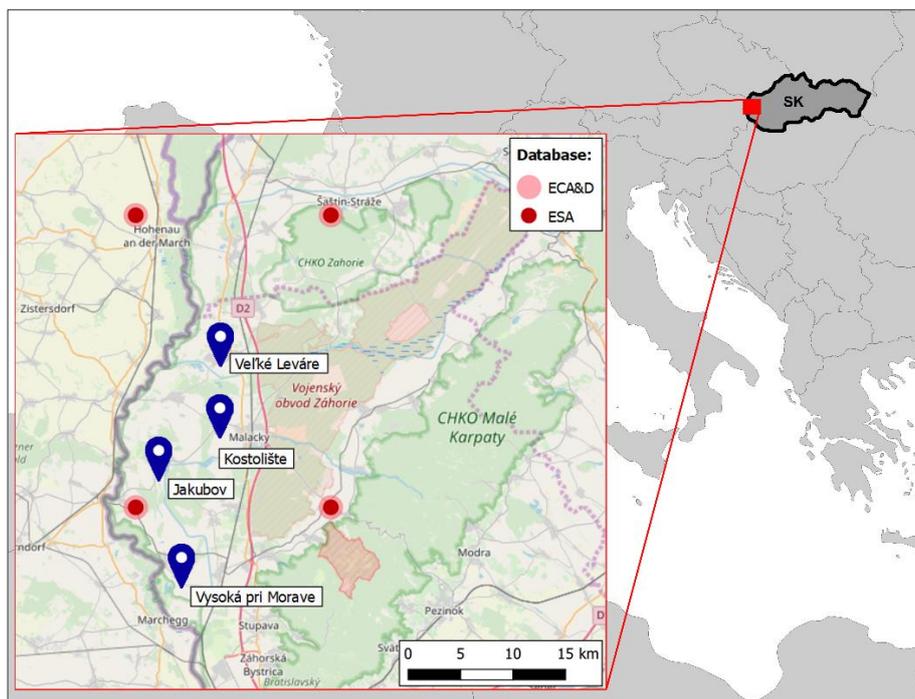


Fig. 1. Location of the measured points, ESA and ECA&D Points

Remotely sensed data was used in addition to the data measured in the field, namely, data about the moisture in the upper level of the soil and evapotranspiration data. Firstly, data on the soil moisture acquired by satellite measurements of the European Space Agency (ESA) was used [7]. The data used was measured by a passive multi-frequency radiometer. The soil moisture data from this source has been available since 1.11.1978 up to the present with a time step of one day. The data is available in the form of a regular grid of points with a distance of 0.25° geographical latitude \times 0.25° geographical longitude. The closest nodes of this grid to the measurement locations of the soil moisture by the neutron probe are shown in Figure 1. This data on the soil moisture differs from the calculated data, as it is from a close, but not identical, location and is related to a different soil depth.

Another variable based on remote sensing techniques and used in this work is evapotranspiration. Evapotranspiration represents the combined loss of soil water from the earth's surface to the atmosphere due to the evaporation of water from the soil or plant surfaces and transpiration via the stomata of the plants [8]. In agricultural production systems, these two losses of water represent a major component of the water balance and are therefore very important processes when estimating soil moisture. Data available from the Global Land Evaporation Amsterdam Model (GLEAM) was used in this work; it is a set of algorithms that estimates the different components of evapotranspiration from remotely sensed data, i.e., transpiration, bare-soil evaporation, interception loss, open-water evaporation, and sublimation. It uses microwave observations, which is an advantage under cloudy conditions. The Priestley and Taylor equation [9] is used in GLEAM for the quantification of the evapotranspiration [10]. Additionally, GLEAM provides data on surface and root-zone soil moisture, which were also used as input data in this work.

The third source of the data are time series of climatic variables. The air temperature has a significantly negative correlation with the soil moisture, while the amount of precipitation has a positive correlation. Climatic data from the European Climate Assessment & Dataset (ECA&D) was used in this paper. ECA&D is currently made up of 69 participating organizations from 63 European countries. The network of basic metering stations from which the data is derived covers the European and Mediterranean regions. It records 12 climatic elements. The main product of this initiative (E-OBS), which was also used in this work, is a daily gridded observational dataset for precipitation, the temperature, and sea level pressure for Europe. The climatic data in ECA&D is provided as a spatial time series in the netCDF [11] format for the period 1950 – 2018, in a spatial scope of 25°N – 75°N to 40°E -75°E, and in a spatial resolution of 0.25°x0.25°.

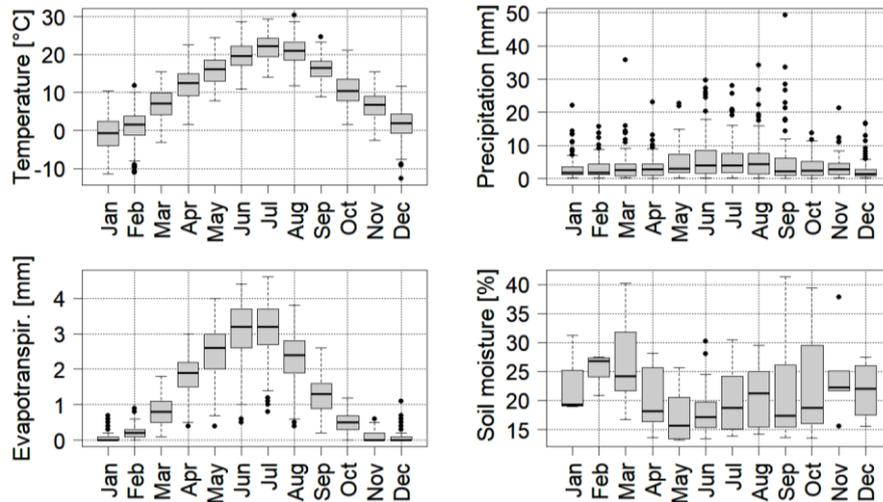


Fig. 2. Overview of the regime of the basic environmental data

3 Methods

In this paper, the soil moisture in the root zone is modelled and is based on the described data. The main objective of the paper was the identification of a suitable method for this task. The methods proposed are briefly described below, together with the reasons for their selection.

Multiple linear regression was used for the comparisons, as it is a standard method used in regression tasks. Multiple linear regression is used for the prediction of output variable Y on the basis of multiple explanatory variables X . The objective of this method is to find a linear relation (mathematical formula) between the variables X and Y .

A major condition for linear regression is that explanatory variables X should not be mutually correlated too highly. However, such a correlation will probably occur in the addressed task; algorithms were therefore used that are more suitable in such a situation. Ridge, Lasso, and Elastic-Net, which we used herein, redefine linear regression to prevent the effect of multicollinearity and help ensure a more stable model. Ridge regression reduces the magnitude of the coefficients. The Lasso regression method regulates the number of variables in the resulting model. Both of these regularization techniques improve the precision and interpretability of the statistical model. Elastic-Net is a combination of the Lasso and Ridge methods [12].

Principal component regression (PCR) was also used to overcome the multicollinearity problem. In PCR, instead of directly using the explanatory variables, the principal components of the explanatory variables are used as independent variables. The central idea of principal component analysis is to reduce the dimensionality of a data set, which often contains a large number of interrelated variables, while retaining as much of the variations present in the data set as possible. This reduction is achieved by transforming the original variables to a new set of variables, i.e., the principal components, which are uncorrelated, and which are ordered so that the first few of which retain most of the variations present in all of the original variables [13].

The previously-described methods are essentially linear. Their advantage is that they are simple to apply and their results interpret well. However, as the collection of soil moisture based on the complicated processes of a hydrological cycle has a strongly non-linear nature, machine learning (ML) regression models were used in this paper. Two models, i.e., XGBoost and Deep Learning Neural Networks, were selected, as they both represent the current state-of-the-art in ML modelling. They are characterised by a high degree of precision in comparison with other regression models. Both models additionally have built-in mechanisms that can handle the multi-collinearity in the input data. Their characteristics are briefly stated below (at the level of acquiring intuition about their functioning, a more detailed description is available in the relevant literature), along with a description of their application.

XGBoost is based on a gradient boosting machine model (GBM). Boosting is a sequential technique that works on the principle of an ensemble. It combines a set of weak learners (usually shallow decision trees) to improve the accuracy of the prediction. Both GBM and XGBoost build a final model in a stage-wise manner through gradually refined estimations. Gradient boosting evaluates the precision of a model in a previous

stage and then develops the next model, the inputs of which are weighted according to the previous results in such a way that the input data from the worst-calculated samples will have a greater weight in the next calculation. The subsequent models are thus mainly focused on the previously inaccurately computed samples. Such "boosting" continues until the desired level of accuracy or a predefined number of iterations (a number of trees) is reached. Although XGBoost basically follow this GBM scheme, there are some important differences and pluses in the modelling details. XGBoost provides parallel tree boosting that helps to solve the problem at hand much faster. XGBoost uses a more regularized model structure to control overfitting, which gives it a better performance and more precise results in comparison with GBM [14]. The price for the advantages of XGBoost is that it has many parameters to tune. A description of the parameters can be found at XGBoost WWW [15].

Deep learning neural networks (DLNN) is the second machine learning model used in this paper. DLNN is a continuation in the development of artificial neural networks that have been devised since the mid-20th century; due to its user-friendly software (SNNS, Genesis, NeuroDimension, applications in Matlab, etc.) and for other reasons, their application in the 1990s recorded a remarkable boom. Later, due to the instability of the results provided (even in the repeated solution of the same task), complicated training and other problems, other machine learning models attracted researchers more, e.g., Support Vector Machines [5], Random Forest [6], Gradient Boosting Machines [14], etc. Deep learning neural networks, which came to the foreground of the AI community at the end of the previous decade, represent the return of artificial neural networks in their "deeper" form (they are characterised by a greater amount of hidden layers). They are capable of processing tasks with much larger data volumes than other algorithms. They currently represent the most suitable alternative for the solution of data-demanding problems involving computer vision, text and image recognition, etc. DLNN has introduced learning from data in such a way that they focus on learning successive layers of a network better, and it better represents the searched-for (computed) target. Unlike the neural networks of the past, modern deep learning provides training stability and good generalizations (with a good set of appropriate parameters). Compared to other models, the DLNN benefits are mainly manifested in problems characterised by large data volumes; however, the improvements of the artificial neural networks implanted in them are also positive for medium-sized problems (regarding the amount of data processed) such as the one addressed in this article. From several software solutions for the design and training of DLNN (i.e., TensorFlow, MXNet, Caffe, Theano, Torch), the version in framework H2O [16] was selected for this paper. H2O follows the model of multi-layer, feed-forward neural networks for predictive modelling. H2O's deep learning functionalities include multi-threaded and distributed parallel computations that can be run on a single or multi-node cluster, an automatic, per-neuron, adaptive learning rate for fast convergence; regularization options such as L1, L2 and/or dropout to prevent overfitting the model; grid search for hyperparameter optimization and model selection; and other advanced features.

Various models frequently show varying abilities in modelling of different aspects of the hydrological processes, so the predictions could be more precise in some part of the problem domain but are less suitable in others. The identification of this fact has led

to the application of an ensemble of models. Many researchers have shown that by combining the output of many predictors, more accurate predictions can be produced than what could be obtained from any of the individual predictors [17]. For this reason, the authors of the present paper assume that it is also important to examine ensembles, which could in some cases eventually offer better results.

A grid search combined with a repeated cross-validation methodology was used to find the parameters of these models. In this approach, a set of model parameters from a predetermined grid is sent to the evaluating algorithm. A set of parameters was generated by the genetic algorithms (GA) (the “chromosome” in GA terminology) and sent to the repeated cross-validation mechanism, which is used for the evaluation of the parameter combinations [18] by developing and evaluating the model with them. N-times repeated k-fold cross-validation is used to find the best parameters for the final model. In the repeated cross-validation, the data set is divided into k subsets, and the training-testing-evaluation is repeated k times. Each time, one of the k subsets is used as the test set, and the rest of the subsets are put together to form a training set. The subdividing of the training data into k-subsets is accomplished n times. The average error across all the k trials is then computed, which is a particular parameter’s “fitness” combination. The chromosome with the best fitness defines the best parameter’s values.

4 Results and Discussion

In this part, we report on the evaluation of the simulation of soil moisture using the models and data previously described. The prediction of soil moisture in the Zahorska Lowlands region in Slovakia (Figure 1) serves as the case study herein. The inputs were shaped into a standard form, e.g., as a table with rows and columns. Each row in the input data includes the date of the soil’s moisture measurement and the measurement value (although these are also the modelled data, they are necessary for the training and testing mechanisms). The soil moisture from the remotely sensed nearby locations from ESA and GLEAM are in the next 5 columns. Remotely-sensed evapotranspiration data from the GLEAM model then follows; finally, the minimal, maximal and average daily temperatures and precipitation from the ECA&D data set are provided.

Simple feature engineering was performed to improve the modelling. As a soil water regime is influenced by the current climate variables as well as by their values from previous days (e.g., the precipitation from preceding days makes soil water saturated for a longer period), we included evapotranspiration, precipitation, and the temperature from twenty days before the date of the predicted soil moisture between the inputs. In addition, with reference to the value which the soil moisture acquires, the exact time of the meteorological event is not the most crucial factor, i.e., it is not decisive whether it rained 13, 14 or 15 days ago. In each of these cases, the impact of the rain on the soil moisture after 2 weeks will be approximately the same. For this reason, aggregated values of the precipitation, evapotranspiration, and the minimum, maximum and average temperatures were also included among the input data. These may be more suitable than daily data because they can lead to a better generalizing model. The aggregation

was performed for precipitation using a summation and for the temperatures and evapotranspiration by an averaging of the values. The aggregation was made for the following intervals of days preceding the date of the prediction of the soil moisture: days 1 – 3, 1 – 5, 3 – 10 and 7 – 20.

The input data includes a total of 136 variables. The number of lines is limited by the amount of field moisture measurements performed with the neural probe from 2009 – 2018, i.e., 129.

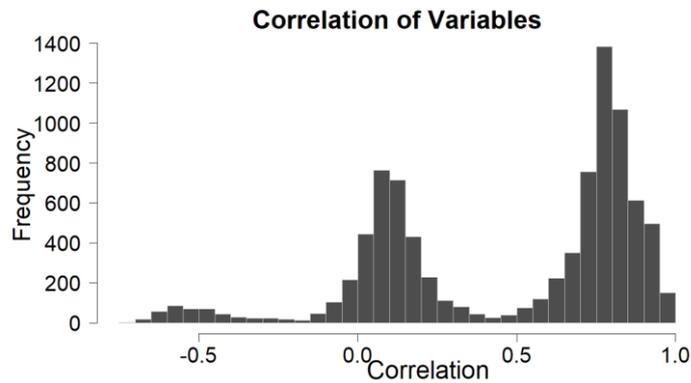


Fig. 3. Mutual correlation of variables

An important feature of this data is that it has a relatively high number of columns compared to the number of lines. Many of the variables in the input data will therefore be highly mutually correlated, as shown in the histogram in Figure 3. These correlated variables are mostly time series of the soil moisture, temperatures and evapotranspiration of consecutive days; there is also a close correlation between the evapotranspiration and temperature values on the same days. The data file thus shows multicollinearity, which is a problem for some methods. The histogram in Fig. 3 also has a distinct top around the 0.1 value, indicating that some variables show almost no correlation, e.g., the evapotranspiration and rain. The smaller top of the histogram can also be seen in this part with negative values; this indicates a negative dependency, e.g., between the values of the soil moisture measured by the satellites and temperatures.

The data was randomly divided into training and testing data sets. The training (calibrating) set was used for all the models to tune their parameters, and the testing set served for an evaluation of the precision of the resulting models.

The calculations were initially performed using multiple linear regression to illustrate the multicollinearity problem and for a comparison. Using this standard method and the above-described inputs, the soil moisture value was calculated and evaluated on the test data set. The comparison with the measured data revealed a correlation coefficient of 0.17, i.e., very inaccurate results. The task solved in this paper is to show various options for acquiring better results by an evaluation of the selected linear and nonlinear methods. All the calculations were performed by setting up the required computer codes in the environment of the statistical computer language R [19].

The LASSO regularisation linear method was used for the reduction of the variables and thus for the removal of the multicollinearity. The glmnet package was used for the calculations [20]. This R package fits LASSO and Elastic-Net regression models using a coordinate descent. For optimisation of the regularisation parameter lambda, which influences the number of variables selected for the resulting LASSO model, a search for the predefined value set (interval 0.3 – 30) was used. Variants of the model obtained with different lambda were evaluated on the training file using a 5-fold cross-validation. The identified optimum value of this regularisation parameter was 6.02. The computed soil moisture is evaluated in Table 1. The resulting model includes 11 variables from the original 136. Elastic-Net was evaluated similarly to LASSO, with the difference that this model requires the optimisation of two parameters, i.e., in addition to lambda, the alpha parameter is also required. Alfa is Elastic-Net's mixing parameter, with values in $(0,1)$, which determines the ratio of LASSO ($\alpha=1$) and Ridge ($\alpha=0$) types of regularization during the training. As shown in the table, this model does not give better results than LASSO, although 18 explanatory variables are selected by this model for the resulting model.

The next method tested for the calculation of the soil moisture in this paper is the principal component regression. As stated above, instead of the original variables, this method uses their linear combinations – principal components (PC) as explanatory variables. The original variables were zero centred and scaled to have unit variance before the analysis was undertaken. R language was used for the specification of the principal components, with the result that 87 principal components were indicated. The first ten, with the proportion of the variance explained, are displayed in Fig. 3. The number of PCs considered in the resulting model were optimised using cross-validation. Four PCs that were identified as further PCs have a very low proportion of their variance explained (Fig. 3). The evaluation of this method using statistical indices is given in Table 1.

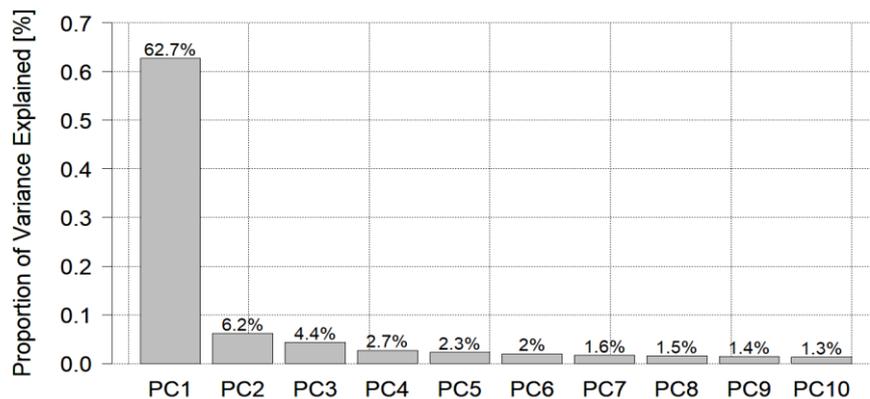


Fig. 4. Principal components with the proportions of variance explained

The calculation of the soil moisture was also performed using machine learning models. They have the potential of capturing nonlinear relations between the explanatory variables and the calculated soil moisture, which can increase the precision of the calculations. The XGBoost model was used first. The optimization of the internal parameters of XGBoost [21] was accomplished by a 10-fold 5 times repeated cross-validation. Seven parameters were tuned, so the genetic algorithms (GA) were applied in the tuning process. Genoud R version of GA was used, which combines evolutionary algorithm methods with a derivative-based, quasi-Newton method [22]. The program can also work without the Newton method. From empirical finding was concluded, that the probability of producing an optimum in a relatively small number of generations tends to increase with the GA population size, so we set population size to 500 and number of generation to 15. However, XGBoost offers quite stable results with differences in optimum RMSE between runs not more than 4%. The evolutionary algorithm in genoud uses nine operators that were set to their default values. The optimized parameters were the maximum number of iterations, the learning rate, the minimum loss reduction gamma, the maximum tree depth, the minimum child weight, the subsample ratio of the training instance, and the subsample ratio of the columns when constructing each tree. A description of the parameters and various recommendations for their settings can be found at XGBoost WWW [15]. The model is evaluated in Table 1.

Table 1. Evaluation of the models

Model	MAE	RMSE	d	r	R ²
LASSO	3.13	3.45	0.62	0.71	0.51
Elastic-Net	2.73	3.00	0.79	0.76	0.58
PCR	2.53	3.12	0.81	0.70	0.49
XGBoost	2.27	2.61	0.89	0.84	0.71
DLNN	2.81	3.10	0.75	0.86	0.75
ensemble	2.49	2.70	0.85	0.86	0.74

MAE - mean absolute error, RMSE – root mean square error, d - coefficient of agreement, r - correlation coefficient, R² – coefficient of determination

The second machine learning model used was the Deep Learning Neural Network. The framework h2o was utilised for the calculations. A 5-fold cross-validation was used for the search of the optimum parameters. The optimised parameters were (after the name of the parameter, its optimised value is in brackets): the number of hidden layers and the number of neurons in them (three layers – 20, 10, 20), the type of activating function (rectifier), the magnitude of the regularisation parameters (both l1 and l2 are equal to 0.01), and the parameter epsilon (1.0e-12) and rho (0.995), which influence the speed of the convergence. A total of 539 combinations of parameters were evaluated. The number of epochs of the development of the neural network was reduced to 50 during the optimisation of the parameters, due to the above-stated number of optimised neural networks (even with this reduction, it ran for 11 minutes on 20 logical cores AMD Ryzen Threadripper 1950X by 64 GB memory). The final calculations with 1000

epochs were computed after the acquisition of the parameters. This number of epochs based on the analysis still did not cause an overtraining of the model; the results are presented in Table 1.

All the models are evaluated in Table 1 using the mean absolute error, RMSE, coefficient of agreement, correlation coefficient, and coefficient of determination.

In this paper, linear and nonlinear models for simulating soil moisture were compared. The benefit of the former is their simple interpretation; the benefit of the latter is the opportunity to capture more complicated relations between the variables investigated. The results indicate that linear modelling should not be abandoned prematurely if the basic multiple linear regression does not show satisfactory results. As stated in Table 1, the more sophisticated methods leading to linear models (LASSO, Elastic-Net, PCR) demonstrated better results, with regard to the multiple linear regression, although not as good as machine learning models. Two best models (DLNN and XGBoost) acquired herein are not equal and therefore do not provide identical results. The dissimilarity of the models can be illustrated by Figure 4 also, where the variable importance of the DLNN and XGBoost models is compared. It can be seen that each model has different preferred variables. The diversity of the models is also shown by the fact that the statistical indicators (Table 1) are not all the best for the same model, three indicators are better for XGBoost and two for DLNN. With approximately the same degree of accuracy of the models, this means that in different conditions (e.g., weather conditions), different models can provide more precise calculations. This points to the suitability of merging models in an ensemble, which was performed by a simple averaging of the XGBoost and DLNN models. The other models were not included in the ensemble due to their lower degree of precision. As shown by the last line of the table, this model gives better balanced results from the point of view of all indicators.

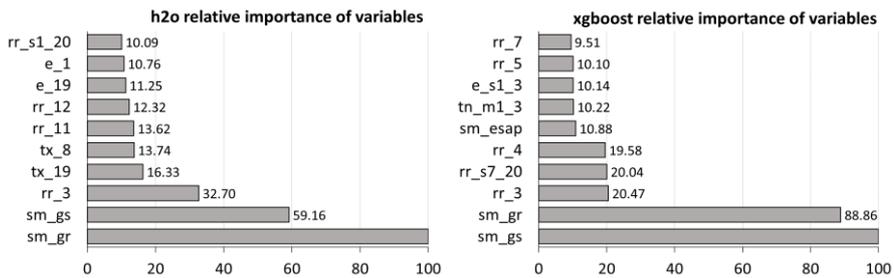


Fig. 5. Comparison of the importance of the first 10 variables of DLNN and XGBoost

5 Conclusion

In this paper, the possibility of using meteorological and remotely-sensed data for the calculation of a complete time series of soil moisture is assessed. Such time series are

useful for the evaluation of the moisture regime of soil or for decisions regarding building irrigation or drainage structures and for some other practical and research tasks. If such data is measured in a specific location by common metering devices, such as tensiometers, neutron probes, time domain reflectometry, etc., such measurements are time demanding and require transport to the location of the measurements, the transport of the samples to the laboratory, etc. If a metering device cannot be permanently installed at a given location (e.g., if a neutron probe is used) and a data logger cannot be used, it is usual that the time series acquired will have longer time steps, which are unsuitable for several reasons.

This task is addressed using data which are described in the article to train (calibrate) models on the parts of the days on which the measurements were performed (training set of data). Several problems related to the multicollinearity of the input data for these models had to be addressed. New variables were constructed (feature engineering) and their contribution to the precision of the results were tested also. The resulting models were verified using different data from days on which the field measurements were available as those which was used for training. The final purpose of this work is to apply the model to data for which the soil moisture measurements were not originally performed.

The results show that some linear statistical models can be used (although not all of them), but mainly nonlinear machine learning models are suitable for a solution of the given task. Extreme Gradient Boosting Machines and Deep Learning Neural Network methods were used. The Extreme Gradient Boosting Machines model and simple ensemble model provided the best results. The comparison of the models is based on several statistical indicators in Table 1. It could be concluded that tested machine learning models are preferred alternative from tested models for the specification of a time series of soil moisture and for evaluating the water regime of soil. Moreover, other models and more sophisticated ensembles can be evaluated for this task in the future with aim of even better results.

Acknowledgments

This work was supported by the Slovak Research and Development Agency under Contract No. APVV-15-0489 and by the Scientific Grant Agency of the Ministry of Education of the Slovak Republic and the Slovak Academy of Sciences, Grant No. 1/0662/19.

References

1. Romano, N.: Soil moisture at local scale: Measurements and simulations. *J. Hydrol.* **516**, 6-20 (2014)
2. Novak, V.; Hlavacikova, H.: Applied Soil Hydrology. In: Springer International Publishing. TATP, vol. 32. Springer, (2018). <https://doi.org/10.1007/978-3-030-01806-1>
3. Petropoulos, G. P., Gareth I., Brian B.: Surface soil moisture retrievals from remote sensing: Current status, products & future trends. *Physics and Chemistry of the Earth, Parts A/B/C.* **83-84**, 36-56 (2015)

4. Maier, H. R., Dandy, G. C.: Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental modelling & software*. **15**(1), 101-124 (2000).
5. Wang, L. (ed.): Support vector machines: theory and applications. In: Springer Science & Business Media, vol. 177, 341 pp. Springer (2005)
6. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
7. Mecklenburg, S., et al.: ESA's soil moisture and ocean salinity mission: Mission performance and operations. *IEEE Transactions on Geoscience and Remote Sensing*. **50**(5), 1354-1366 (2012)
8. Hatfield, J. L., Prueger, J. H.: Spatial and temporal variation in evapotranspiration. In: *Evapotranspiration: from measurements to agricultural and environmental applications*. pp. 1-16, IntechOpen (2011)
9. Priestley, C.H.B., Taylor, R. J.: On the assessment of surface heat flux and evaporation using large-scale parameters. *Monthly weather review*. **100**(2), 81-92 (1972)
10. Martens, B., et al.: GLEAM v3: Satellite-based land evaporation and root-zone soil moisture. *Geoscientific Model Development*. **10**(5), 1903-1925 (2017)
11. Rew, R. K., G. P. Davis, Emmerson S.: NetCDF User's Guide. An Interface for Data Access. 2.3, (1993) <https://www.unidata.ucar.edu/software/netcdf/>
12. Friedman J., Hastie T., Tibshirani R.: Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*. **33**(1), 1-22 (2010)
13. Jolliffe, I.: Principal component analysis. In: *International Encyclopedia of Statistical Science*, pp. 1094-1096. Springer, Berlin, Heidelberg 2014. https://doi.org/10.1007/978-3-642-04898-2_455
14. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. ACM (2016)
15. XGBoost Homepage. <https://xgboost.readthedocs.io/en/latest/>. Accessed: 16 Feb 2019
16. Arora, A., et al.: Deep Learning with H2O. pp. 47. H2O.ai, Inc (2016) <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/booklets/DeepLearningBooklet.pdf>
17. Serafino, F., Pio, G., & Ceci, M. Ensemble Learning for Multi-Type Classification in Heterogeneous Networks. *IEEE Transactions on Knowledge and Data Engineering*, 30(12), 2326-2339 (2018)
18. Hastie, TJ. Tibshirani, RJ. Friedman, JH.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. In: Springer series in statistics, vol. 2009, Springer, New York (2009)
19. R Core Team: R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2017). <https://www.R-project.org/>
20. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*. **33**(1), 1 (2010)
21. Chen, T., et al.: (2015). xgboost: Extreme Gradient Boosting, 2018. R package version 0.71.2 (2018) <https://CRAN.R-project.org/package=xgboost>
22. Mebane Jr, W. R., Sekhon, J. S. Genetic optimization using derivatives: the rgenoud package for R. *Journal of Statistical Software*, 42(11), 1-26, (2011)

Vehicle Classification in Video Using Deep Learning

Mohammad O. Faruque, Hadi Ghahremanezhad, and Chengjun Liu

New Jersey Institute of Technology
Department of Computer Science
Newark, NJ 07102 USA

Abstract. Vehicle classification in videos has broad applications in intelligent transportation and smart cities. The vehicle classes are defined according to the Federal Highway Association (FHWA) vehicle types, and two popular deep learning methods, namely, the Faster R-CNN and the YOLO, are applied for vehicle classification. The Faster R-CNN and the YOLO are two representative deep learning methods with applications in object detection and classification. First, three training data sets are manually created from two videos in the low video quality category for training the Faster R-CNN and the YOLO deep learning methods. Second, new videos that are not seen during training are used to evaluate the vehicle classification performance for the deep learning methods. In particular, the comparative evaluation includes the training time, the testing time, the vehicle classification accuracy, as well as the generalization performance of the deep learning methods. The experiments using the New Jersey Department of Transportation (NJDOT) traffic videos show the feasibility of vehicle classification in videos using deep learning methods.

Keywords: Vehicle classification, intelligent transportation, smart cities, deep learning, Faster R-CNN, YOLO.

1 Introduction

Classifying vehicles in traffic video into different categories has broad applications in intelligent transportation and smart cities. Based on the Federal Highway Association (FHWA) vehicle types, we define six vehicle categories: bike, car, truck, van, bus, and trailer (Fig.1). Vehicle classification in the video, therefore, classifies the vehicles into these predefined six vehicle categories.



Fig. 1. Six types of vehicles used for classification.

There are various challenging issues in vehicle classification from the video due to the camera viewing directions, the illumination variations, and weather conditions. Fig. 2 shows some challenges for vehicle classification. The images in Fig. 2 are all collected from the New Jersey Department of Transportation (NJ-DOT) traffic video sequences. Note that these videos are captured from similar camera angles during day time. In general, there can be more different visual scenarios, and the vehicle classification problem may become more challenging. Changes in environmental conditions, the background of objects, time of the day, occlusions, blur, motion, and camera resolution all make the vehicle classification task more challenging.

We have applied two representative deep learning methods, namely, the Faster R-CNN and the YOLOv3 deep learning methods [23], [22], for vehicle classification in the video. We have used eight traffic videos with the encoding quality of 15fps frame rate and 352x240 spatial resolution to evaluate the vehicle classification performance of deep learning methods. For the training data, we have collected training samples from two traffic videos (video 1 and video 4) in which we manually annotate the vehicles. These training samples are used to define three training data sets corresponding to the two traffic videos: video 1, video 4, and a mixed set of video 1 and video 4. These different data sets thus help us evaluate the comparative generalization performance of the two deep learning methods. For testing, the videos that are not seen during training are applied to evaluate the testing time, the vehicle classification accuracy, as well as the generalization performance of the deep learning methods.



Fig. 2. Some visual challenges in object classification. (a) Various instances of one class with differences in angle, color, size, and other visual attributes. (b) Similar objects from different categories. A good classification method should be able to detect the trivial differences to categorize objects correctly.

2 Background

Object detection and classification has been a popular topic in computer vision and video analysis. Many methods have been published in the literature using statistical methods, such as Support Vector Machine (SVM) [19], efficient SVM (eSVM) [3], clustering-based discriminant analysis [2], the Bayesian Discriminating Features (BDF) method [13], and Adaboost [26]; local features methods, such

as Local Binary Patterns (LBP) [18], Scale Invariant Feature Transform (SIFT) [16], Histogram of Oriented Gradient (HOG) [4], and Feature Local Binary Patterns (FLBP) [9]; neural networks and deep learning methods [24], [14]. More recently, object detection and classification methods based on deep learning have had a good amount of success in many competitions, such as the ILSVRC large scale detection challenge [5], the PASCAL VOC detection challenge [7][6], and the MS COCO large scale detection challenge [12].

The Convolutional Neural Network (CNN) is one of the popular Deep Neural Network (DNN) architectures. DNN usually denotes a feed-forward artificial neural network that has multiple hidden layers between the input and output layers [1].

Convolutional Neural Networks are a specific type of DNNs that are feasible for large input data with locally meaningful connected patterns like images. During each forward pass in a CNN, each convolutional layer extracts features utilizing the learned convolution filters and by updating the weights learned through the training process. The convolution layers are usually followed by activation layers like Rectified Linear Unit (ReLU) [17] (which gets rid of the negative values) and pooling layers.

The Region based CNN (R-CNN) uses region proposals through selective search [25] that applies different window sizes to evaluate the entire image. First, in the selective search process, it extracts around 2000 regions. Then, the R-CNN applies a custom version of the AlexNet [10] to determine a valid region. At the final fully connected layers, it utilizes a number of binary support vector machines to classify the objects.

The Fast R-CNN, which improves upon the R-CNN as R-CNN is slow because it uses a forward pass for each proposed region individually, was proposed in 2015 [8]. The Fast R-CNN is faster than R-CNN because of combining different parts of the process and sharing computations. Since the region proposals of each image have a high overlap with each other, this approach tries to share the convolution calculations along the network layers. In each forward pass, the entire image and all its region proposals are fed to the CNN at the same time. The region proposals of each image share the generated feature maps along the network layers, and thus, the speed of the model improves by reducing the time of computations. For this to happen, the last max-pooling layer of the pre-trained CNN is replaced with a Region of Interest Pooling (RoI-pooling) layer which takes region proposals with different sizes and outputs fixed-length feature vectors.

The Faster R-CNN [23], which improves upon the Fast R-CNN, was introduced in 2016, and it combines the region proposal and the CNN modules [8]. The Faster R-CNN eliminates the "selective search" algorithm and instead uses another network called Region Proposal Network (RPN) to generate object proposals and also learn from the Fast R-CNN network [8].

You Only Look Once (YOLO) deep learning method was introduced in 2015 [20]. In this approach, an input image is divided into an $S \times S$ grid, and each cell in the grid is used to detect only one object (in case there exists one) whose

center falls in that cell. Fig 4 shows the idea of vehicle classification using YOLO. In each cell, a fixed number (B) of bounding boxes with their confidence scores is generated. The confidence scores are calculated by multiplying the probability of each object and their intersection over the union of the predicted box and the ground truth box.

YOLOv2 is an improved version upon YOLO, as the error analysis of YOLO showed that it does not perform well in several cases, such as producing a significant number of localization errors and having relatively low recall compared to region proposal-based methods [21]. YOLOv2 introduces several improvements upon YOLO, such as batch normalization, high-resolution classifier, dimension clusters, direct location prediction, and multi-scale training. Darknet-19 was used as the base classification model of YOLOv2. Like VGG models it mostly uses 3×3 filters, and after every pooling step, it doubles the number of channels. In total, it uses 19 convolutional layers and five max-pooling layers.

YOLOv3 is introduced as an incremental update [22]. To integrate well with the Open Images dataset, it replaces the softmax layers with the independent logistic classifiers and uses binary cross-entropy loss for the class predictions during training. Other changes include using a new classifier, using Darknet-53 instead of Darknet-19, and making detections and classifications in three different scales. These changes help YOLOv3 to identify small objects. Finally, it is claimed that YOLOv3 is three times faster than the SSD method with similar accuracy to the SSD [15] and the RetinaNet [11] models.

3 Vehicle Classification in Video Using Deep Learning

We apply two representative deep learning methods, the Faster R-CNN, and the YOLOv3, for vehicle classification in the video. The training data sets are manually created using two NJDOT traffic videos. Specifically, three training data sets are created corresponding to the training samples from video 1, video 4, and a mixed set of video 1 and video 4, respectively.

3.1 Vehicle Classification in Video Using the Faster R-CNN Deep Learning Method

We apply the Faster R-CNN deep learning method for vehicle classification in the video. The VGG16 network is used with the Faster R-CNN deep learning method, and Fig. 3 shows the system architecture of the Faster R-CNN model. Note that the Region Proposal Network (RPN) network works proportionally to the Fast R-CNN network and uses it to generate better region proposals in the process of training [8]. In this version, the RPN is fine-tuned by pre-trained convolution network on image classification task. In this phase, positive samples are the ones with the Intersection over Union (IoU) more than 0.7, and negative samples are the ones with IoU less than 0.3, and the rest of the samples stays ignored. In this module, a small $n \times n$ (by default $n=3$) window is slid over the feature map of the entire image. At the center of each sliding window, nine

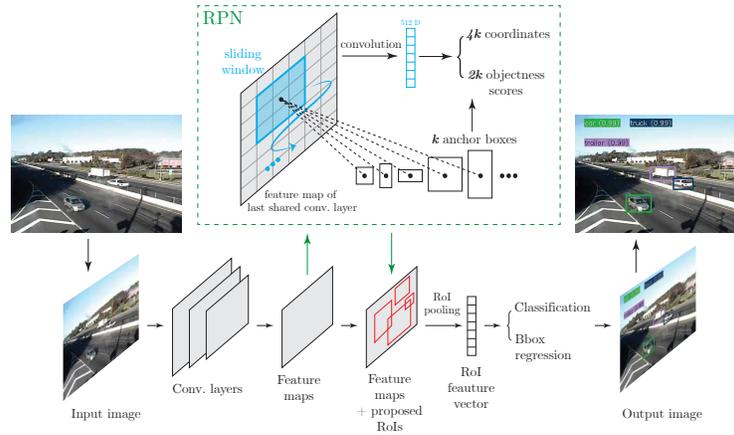


Fig. 3. The system architecture of the Faster R-CNN model. The RPN network works proportionally to the Fast R-CNN network and uses it to generate better region proposals in the process of training.

anchors with three different scales and three different ratios are generated. Then each anchor is fed to classifiers and bounding box regressors to be classified as foreground or background and also to be refined. These anchors are the region proposals that are used to train the Fast R-CNN model, and simultaneously, the output of the Fast R-CNN is used to initialize the training of the RPN. These two networks share convolutional layers, and therefore, this approach becomes faster than the Fast R-CNN with the selective search. After RPN generates the region proposals, these regions have different sizes, and they are fed to CNN after RoI-pooling.

In particular, the VGG16 network has a total of 13 convolutional layers, five max-pooling layers, three fully connected layers, and one softmax classification layer. The size of convolution filters applied to the feature maps is by default 3×3 , and the size of max-pooling layers is 2×2 . The rectification non-linear activation is applied to all the hidden layers of this network. In the pooling layers, the act of down-sampling is done to reduce the size of the input feature map to produce more robust features. Down-sampling helps the model to eventually get to a vector containing class scores at the end of the convolutional network. The final fully connected layers are responsible for calculating the score for each class and generating the output. Each neuron in these layers is connected to all the neurons from the previous map. The fully connected layers are usually adjusted for different vision tasks.

3.2 Vehicle Classification in Video Using the YOLOv3 Deep Learning Method

We apply the YOLOv3 deep learning method for vehicle classification in the video. Note that we have initially applied the full YOLOv3 for vehicle classi-

fication in the video, but it was not able to achieve real-time performance. To improve the computational efficiency, we apply the "tiny" configuration, which uses 13 convolutional layers instead of 75 layers, and six max-pool layers. In particular, the input image is first divided into an $S \times S$ grid. Note that each cell in the grid is used to detect only one object (in case there exists one) and the center of the object falls in that cell.

Fig. 4 shows the idea of our vehicle classification in a video using the YOLOv3 deep learning method. Note that among the many bounding boxes generated in one single network, the ones with the highest scores are chosen as the final results. [20]. In each cell, a fixed number (B) of bounding boxes with their confidence scores is generated. The confidence scores are calculated by multiplying the probability of each object and their "intersection over union" of the predicted box and the ground truth box. Each bounding box is indicated by five numbers: a quadruple (x, y, w, h) , and the confidence score of the box. X and Y are the coordinates of the center of the box, and w and h are the width and height of the box, respectively. These four numbers are float values relative to the absolute width and height of the image, and they can be somewhere between 0.0 and 1.0. The confidence score indicates the likeliness of the box containing an object. Each grid cell contains conditional class probabilities for the number of different classes, and therefore, for each category of objects, there is one probability in each cell, regardless of the value of B . Note that the conditional class probability means that the probability of the object belonging to a specific class is conditioned on the box containing an object. Thus, for each grid cell, there are $B \times 5$ numbers indicating the bounding box information and the C class probabilities. This prediction information is encoded as a tensor in the shape of $(S, S, B \times 5 + C)$.

Like R-CNN, the non-maximum suppression algorithm is used in YOLO to ignore the repetitive bounding boxes around the same object and to consider the box with the highest score value.

We use eight NJDOT traffic videos for evaluating both the Faster R-CNN deep learning vehicle classification method and the YOLOv3 deep learning vehicle classification method, which are trained using the three training data sets, respectively. In the process of evaluation, we consider two types of false positive (detecting background as a vehicle and misclassification) along with one false negative (missing data in the region of interest). We also count the ground truth or the unique counting to derive the success rate. We use the following formulas to calculate the success rate of the classifications:

$$ER_{(Error\ rate)} = 100 \times \frac{FP_1 + FP_2 + FN}{GT_{(ground\ truth)}} \quad (1)$$

$$Classification\ success = 100 - ER \quad (2)$$

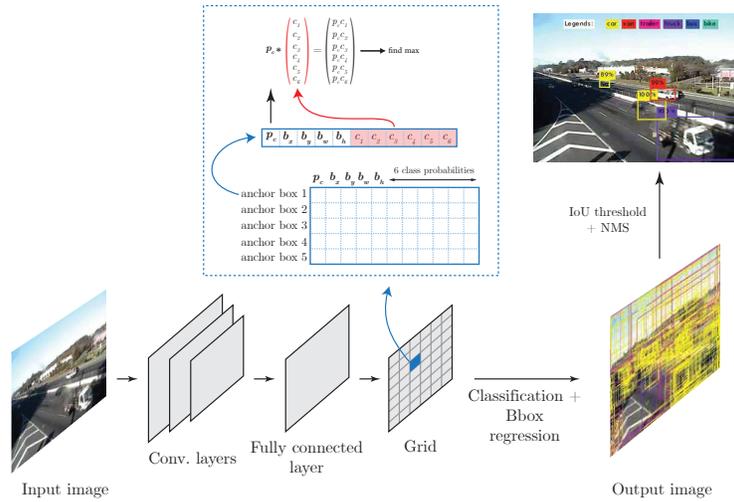


Fig. 4. Vehicle classification in the video using the YOLOv3 deep learning method. Among the many bounding boxes generated in one single network, the ones with the highest scores are chosen as the final result.

4 Experiments

Experiments are implemented using eight NJDOT traffic videos to evaluate the Faster R-CNN deep learning vehicle classification method and the YOLOv3 deep learning vehicle classification method. We first create three training data sets using two videos, video 1 and video 4. Specifically, all the 13,500 frames of the first video, video 1, are used for defining the training data set 1. For each training frame, the ground-truth bounding boxes are manually labeled. The training data set 2, however, contains only 1,227 frames by selecting only one frame from every 10 frames of video 4. These frames are further labeled manually for the vehicles and their corresponding class information. The training data set 3 includes all the 14,727 frames set 1 and set 2 to define a mixed training model. Table 1 summarizes the three training data sets.

Table 1. Three Training Data sets

	YOLO	Faster RCNN
Source (15fps, 352x240)	Reason for selection	Annotated images
Video 1	Uniformity of illumination	13500
Video 4	Illumination variance	1227
Videos 1 & 4	Mixture of data sets	14727

The first set of experiments we carry out applies the training data set 1. For training, we have used Nvidia GTX-745 GPU. In our test, YOLO has faster training and testing rate, unlike Faster R-CNN. For testing, Table 2 shows the average testing time for both methods. Note that each video is 15 minutes in length. The results in Table 2 indicate that YOLO is much faster than Faster R-CNN.

Table 2. Average testing time from eight videos corresponding to the three training data sets

Data Set 1		Data Set 2		Data Set 3	
YOLO	Faster R-CNN	YOLO	Faster R-CNN	YOLO	Faster R-CNN
6m2s	2h40m38s	6m4s	2h46m24s	6m5s	2h44m35s

Table 3. Vehicle classification results using the training samples from Video 1

Video	Unique counting	FP (BG as Vehicle)	FP (Mis-classification)	FN (Miss in ROI)	Classification Success	Errors
Detector →		YOLO				
Video 1	821	3	1	0	99.51%	0.49%
Video 2	881	2	23	3	96.82%	3.18%
Video 3	1057	15	23	0	96.40%	3.60%
Video 4	960	10	0	0	98.96%	1.04%
Video 5	1008	10	15	0	97.52%	2.48%
Video 6	988	9	24	0	96.66%	3.34%
Video 7	1017	6	10	0	98.43%	1.57%
Video 8	1148	5	26	1	97.21%	2.79%
Detector →		Faster RCNN				
Video 1	821	0	11	1	98.54%	1.46%
Video 2	881	1	22	2	97.16%	2.84%
Video 3	1057	2	35	8	95.74%	4.26%
Video 4	960	3	46	25	92.29%	7.71%
Video 5	1008	4	65	41	89.09%	10.91%
Video 6	988	3	49	43	90.38%	9.62%
Video 7	1017	2	54	52	89.38%	10.62%
Video 8	1148	3	74	44	89.46%	10.54%

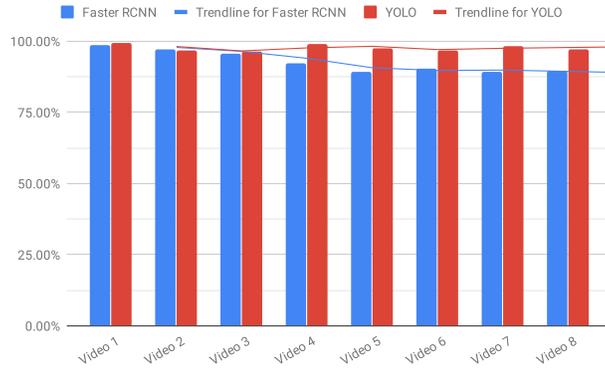


Fig. 5. Classification Success chart for YOLO and Faster R-CNN based on video 1 data

The training data set 1 from video 1 contains almost uniform illumination. The testing vehicle classification results on all the eight videos are shown in Table 3 and Fig. 5. Note that for video 1, as all the frames have been used for training the deep learning algorithms, the testing performance on video 1 resembles that of the training. The other seven videos are never seen during training. Hence the vehicle classification results on these seven new videos demonstrate the generalization performance of the deep learning methods. The results in Table 3 and Fig. 5 thus reveal that YOLO generalizes well across all the eight videos and it performs better than the Faster R-CNN.

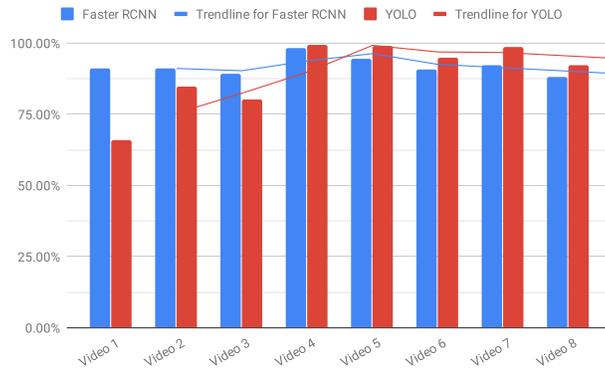


Fig. 6. Classification Success chart for YOLO and Faster R-CNN based on video 4 data

Table 4. Vehicle classification results using the training samples from Video 4

Video	Unique counting	FP (BG as Vehicle)	FP (Misclassification)	FN (Miss in ROI)	Classification Success	Errors
Detector →		YOLO				
Video 1	821	8	34	236	66.14%	33.86%
Video 2	881	2	60	73	84.68%	15.32%
Video 3	1057	1	28	181	80.13%	19.87%
Video 4	960	0	5	0	99.48%	0.52%
Video 5	1008	1	9	0	99.01%	0.99%
Video 6	988	2	48	1	94.84%	5.16%
Video 7	1017	3	11	0	98.62%	1.38%
Video 8	1148	10	75	2	92.42%	7.58%
Detector →		Faster RCNN				
Video 1	821	5	53	16	90.99%	9.01%
Video 2	881	2	55	20	91.26%	8.74%
Video 3	1057	1	80	31	89.40%	10.60%
Video 4	960	0	15	3	98.13%	1.88%
Video 5	1008	1	44	10	94.54%	5.46%
Video 6	988	2	73	18	90.59%	9.41%
Video 7	1017	2	64	14	92.13%	7.87%
Video 8	1148	1	98	36	88.24%	11.76%

The second set of experiments we implement utilizes the training data set 2. The training data set 2, which has about 10% of the frames from video 4, is designed to investigate the generalization capability by using only a fraction of the training frames in a video. In contrast to the training data set 1 where all the frames from video 1 are used for training, the training data set 2 will reveal the generalization performance on all the eight videos including video 4. In addition, the training data set 2 contains more challenging factors for evaluating the generalization performance, such as the illumination differences due to shadows and sun lights.

The vehicle classification performances on the eight videos are shown in Table 4 and Fig. 6. Note that the vehicle classification result on video 4 is not the training performance because not all the frames from video 4 are used for training. As the illumination variations tend to increase from video 1 to video 8, the vehicle classification results in Fig 6 show that both YOLO and Faster R-CNN display deteriorate generalization performance.

Table 5. Vehicle classification results using the training samples from Video 1 & 4

Video	Unique counting	FP (BG as Vehicle)	FP (Misclassification)	FN (Miss in ROI)	Classification Success	Errors
Detector →		YOLO				
Video 1	821	0	2	0	99.76%	0.24%
Video 2	881	0	35	0	96.03%	3.97%
Video 3	1057	0	24	0	97.73%	2.27%
Video 4	960	0	10	0	98.96%	1.04%
Video 5	1008	0	19	0	98.12%	1.88%
Video 6	988	0	37	1	96.15%	3.85%
Video 7	1017	0	21	0	97.94%	2.06%
Video 8	1148	0	34	3	96.78%	3.22%
Detector →		Faster RCNN				
Video 1	821	0	13	4	97.93%	2.07%
Video 2	881	4	16	1	97.62%	2.38%
Video 3	1057	1	17	1	98.20%	1.80%
Video 4	960	0	6	0	99.38%	0.63%
Video 5	1008	2	21	2	97.52%	2.48%
Video 6	988	2	28	4	96.56%	3.44%
Video 7	1017	3	29	5	96.36%	3.64%
Video 8	1148	4	41	2	95.91%	4.09%

**Fig. 7.** Classification Success chart for YOLO and Faster R-CNN based on video 1, 4 data

The third set of experiments we carry out uses the training data set 3. The training data set 3 is a mixed data set containing frames from both video 1 and video 4. The vehicle classification performances on the eight videos are shown in Table 5 and Fig. 7. As more training data is used for training the deep learning algorithms, the vehicle classification results show that both YOLO and Faster R-CNN achieve better generalization performance. In particular, YOLO achieves better classification success rates than Faster R-CNN.

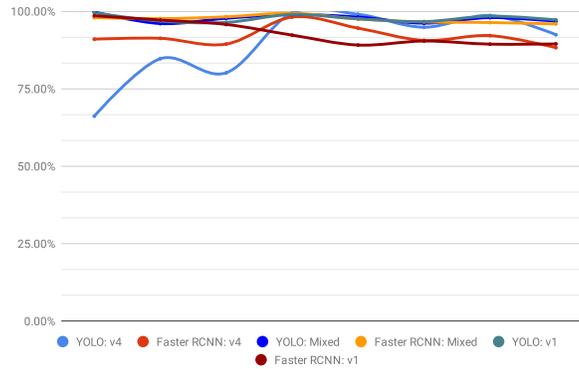


Fig. 8. Comparison of classification success chart for various data-sets

Fig. 8 illustrates the classification success rate using various training data sets. This figure essentially suggests that the training models with mixed images captured at different times of day with illumination variances help improve the generalization performance of the YOLO and the Faster R-CNN deep learning methods.

Fig. 9 shows some example testing results of the YOLO and the Faster R-CNN deep learning methods using the NJDOT traffic videos that are not seen during training. Specifically, Fig. 9 (a) and (c) show the vehicle classification results by using the YOLOv3 deep learning method, and Fig. 9 (b) and (d) show the vehicle classification results by using the Faster R-CNN deep learning method. Fig. 9 reveals that the vehicles in all the frames are correctly classified without any false classification, and the YOLO method detects the vehicles more precisely (in tighter bounding boxes) than the Faster R-CNN method.

Through our research, we find that the deep learning models need a considerable amount of data and computational resources. The process of annotating and preparing necessary data to train a deep learning model is time-consuming. Besides, the complexity of multiple hidden layers in a deep network makes the interpretation and parameter configuration difficult. One of the main observations in this research is the limited ability of the deep learning methods in generalization. When a deep neural network is trained on specific data, which is gathered by intensive efforts, it will perform well on the same data in testing. However, when facing new slightly different data, even if it resembles the training data, the performance drops considerably. This is mainly because of the lack of reasoning and understanding the data, and pure dependence on experience and a large number of iterations. In the case of unfamiliar situations, the performance of deep learning models is not comparable to human performance. Observations from this research show that even the slight alternations in the visual situation, e.g., the presence of shadow or change of size and resolution makes the model less accurate. If the same model is used to detect vehicles in a more different situation, the results are expected to be even less accurate. For instance, changes

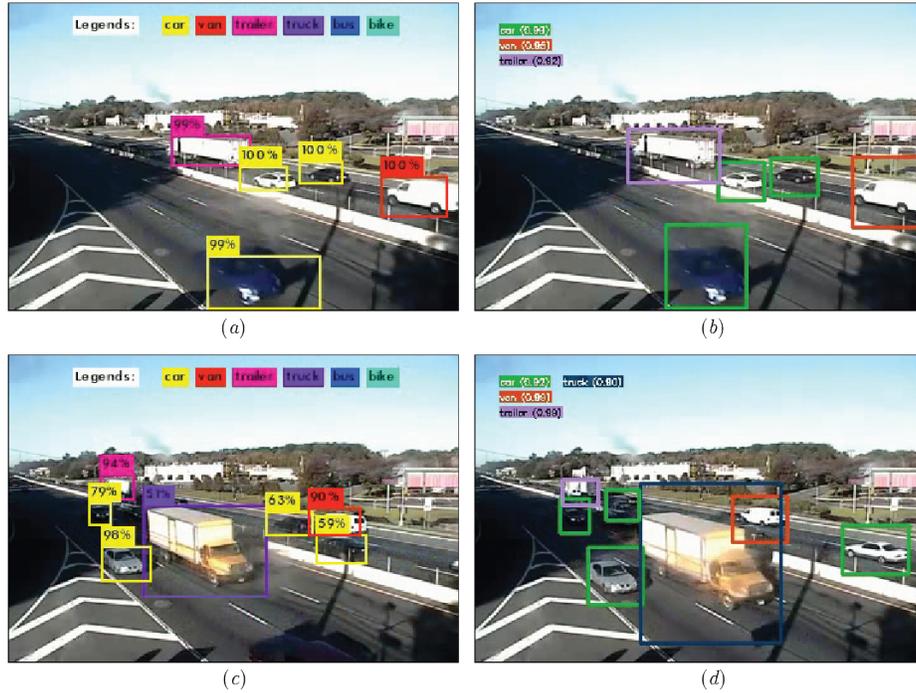


Fig. 9. Example testing results of the YOLO and the Faster R-CNN deep learning methods using the NJDOT traffic videos that are not seen during training. The frames in (a) and (c) show the vehicle classification results by using the YOLOv3 deep learning method. The frames in (b) and (d) show the vehicle classification results by using the Faster R-CNN deep learning method.

in the angle of the camera overlooking the highway, lightening in different times of a day, and weather situations can result in more false positives, false negatives, and misclassifications.

YOLO is much faster in comparison with Faster R-CNN, and in case of their use for object detection and classification in video data, YOLO can process frames about 30 times quicker than Faster R-CNN. This property makes YOLO applicable for real-time detection in a video. The end-to-end training in a single network also improves the accuracy of this method. Another benefit of YOLO over methods based on region proposals is the fewer number of false positives, which is to detect a part of the background as an object. This is because YOLO sees the entire image as a whole and therefore, has a better outlook on the image. Newer versions of YOLO like YOLO9000 and YOLOv3 have improved the accuracy and mostly the speed of the original approach. The improvements are the cause of some modifications like training the model on multiple datasets and at multiple scales, using anchors to perform classification per anchor box instead of each grid cell, and pre-training on ImageNet at multiple scales.

5 Conclusions

Two representative deep learning methods, the Faster R-CNN, and the YOLO are applied for vehicle classification in videos, which has broad applications in intelligent transportation and smart cities. According to the Federal Highway Association (FHWA) vehicle types, six vehicle classes are defined: bike, car, truck, van, bus, and trailer. The training data sets are manually created from two New Jersey Department of Transportation (NJDOT) traffic videos and three training data sets are formed from video1, video 4, and both videos, respectively. New NJDOT traffic videos that are not seen during the deep learning network training are used to evaluate the vehicle classification performance in video. Using different training data sets, the Faster R-CNN and the YOLO deep learning methods display different performance in terms of the training time, the testing time, the vehicle classification accuracy, and the generalization performance. The experiments show the feasibility of vehicle classification in videos using deep learning methods and reveal that the YOLO deep learning method is much faster than the Faster R-CNN deep learning method. Some limitations of the deep learning methods, such as the generalization performance, are discussed in the paper as well.

Acknowledgments: We are grateful to the anonymous reviewers, whose comments and suggestions help improve the quality of the paper. This work is partially supported by the NSF grant 1647170.

References

1. Bengio, Y., et al.: Learning deep architectures for ai. *Foundations and trends® in Machine Learning* **2**(1), 1–127 (2009)
2. Chen, S., Liu, C.: Clustering-based discriminant analysis for eye detection. *IEEE Transactions on Image Processing* **23**(4), 1629–1638 (2014)
3. Chen, S., Liu, C.: Eye detection using discriminatory haar features and a new efficient svm. *Image and Vision Computing* **33**, 68–77 (2015)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. vol. 1, pp. 886–893. IEEE (2005)
5. Deng, J., Berg, A., Satheesh, S., Su, H., Khosla, A., Fei-Fei, L.: Imagenet large scale visual recognition competition 2012 (ilsvrc2012). See net.org/challenges/LSVRC (2012)
6. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. *International journal of computer vision* **111**(1), 98–136 (2015)
7. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. *International journal of computer vision* **88**(2), 303–338 (2010)
8. Girshick, R.B.: Fast R-CNN. *CoRR* **abs/1504.08083** (2015), <http://arxiv.org/abs/1504.08083v1>

9. Gu, J., Liu, C.: Feature local binary patterns with application to eye detection. *Neurocomputing* **113**, 138–152 (2013)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. pp. 1097–1105 (2012)
11. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. *CoRR* **abs/1708.02002** (2017)
12. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: *European conference on computer vision*. pp. 740–755. Springer (2014)
13. Liu, C.: A bayesian discriminating features method for face detection. *IEEE transactions on pattern analysis and machine intelligence* **25**(6), 725–740 (2003)
14. Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M.: Deep learning for generic object detection: A survey. *arXiv preprint arXiv:1809.02165* (2018)
15. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: *European conference on computer vision*. pp. 21–37. Springer (2016)
16. Lowe, D.G.: Object recognition from local scale-invariant features. In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. vol. 2, pp. 1150–1157. Ieee (1999)
17. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. pp. 807–814 (2010)
18. Ojala, T., Pietikäinen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence* **24**(7), 971–987 (2002)
19. Osuna, E., Freund, R., Girosit, F.: Training support vector machines: an application to face detection. In: *Computer vision and pattern recognition, 1997. Proceedings., 1997 IEEE computer society conference on*. pp. 130–136. IEEE (1997)
20. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 779–788 (2016)
21. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. *arXiv preprint* (2017)
22. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018)
23. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR* **abs/1506.01497** (2015), <http://arxiv.org/abs/1506.01497v1>
24. Rowley, H.A., Baluja, S., Kanade, T.: Neural network-based face detection. *IEEE Transactions on pattern analysis and machine intelligence* **20**(1), 23–38 (1998)
25. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. *International journal of computer vision* **104**(2), 154–171 (2013)
26. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. vol. 1, pp. I–I. IEEE (2001)

A Model for Diagnosis of Alzheimer's Disease Using Haralick Texture Features and Meta Feature Selection

Iago Richard Rodrigues Silva, Marília Nayara Clemente de Almeida Lima, Wellington Pinheiro dos Santos, and Roberta Andrade de Araújo Fagundes

Department of Computer Engineering, University of Pernambuco,
Recife, Pernambuco, Brazil
{irrs,mncal,wps}@ecomppoli.br, roberta.fagundes@upe.br

Abstract. Alzheimer's disease (AD) is a neurodegenerative disease that results in the compromise of behavioral and intellectual skills, such as memory decline, language, and perception of the patient. Machine Learning has been a powerful method for AD diagnosis. In this work we propose a model for multi-class AD diagnosis based extraction of Haralick texture features and the use of the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) to feature selection for the classification using magnetic resonance imaging (MRI). This model aims to classify AD vs. MCI (Mild Cognitive Impairment) vs. HC (Healthy Controls). The database used is the Alzheimer's Disease Neuroimaging Initiative (ADNI). We select twenty-one slices from the cerebral ventricle to apply the feature extraction algorithm. The Algorithm Synthetic Minority Over-sampling Technique (SMOTE) is applied to balance the database classes. The data are partitioned with the 10-folds cross-validation method. We used Random Forest Algorithm to perform the model. The results of accuracy are 0.7890 using GA, and 0.7766 using PSO. We can prove the efficiency of the model for the diagnosis of Alzheimer's disease using a multiclass approach, comparing with state of the art methods.

Keywords: Alzheimer's Disease Diagnosis · Haralick Texture Features · Meta Feature Selection.

1 Introduction

Alzheimer's disease (AD) is one of the most common neurodegenerative disorders. It affects the elderly due to changes that occur in the brain. The disease can compromise behavioral and intellectual abilities, such as memory decline, language, and perception [11]. AD is closely related to aging and affects the elderly. There is no cure

* Supported by Coordination for the Improvement of Higher Education Personnel (CAPES). Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf

for Alzheimer. However, when the disease is diagnosed early and adequately, existing treatments can soften the individual's quality of life [11].

Several anatomical characteristics evidence a patient that contains AD. There are anatomical alterations that are associated with AD, such as atrophy of cortical regions of the brain and hippocampus region, in contrast to ventricular enlargement. [11]. All of these can be detected by imaging analysis such as magnetic resonance imaging (MRI) and positron emission tomography (PET-SCAN).

Currently, several Machine Learning approaches have been used for the diagnosis of AD. Generally, computer vision techniques are used to obtain quantifiable values of MRI images. After that, the data are statistically analyzed and used in the Machine Learning algorithms. There is a way without using computer vision when the works propose a deep learning approach. Usually, greater number of previous works use supervised techniques, while sizable effort of papers are using unsupervised techniques. Models based on Neural Networks Multilayer Perceptron [14], SVM [26], or Random Forest [15]. The classes used following the AD cycle, such as HC (Healthy Controls), MCI (Mild Cognitive Impairment), and AD.

The value of accuracy found in the state of the art for binary classifications of AD (AD vs. HC or AD vs. ICM) exceeds 90%. Recently, the multiclass approach (AD vs. MCI vs. HC) has been the subject of investigations by researchers [5]. However, the data have similar characteristics, and this causes some confusion in the learning of the algorithms. Several approaches in recent literature have the objective of solving this problem. Also, when an unbalanced database is used in multiclass approaches, learning tends to overfitting.

Given this context, our goal in this work is to perform the diagnosis of Alzheimer's disease using a multiclass approach without overfitting. The database used in our method is the Alzheimer's Disease Neuroimaging Initiative (ADNI) [10]. For the feature extraction we used Haralick Textures [8]. With the objective of reducing overfitting and projecting better learning, we used the Synthetic Minority Over-sampling Technique (SMOTE) algorithm [3]. Finally, we performed an evaluation and selection of the generated features through the Haralick technique. We used Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) algorithms to select the most relevant features for classification. The model was evaluated using the Random Forest algorithm.

The rest of the paper is organized as follows. Section 2 presents the background in AD classification. Section 3 shows the methods used to perform this work. Section 4 presents the experiment settings, results, and discussion. Finally, section 5 presents the conclusions and proposals for future work.

2 Related Works

This section presents some recent work published in the area of AD diagnosis using Machine Learning. The first subsection presents a binary classification approach, then the second subsection presents a multiclass approach.

2.1 Binary AD Classification

Several works present in the literature address the classification of Alzheimer's disease. A good part of these is proposed to present a binary classification approach, such as AD vs. MCI and AD vs. HC. The purpose of these works is to identify characteristics that evidence the progression of Alzheimer's disease in patients. Due to the fact of using two classes in this process, the average accuracy reached is around 90% [7] [24] [13]. Much of this work follows a methodology regarding the way of identifying patterns in images. The method usually consists of image acquisition, pre-processing, extraction, and selection of characteristics and application of machine learning algorithm.

In general, the pre-processing of images consists of the application of basic techniques of Digital Image Processing. Some works include the investigation of form extraction [18], being necessary the application of segmentation of images for this purpose. While other works aim at texture extraction [12], being the only conversion required to grayscale. The way these features are extracted is critical to the machine learning process. Because the learning algorithms will be executed to give the diagnosis. About the learning algorithms, SVM [22] and Random Forest [4] are the most used in the literature for diagnosis of Alzheimer's disease.

2.2 Multiclass AD Classification

In contrast, there is also another way to the classification of AD, using a multiclass approach. It's recent and consists of the use of the three classes that define the cycle of Alzheimer's disease (AD, MCI, and HC) [5]. The classification method used is all vs. all. The problem with this approach is that it causes data confusion. This produces a poor quality in learning intelligent algorithms. This fact is evidenced in the literature, and next we present some works that use this approach.

The work in [23] presents a new approach to AD classification using Graphs. The method consists of extracting features from pixel intensity values of segmented regions. The brain regions used are that part of the gray matter volume. After that, the data are normalized and used in the proposed classifier. The classifier is named Nonlinear Graph Fusion (NGF), which consists of unifying MRI, PET-SCAN and cerebrospinal fluid (CSF) data, using graph theory. The result of the accuracy of the model was 0.6020. It was superior to other approaches that were used in the comparison.

In another work [2] is presented an AD classification approach using Artificial Neural Networks (ANN) is presented. The ANN uses linear projections on the data to reach discriminatory spaces. With this, the model allows a greater generalization in the task of classification and a higher rate of correctness in the MCI class. For extracting features, the authors used software that provides information about the cortical region, such as area, volume, etc. The result of accuracy obtained through the model was 0.709.

The work in [5] aims to extract textures from MRI images. Two distinct models are presented, both using a Stacked Auto-Encoder (SAE) approach. The model consists of the learning in the atrophy in regions such as white and gray matters, cortical thickness, among others. The texture extraction step is based on the Fractal Di-

mension Co-occurrence Matrix approach (FDCM). The result of accuracy obtained through both models was 0.7330.

In another work [9] a Convolutional Neural Network (CNN) network model for AD classification is presented. This model consists of the application of the CNN network directly in the images, without the feature extraction step [21]. Filters are applied to the convolutional layers, and the network changes filters according to learning. These filters learn the main characteristics relevant to the classification step. Finally, softmax regression is used to classify the data. The result of accuracy obtained through CNN is 0.7375.

In this context, there are some highlights in the model we propose. The first point is the use of Haralick texture features is not a field explored in the literature. This extractor is used in several applications but has not yet been applied for Alzheimer's diagnosis. We hope that, as in other applications, this extractor may provide good results for the diagnosis of Alzheimer's disease in MRI slices. Pathological features to be used in this work in the feature extraction step can be quantified through texture extractors. Because of this, we justify the possibility of gain in results. The second point is the use of intelligent algorithms to reduce the number of features to use in Machine Learning algorithms. Although this point has already been consolidated in the literature, we hope that the use of these techniques can provide good accuracy results in the model without loss of information. In this way, we hope that our model combined with the cited techniques will provide good results.

3 Methods

In this section, we present the methods used in this work. The tasks such as image dataset used, pre-processing, feature extraction, class balancing, feature selection, and classification. The pipeline of the these methods is illustrated in Fig. 1. Then, the next subsections explain each of the stages declared in the pipeline.

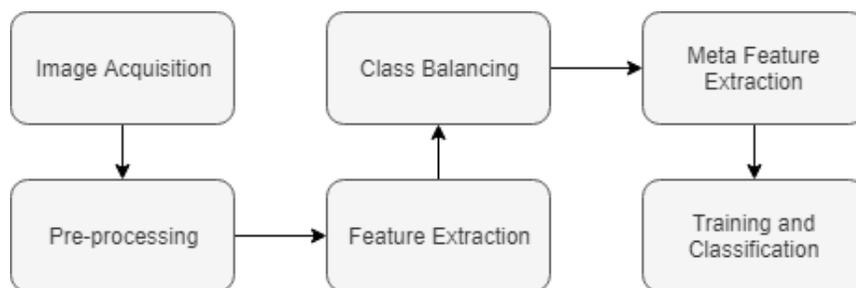


Fig. 1. The pipeline of this work.

3.1 Image Acquisition and Pre-processing

Data used in the preparation of this article were obtained from the Alzheimer’s Disease Neuroimaging Initiative (ADNI) database [10]. The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer’s disease (AD).

We used in this work is all data from ADNI-1 dataset. It contains MRI scans from 3T resonance devices. We collected all exams from this collection through a longitudinal approach within three years. It includes 192 exams of HC patients, 295 of patients with MCI and 98 patients with AD.

For each exam, we perform the conversion of the three-dimensional plane to a two-dimensional plane. According to previously performed works, we selected the slices corresponding to the region where the cerebral ventricle is located, the Fig. 2 illustrates this procedure. That region (above the eyes) can demonstrate the patient’s health about AD. With this, it will be possible to extract features that produce an excellent generalization capacity for the Machine Learning algorithms. It was previously reported in [21] and [20].

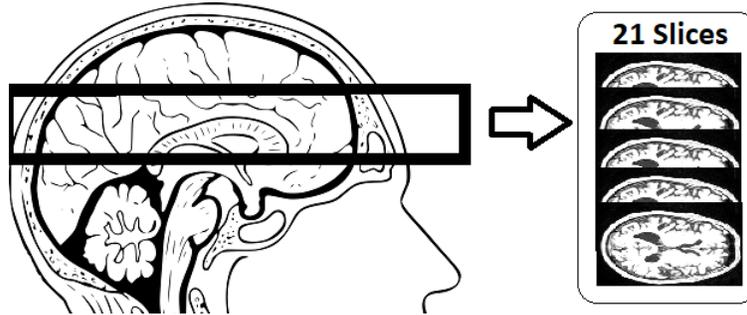


Fig. 2. Process of selecting the slices for use in the experiments of this work.

3.2 Feature Extraction

The textures characterize this variation effect of tonalities present in an image. They can be described as measures that are capable of quantifying features contained in an image. Unlike form descriptors, the segmentation of images is not necessary to obtain the features.

Haralick et al. (1973) proposed a statistical approach for creating descriptors for texture extraction, based on the co-occurrence matrix [8]. These features provide information related to texture, from the assessment of the probability of occurrence of

combinations between the gray levels of the image. The co-occurrence matrix stores the likelihood of each intensity of 256 possible tonalities.

Values obtained with texture extraction as energy, entropy, correlation, inertia, and homogeneity are obtained through calculations in the co-occurrence matrix. With texture extraction, it's not necessary to use image segmentation algorithms [19]. With these features, we can obtain quantifiable properties of specific shapes present in an image. To use the machine learning algorithms, we created a new database containing the features of Haralick. We performed a feature extraction process by selecting the slices in the region above the eyes, as explained in the previous subsection.

In each exam, we selected 21 slices (image group), which correspond to the region of interest. We perform this step in each examination is described as follows: (I) in each of these 21 slices we extracted the Haralick textures; (II) we unified all data into a single vector X . Also, the output Y , corresponding to patient health (AD, MCI or HC), was concatenated into vector X , creating a vector Z . We performed that process in all the examinations contained in the ADNI database. We performed this process to create a new database. This dataset is composed of instances of the vector Z of all the exams of the dataset, according to Fig. 3

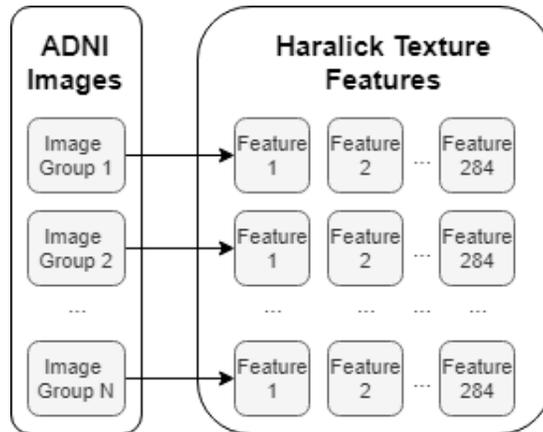


Fig. 3. Extraction process of textures using the Haralick method.

3.3 Class Balancing

The database generated by previous section is unbalanced relative to the classes. As stated earlier, this can cause overfitting. To solve this problem, we apply a class balancing technique. It consists in increasing the number of instances of the minority classes, approaching the majority.

We used the Synthetic Minority Over-sampling Technique (SMOTE) algorithm [3]. This method consists of the creation of artificial instances through an interpola-

tion performed on the data corresponding to the minority classes. The algorithm is based on primordial concepts of the K-Nearest Neighbor algorithm (K-NN). It uses K neighbor instances of the minority class to perform the interpolation, to then create a new artificial instance. The number of neighbors used in this work to carry out class balancing was $K = 5$. The Table 1 shows the results of the number of instances per class obtained after application of SMOTE.

Table 1. A comparison between the number of instances before and after applying the SMOTE class balancer.

Class	Number of instances before SMOTE	Number of instances after SMOTE
AD	98	288
MCI	295	295
HC	192	294

3.4 Meta Feature Selection

The Feature selection techniques aim to reduce the dimensionality of the database to increase the separability between classes. One of the most commonly used techniques in the literature is evolutionary selection and methods based on swarm intelligence. Some of the algorithms most used for this purpose are the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO)[6].

The GA is based on Darwin's selection. GA uses a population with multiple individuals or chromosomes to obtain the final solution to the problem. Individuals are made up of genes, and these are formed by binary values (0 and 1). The choice of the best individual is made using a fitness function. In this way, fit individuals are chosen to produce a new solution through the crossover process. The mutation process is still used so that individuals are not trapped in a local optimum [1].

The PSO is inspired by the collective behavior of the flock of birds. The population consists of particles representing a point in multidimensional space. And the particles move in space through a specific velocity to find the ideal solution to the problem. The choice of the best particles is made through the fitness function [25].

In this work, we use the two algorithms to perform tests in the selection of features. In the database generated through the previous steps, the application of each algorithm is executed, with a variety of parameters. The variation of parameters in search of the best selection result for a later classification was chosen as an approach. The parameters used were defined according to previously published works in the field of medical image analysis [17]. The parameters used in the experiments using GA and PSO are described in the Tables 2 and 3, respectively.

Table 2. Parameters used for GA experiments to select features.

Crossover probability	Population Size	Mutation probability	Number of Generations
0.4 to 0.6	40 to 60	0.1 to 0.3	40 to 60

Table 3. Parameters used for PSO experiments to select features.

Individual Weight	Population Size	Iterations	Inertia Weight
0.34	40 to 60	100 to 150	0.33

3.5 Training and Classification

We use Random Forest algorithm to perform the experiments. The algorithm creates a random forest [16], which corresponds to a combination of decision trees to obtain a maximization of the accuracy of the classification model. Random Forest separates the data into random subsets according to their characteristics. It allows a high generalization ability of the algorithm in the models created from the data generation [16]. The primary parameter to be used in a machine learning process using this algorithm is the number of trees to be used in the experiment. In this work, we changed the parameter of trees following the experiments.

4 Experiments and Results

The first subsection present the environment configuration settings of the experiments. Then the next subsections present the results, discussion and comparison with another published works.

4.1 Experimental Settings

We validate our model using classes AD, MCI, and HC, using a multiclass classification method (AD vs. MCI vs. HC). After taking the data generated after the application of SMOTE, we apply the meta-heuristic algorithm to learn the best features using the parameters described in the [meta feature selection] section. With all the experiments performed, we took into account the test that maximized accuracy using Random Forest for both GA and PSO.

As a result, we generated two more databases. The first contains the best features selected by the GA algorithm. The second one includes the best features chosen by the PSO algorithm. With the mentioned databases, other experiments were carried out using Random Forest to evaluate the models. Before that, we used the Cross Validation method with $K = 10$ (folds) for data partitioning. The algorithm Random Forest was executed with three different parameters in these databases, with the variation in the parameter of trees, having as values 30, 65 and 100.

We adopted three metrics to evaluate the proposed model. These metrics depend on some rates, such as true positive (TP) and negative (TN), and false positive (FP) and negative (FN). The metrics we use are accuracy (ACC), sensitivity (SEN) and specificity (SPE). The SEN and SPE metrics have their values for each class of the classification problem, totaling three SEN and three SPE for each evaluated model. In this work, we use the averages of these metrics for evaluation. The eqs. 1, 2 and 3 present the formulas corresponding to the metrics used.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$SEN = \frac{TP}{TP + FN} \quad (2)$$

$$SPE = \frac{TN}{TN + FP} \quad (3)$$

4.2 Results and Discussion

After applying the model and running the experiments, we collected the best results obtained for each database. The configuration that provided the best results by observing the metrics and using GA has the following parameters: crossover probability - 0.6, population Size - 40, mutation probability - 0.1 and number of generations - 60. For the PSO, the ideal configuration found in the experiments was: individual weight - 0.34, inertia weight - 0.33, iterations - 150 and population size - 40. The results are found in Tables 4 and 5. Table 6 shows the results obtained with the base containing all the features.

Table 4. Results obtained by applying feature selection using GA.

Classifier	Number of trees	ACC	SEN	SPE	Number of Features
Random Florest	30	0.7742	0.7065	0.8474	126
Random Florest	65	0.7788	0.7058	0.8510	126
Random Florest	100	0.7890	0.7155	0.8880	126

Table 5. Results obtained by applying feature selection using PSO.

Classifier	Number of trees	ACC	SEN	SPE	Number of Features
Random Florest	30	0.7766	0.7062	0.8828	50
Random Florest	65	0.7708	0.7055	0.7054	50
Random Florest	100	0.7708	0.7055	0.7054	50

We achieved a reduction of 57 % of the number of attributes. In this way, GA showed satisfactory results. According to the Tables 6 and 4, it obtained a similar

Table 6. Results obtained with the application without the feature selection.

Classifier	Number of tree	ACC	SEN	SPE	Number of Features
Random Florest	30	0.7697	0.7163	0.8794	295
Random Florest	65	0.7890	0.7262	0.8897	295
Random Florest	100	0.7856	0.7158	0.8897	295

effect to the results obtained with a complete database. Considering the number of trees equal to 100 an ACC, SEN, and SPE of GA was superior to the other approaches used. It was observed that with the increase in the number of trees, there is also gain in the accuracy of the model using GA. Also, the results presented greater ACC than the attributes selected with the PSO. We observed the SEN and SPE of the GA were even higher than the PSO. All this shows promising results for the use of GA. In this way, there is evidence that the GA presented better results than the PSO.

About the PSO, although it obtained lower results in the analyzed metrics. It was possible to remove the largest number of features (83% of features), it's showed in the Table 5. The results are very close to those obtained with the GA and complete database. It is worth mentioning that the PSO presented better results with Random Forest when the number of trees was smaller, due to the low amount of features after the selection of features.

In this context, the results obtained show positive results using the selection of features with PSO and GA. There was a significant reduction of features without a substantial decrease of ACC concerning the classification with all features. The reduction of features in this proposed system reduces the complexity of the model and greater the results without a considerable loss in the provision of generalization capacity.

We show the results in Fig. 4. All the results don't present outliers. We observed that the variability using all data is smaller than the other two models. We also noted that the GA results provide a more significant part of the values above the median. Finally, the PSO presented a higher variability than the others.

All results presented in all metrics were higher than 0.70. It demonstrates that there was a balance of hits across all classes, reducing the risk of overfitting. A diagnostic support system needs to have a high capacity for generalization and correctness in both categories. Given this context, we can affirm that this system can solve and has promising results for the diagnosis problem of AD in a multiclass approach.

4.3 Comparison With Previous Works

State of the art for AD classification using a multiclass approach, on average, contains an ACC level above 0.7. For comparison, we selected works with similar experimental characteristics. Table 7 presents information about the comparative work, such as references, approaches and results obtained.

According to the Table 7, we can see that most of the works have a neural network approach. While [2] uses a slice extraction approach using anatomical physical information to extract features, [5] and [9] present fully ANN-based approaches. A

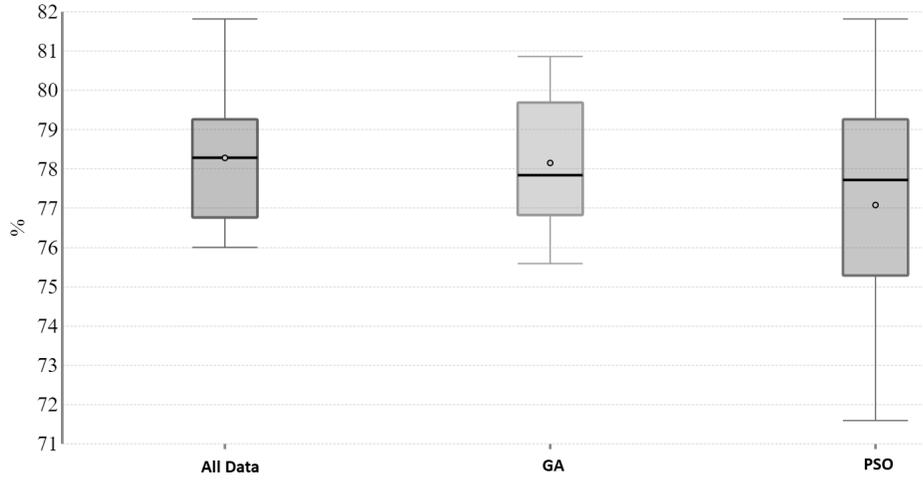


Fig. 4. Averages of the best accuracy obtained with experiments.

Table 7. Comparison with papers published in the literature with a proposal similar to ours.

Ref.	Dataset	Feature Extraction (Selection)	Learning Algorithm	ACC
Tong et al. (2015) [23]	ADNI	Image Pixels	NGF	0.6020
Cardenas-Peña et al. (2016) [2]	ADNI	Cortial Area and Volume	ANN	0.7090
Dolph et al. (2017) [5]	ADNI	SAE	Elastic Net	0.7330
Islam and Tabrez (2017) [9]	OASIS	CNN	Softmax CNN	0.7375
Our Work	ADNI	Haralick (GA)	Random Forest	0.7890

trend of recent years is the use of deep learning, which was the focus of these last two works, Dolph used SAE and Islam used CNN. Both papers have obtained acceptable ACC results for the literature. These works have essential contributions to the proposed theme. It's because they have great potential for improvement due to current advances in Deep Learning in literature.

Our work obtained a high success rate in both classes, which may have contributed to a higher ACC about the others. There is evidence to indicate that the use

of the SMOTE algorithm contributed to this achievement. Class balancing produces a fairer ranking for all classes involved in a Machine Learning project. This is a contribution of our work to the diagnostic area of AD using Machine Learning. Another contribution is to prove that texture extractors, specifically Haralick, can provide good features for AD multiclass classification. Finally, another contribution is the reduction of features used in the project using metaheuristics. It can reduce training time and data testing at work, help in lowering overfitting and high data generalization capability. It happens without much loss of information.

It's important to emphasize that the results were obtained through our experimental settings. We used all data from the ADNI database, for example. Some works of state of the art specify that they follow a traditional model of utilizing up to two years of that dataset [5]. Our model is based on the use of every dataset, and so all the exams that it contains. It's possible that the obtained results will not be found with the reduced ADNI dataset. It's also possible that this phenomenon occurs in other databases such as OASIS. The previously analyzed works can offer better accuracy in different experimental conditions.

5 Conclusion

In this work, we propose a new model for classifying AD. That model consists of extracting features using Haralick textures and attribute selection using meta-heuristic algorithms. The database used was all data from the ADNI database. We used images in the region of the cerebral ventricle in the experiments. We performed the tests using GA and PSO algorithms for attribute selection and Random Forest for data classification. We obtained good results, with ACC of 0.7890 using GA, and 0.7766 using PSO.

In this way, through this work, we can verify the efficiency that the features generated by the textures of Haralick provided to this project. Also, we could also attest that the GA and PSO algorithms can significantly reduce the number of features to be used. It was possible without a significant loss of information, according to the results analysis. Finally, we can say that this work contributes to the literature in the context of multiclass AD classification, providing results with a good generalization capacity of the model.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and FACEPE, Brazilian agencies. Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company;

CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

References

1. Beasley, D., Bull, D.R., Martin, R.R.: An overview of genetic algorithms: Part 1, fundamentals. *University computing* **15**(2), 56–69 (1993)
2. Cárdenas-Peña, D., Collazos-Huertas, D., Castellanos-Dominguez, G.: Centered kernel alignment enhancing neural network pretraining for mri-based dementia diagnosis. *Computational and Mathematical Methods in Medicine* **2016** (2016)
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research* **16**, 321–357 (2002)
4. Dimitriadis, S., Liparas, D., Tsolaki, M.N., Initiative, A.D.N., et al.: Random forest feature selection, fusion and ensemble strategy: Combining multiple morphological mri measures to discriminate among healthy elderly, mci, cmci and alzheimer's disease patients: From the alzheimer's disease neuroimaging initiative (adni) database. *Journal of neuroscience methods* **302**, 14–23 (2018)
5. Dolph, C.V., Alam, M., Shboul, Z., Samad, M.D., Iftekharuddin, K.M.: Deep learning of texture and structural features for multiclass alzheimer's disease classification. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. pp. 2259–2266. IEEE (2017)
6. Ghamisi, P., Benediktsson, J.A.: Feature selection based on hybridization of genetic algorithm and particle swarm optimization. *IEEE Geoscience and Remote Sensing Letters* **12**(2), 309–313 (2015)
7. Han, Y., Zhao, X.M.: A hybrid sequential feature selection approach for the diagnosis of alzheimer's disease. In: *Neural Networks (IJCNN), 2016 International Joint Conference on*. pp. 1216–1220. IEEE (2016)
8. Haralick, R.M., Shanmugam, K., et al.: Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics* (6), 610–621 (1973)
9. Islam, J., Zhang, Y.: A novel deep learning based multi-class classification method for alzheimer's disease detection using brain mri data. In: *International Conference on Brain Informatics*. pp. 213–222. Springer (2017)
10. Jack Jr, C.R., Bernstein, M.A., Fox, N.C., Thompson, P., Alexander, G., Harvey, D., Borowski, B., Britson, P.J., L. Whitwell, J., Ward, C., et al.: The alzheimer's disease neuroimaging initiative (adni): Mri methods. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine* **27**(4), 685–691 (2008)

11. Kandel, E.R., Schwartz, J.H., Jessell, T.M., of Biochemistry, D., Jessell, M.B.T., Siegelbaum, S., Hudspeth, A.: Principles of neural science, vol. 4. McGraw-hill New York (2000)
12. Liu, J., Wang, J., Hu, B., Wu, F.X., Pan, Y.: Alzheimer's disease classification based on individual hierarchical networks constructed with 3-d texture features. *IEEE transactions on nanobioscience* **16**(6), 428–437 (2017)
13. Liu, M., Zhang, J., Adeli, E., Shen, D.: Landmark-based deep multi-instance learning for brain disease diagnosis. *Medical image analysis* **43**, 157–168 (2018)
14. Munteanu, C.R., Fernandez-Lozano, C., Abad, V.M., Fernández, S.P., Álvarez-Linera, J., Hernández-Tamames, J.A., Pazos, A.: Classification of mild cognitive impairment and alzheimer's disease with machine-learning techniques using 1h magnetic resonance spectroscopy data. *Expert Systems with Applications* **42**(15-16), 6205–6214 (2015)
15. Oppedal, K., Eftestøl, T., Engan, K., Beyer, M.K., Aarsland, D.: Classifying dementia using local binary patterns from different regions in magnetic resonance images. *Journal of Biomedical Imaging* **2015**, 5 (2015)
16. Oshiro, T.M., Perez, P.S., Baranauskas, J.A.: How many trees in a random forest? In: International Workshop on Machine Learning and Data Mining in Pattern Recognition. pp. 154–168. Springer (2012)
17. Rodrigues, A.L., Bezerra, R.S., de Santana, M.A., dos Santos, W.P., Azevedo, W.W., de Lima, R.C.: Seleção de atributos para apoio ao diagnóstico do câncer de mama usando imagens termográficas, algoritmos genéticos e otimização por enxame de partículas. In: II Simpósio de Inovação em Engenharia Biomédica. pp. 15–20 (2018)
18. Shams-Baboli, A., Ezoji, M.: A zernike moment based method for classification of alzheimer's disease from structural mri. In: 2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA). pp. 38–43 (April 2017)
19. Silva, I.R.R., Fagundes, R.A.A., de Farias, T.S.M.C.: Techniques for automatic liver segmentation in medical images of abdomen. *IEEE Latin America Transactions* **16**(6), 1801–1808 (June 2018). <https://doi.org/10.1109/TLA.2018.8444402>
20. Silva, I.R.R., Silva, G.S.L., Souza, R.G., Santos, W.P., Fagundes, R.A.A.: Model based on deep feature extraction for diagnosis of alzheimer's disease. In: 2019 International Joint Conference on Neural Networks (IJCNN) (July 2019)
21. Silva, I.R.R., de Souza, R.G., Silva, G.S., de Oliveira, C.S., Cavalcanti, L.H., Bezerra, R.S., Roberta, A.d.A., dos Santos, W.P.: Utilização de redes convolucionais para classificação e diagnóstico da doença de alzheimer. In: II Simpósio de Inovação em Engenharia Biomédica. pp. 73–76 (2018)
22. Sun, Z., Qiao, Y., Lelieveldt, B.P., Staring, M., Initiative, A.D.N., et al.: Integrating spatial-anatomical regularization and structure sparsity into svm: Improving interpretation of alzheimer's disease classification. *NeuroImage* (2018)
23. Tong, T., Gray, K., Gao, Q., Chen, L., Rueckert, D.: Nonlinear graph fusion for multi-modal classification of alzheimer's disease. In: International Workshop on Machine Learning in Medical Imaging. pp. 77–84. Springer (2015)
24. Tong, T., Gray, K., Gao, Q., Chen, L., Rueckert, D., ADNI: Multi-modal classification of alzheimer's disease using nonlinear graph fusion. *Pattern recognition* **63**, 171–181 (2017)
25. Wang, X., Yang, J., Teng, X., Xia, W., Jensen, R.: Feature selection based on rough sets and particle swarm optimization. *Pattern recognition letters* **28**(4), 459–471 (2007)
26. Xiao, Z., Ding, Y., Lan, T., Zhang, C., Luo, C., Qin, Z.: Brain mr image classification for alzheimer's disease diagnosis based on multifeature fusion. *Computational and mathematical methods in medicine* **2017** (2017)

Mining Data Stream to Detect Behavior Change in a Real-Time Strategy Game

Eldane Vieira, Rita Maria Silva Julia, and Elaine Ribeiro de Faria

Computer Department, Federal University of Uberlândia, Uberlândia, Minas Gerais,
Brazil

eldanevieira@gmail.com, {rita,elaine}@ufu.br

Abstract. The ability to progressively capture the opponent's profile in the course of a match is a feature that a player agent has to possess. Besides, an accurate perception of the relevant aspects from the environment is essential for allowing these agents to perform appropriate decision-making. So, it is crucial that the player agent detects, in real-time, any change in the behavior of the opponent. Motivated by this, the authors of this paper propose an approach that endows a player agent in the real-time strategy game StarCraft, with the ability to dynamically detect eventual changes in the playing style of its opponent. This approach is based on the algorithm M-DBScan for Data Stream. In this algorithm, the part responsible for detecting changes corresponds to a Markov Chain based proceeding. The opponent is represented by a set of features that indicates the powers of attack and defense of the opponent. Thereby, the M-DBScan copes with the task of dynamically detecting novelties in the game play of the opponent through a real-time analysis of the data that arrives from the stream, which represents the adversary in a certain moment of the game. The proposed approach was validated through hypothetical games, in which the changing in the behavior of the opponent is completely controlled and known. The aim, in this case, is to verify the accuracy of the changes detected by M-DBScan to show that it is an appropriate technique for a real-time strategy game scenario, which really occurred in the experiments performed.

Keywords: Data mining · Data stream · Novelty detection · Real-time strategy game · Incremental clustering

1 Introduction

Techniques of Artificial Intelligence (AI) have been used in different problems and among these there are those that belongs to a very challenge category of problems which is characterized by a dynamic scenario, in which changes can occur in that which was already known.

This dynamic scenario is usually characterized by the existence of a data stream that brings up some challenge issues, such as the lack of control over the order that data arrives; the unknown amount of data that can be obtained from the stream, and for how long this data can arrive; changes that can happen in the

data; the data that can be obtained at very high frequency [12, 2]. Considering the issues from a dynamic scenario, restrictions such as memory storage and the processing time become important factors for a real-time response.

In order to overcome the challenges from the data stream scenario, the learning process must be incremental, as it must reflect the most recent change that the data from the stream can represent. The data in this scenario can become irrelevant and harmful to the models trained from this data, so a forgetting mechanism to discard old data must be applied.

This work uses the game StarCraft, developed by *Blizzard Entertainment*, using specifically the expansion StarCraft: BroodWar as a case of study. This game belongs to the category *real-time strategy* (RTS), and in this type of game it is common to perform tasks such as extraction of resources from environment, construction of structures using the resources accumulated and the production of new army units.

The frequent choice of games as a case of study in AI research is due to the fact that games can reproduce challenges similar to those seen in the real world, such as unpredictability, existences of events produced by your adversary that try to minimize your success in the game and the need for adaptation due to changes in the environment. In a game there are also challenges related to the dynamic scenario: first, the agent must be capable of identifying changes in the behavior of a player; second, the agent should be able to adapt its strategy according to the change identified from the data that describes its adversary at a given moment. The present work will provide a solution to the first challenge in which some related works ([21],[19],[13]) has achieved great success in this issue.

In order to solve the challenge of detecting changes in the playing style observed in the behavior of the opponent, in this work, the algorithm Micro-Clustering DBScan (M-DBScan) is used, since it has been applied over a *first person shooter* game data, presenting satisfactory results[19].

In this work, the *Brood War Application Programming Interface* (BWAPI) was used, as it provides the means to interact and implement changes in the game StarCraft. It was also used the engine named UAlbertaBot [5, 6]. Thus, the aims of this paper were the development of an approach that analyses the adversary in a way that it will point out any changes that can be considered a novelty.

This paper is organized as follow: in section 2, the theoretical foundations are presented; in section 3, the related works are described; in section 4, a description is given to how the novelty detection is applied in the data from the game StarCraft; in section 5, experiments and results are presented; and in section 6,, the conclusion and future works are presented.

2 Theoretical Foundations

This section presents a short description of some concepts and techniques used in the development of the work presented in this paper.

2.1 Player Modeling

Player modeling is defined as the study of computational models of game players and its study aims to understand how the player interacts with a game, which may include the detection and prediction of its features, and also how to model and express these features [22].

Although player modeling is commonly used in human-computer interaction, it has also been used in the context where player is an agent, and this case, player modeling can be used to improve these agents as described in [20, 13].

The present paper aims at identifying novelties in the playing style of the adversary, creating a model that represents it. This model will be adapt over time, in order that it continually reflects recent information.

2.2 Novelty Detection in data stream

The task of detecting novelties in the behavior of a player can observe the features that describe relevant aspects in playing style. One way to accomplish this task is use a novelty detection technique. Novelty detection can be defined as a process capable of identifying an instance of data that differs from some already known concepts, this has received a great deal of attention in the research area of machine learning and data mining[9].

The interaction between the player, from whom one aims to detect novelties, and the game may provide a flow of data with the characteristics of a data stream approach. Being that, a data stream can be defined as a sequence of data instances that arrives continuously, and there are some features that illustrate how the data stream scenario can be characterized [10]: the data in the stream arrives online; there is no control over the order in which the data arrives; there is no limit in the size of a data stream; once a data instance from a data stream has been processed it is usually discarded, as the system has to deal with limited resources as memory.

2.3 M-DBScan

The algorithm M-DBScan was created to detect behavior changes in a dynamic scenario, such as data stream. This algorithm has an incremental clustering process, based on DenStream [4] which is a clustering technique applied in a data stream. In addition, the clustering process is followed by a mechanism to detect novelties used for identifying behavior change [18].

The clustering applied in the M-DBScan algorithm is divided into online and offline phases. So, for each data point that arrives from data stream, the incremental clustering process is executed, and later the clusters produced are used in a novelty detection module that aided by a sliding window the algorithm is capable to indicate the occurrence of behaviors change.

To understand the dynamics for creating and changing clusters, the essential concepts of core-micro-cluster, p-micro-cluster and o-micro-cluster, proposed originally in the algorithm DenStream, are described in the following:

Core-micro-cluster A core-micro-cluster (c-micro-cluster) can be represented by its weight (w), center (c) and radius (r), so for a group of points p_1, \dots, p_n with time stamps T_1, \dots, T_n . The weight is calculated by equation 1, the center of a c-micro-cluster can be calculated by equation 2, and radius of a c-micro-cluster can be determined by equation 3.

The issue of clustering a data stream is that the weight of a cluster has its value decreased exponentially by a fading function, which receives as its parameter the time t , $f(t) = 2^{-\lambda t}$, where λ is a decay factor and $\lambda > 0$.

$$w = \sum_{j=1}^n f(t - T_j), \quad (1)$$

$w \geq \mu$, μ is a minimum amount of points, and t is the current time.

$$c = \frac{\sum_{j=1}^n f(t - T_j)p_j}{w} \quad (2)$$

$$r = \frac{\sum_{j=1}^n f(t - T_j)dist(p_j, c)}{w} \quad (3)$$

$r \leq \epsilon$, ϵ is a radius limit and $dist(p_j, c)$ is the Euclidean Distance between the point p_j and the center c .

A c-micro-cluster has been defined as a dense micro-cluster, and the set of c-micro-clusters must correspond to the coverage of core objects in the cluster. Clusters and outliers can have their classifications changed as the data stream proceeds, so the algorithm M-DBScan uses two structures: the potential c-micro-cluster (p-micro-cluster) and the outlier-micro-cluster (o-micro-cluster).

P-micro-cluster A p-micro-cluster can be represented by a tuple $\{\overrightarrow{CF^1}, \overrightarrow{CF^2}, w\}$, considering a group of points p_1, \dots, p_n with time stamps T_1, \dots, T_n , the weighted linear sum of points represented by $\overrightarrow{CF^1}$ is calculated by equation 4, the weighted squared sum of points represented by $\overrightarrow{CF^2}$ is calculated by equation 5, and the weight w is determined by equation 1, but for a micro-cluster be classified as a p-micro-cluster, it must satisfy the restriction $w \geq \beta\mu$, being β ($0 < \beta \leq 1$) an outlier threshold that multiplied by the minimum amount of points (μ), defines the reference value used to classify a micro-cluster as p-micro-cluster.

$$\overrightarrow{CF^1} = \sum_{j=1}^n f(t - T_j)p_j \quad (4)$$

$$\overrightarrow{CF^2} = \sum_{j=1}^n f(t - T_j)p_j^2 \quad (5)$$

The center of a p-micro-cluster (cPMC) is determined by equation 6, and its radius (rPMC) is calculated by equation 7, $rPMC \leq \epsilon$ (ϵ is a maximum boundary for the radius of micro-clusters), .

$$cPMC = \frac{\overrightarrow{CF^1}}{w} \quad (6)$$

$$rPMC = \sqrt{\frac{|\overrightarrow{CF^2}|}{w} - \left(\frac{|\overrightarrow{CF^1}|}{w}\right)^2} \quad (7)$$

O-micro-cluster An o-micro-cluster is represented by a tuple $\{\overrightarrow{CF^1}, \overrightarrow{CF^2}, w, t_0\}$. The definitions of $\overrightarrow{CF^1}$, $\overrightarrow{CF^2}$ and w , are the same as those presented in the description of p-micro-cluster in section 2.3. The data t_0 represents the creation time of an o-micro-cluster, and it is used to determine the o-micro-cluster life span. The main difference between an o-micro-cluster and a p-micro-cluster is their weight, because if an amount of points from data stream is assigned to an o-micro-cluster, its weight will change, and thereafter, the o-micro-cluster can become a p-micro-cluster if the restriction applied in the weight of a p-micro-cluster ($w \geq \beta \cdot \mu$) is satisfied.

A p-micro-cluster which does not receive any points in a interval of time can be reclassified as an o-micro-cluster because of a decay factor used over its tuple $\{\overrightarrow{CF^1}, \overrightarrow{CF^2}, w\}$.

Online and offline phases of M-DBScan The online part of the algorithm M-DBScan can be described as a micro-cluster maintenance process, in which a group of p-micro-cluster and o-micro-cluster is maintained in an online way, in order to discover, in the offline part, clusters from data stream points.

The *online part* can be summarized by the following steps, and this procedure is repeated for every point p that arrives from the data stream:

1. Since the point p has just arrived, the algorithm tries to merge it into the closest p-micro-cluster pmc . If p is reachable by the radius of pmc , then p is merged into pmc .
2. Otherwise, the algorithm will try to merge p into the nearest o-micro-cluster omc . If p is reachable from the radius of omc , then p is merged into omc . Then, one checks if the weight of omc has exceeded the limited of an o-micro-cluster, becoming a p-micro-cluster. If this has happened, the omc will be promoted as a new p-micro-cluster and it is removed from the outlier-buffer, where the o-micro-clusters are kept.
3. Else, a new o-micro-cluster new_omc is created to allocate the point p and the new_omc is inserted into the outlier-buffer.

After the online phase, a decay factor is applied to every micro-cluster, except the one that received the last data instance that arrived from the stream.

Considering a previous defined interval of time, micro-clusters can pass through a *rating verification*, in which a p-micro-cluster can be re-classified as an o-micro-cluster if it does not satisfy the criterion of weight for a p-micro-cluster. In this process, some o-micro-cluster can be deleted if it also does not satisfy the criterion of weight necessary for an o-micro-cluster.

The *offline part* of M-DBScan generates clusters based on the result produced by the online part. In this phase, a variant of DBSCAN algorithm is applied. In order to determine the final clusters, the offline part adopts the concepts of density and connectivity, so each density-connected p-micro-clusters produce a cluster or will be part of one [4]:

- Directly density-reachable: A p-micro-cluster PMC_1 is directly density-reachable from a PMC_2 if $weightPMC_2 > \mu$ and $dist(centerPMC_1, centerPMC_2) \leq 2.\epsilon$, where $dist(centerPMC_1, centerPMC_2)$ is the Euclidean Distance between the centers of PMC_1 and PMC_2 .
- Density-reachable: A p-micro-cluster PMC_1 is density-reachable from a p-micro-cluster PMC_n , if there is a chain of p-micro-clusters between PMC_1 and PMC_n , such that PMC_{i+1} is directly density-reachable by PMC_i , where $1 \leq i \leq n$.
- Density-connected: A p-micro-cluster PMC_1 is density-connected from a p-micro-cluster PMC_n , if there is a p-micro-cluster PMC_k such that both PMC_1 and PMC_n are density-reachable from PMC_k .

Once the offline phase has been concluded, the novelty detection module, which uses Markov Chain (MC), starts. In this process each state in the MC represents a group produced in the offline phase, and the transitions between the states follow the dynamics observed in the arrivals of new data and how the assignment of this data to a micro-cluster occurred.

Novelty Detection Module In order to detect novelties the M-DBScan algorithm uses entropy measures to identify changes. The entropy measures used in this work are spatial entropy and temporal entropy.

The spatial entropy estimates the spatial distribution of data into clusters. The number of examples in a cluster i is denoted as P_i , and it is updated every time a new example is inserted into it (equation 8). However, if there are other groups, and these have not received data at the current moment of time, so they will be update by equation 9, where η_s is a weighting parameter.

$$P_i = (1 - \eta_s).P_i + \eta_s \quad (8)$$

$$P_i = (1 - \eta_s).P_i \quad (9)$$

The spatial entropy measure ($H(G)$) is calculated according to equation 10, in which G is the set of clusters and N is the total number of clusters.

$$H(G) = - \sum_{i=0}^N P_i \cdot \log_2(P_i) \quad (10)$$

Since the states in the MC are the representation of groups produced in the offline part of M-DBScan, the probabilities in the MC transitions are up dated every time that a data arrives from the stream. The sum of probabilities linked to transitions with the same origin state has 1 as its maximum value, when a transition is no longer activated, it will suffer a value deterioration.

The probability between the states i and j from a MC is represented by $P_{i,j}$, where i is the state that receives a data sample at the moment $T - 1$ and j represents the state that received the data sample at the moment T . A weighting factor η_t is used to control the intensity of each probability update.

Considering that the state j is the one that has just received data, and its previous state was i , so the update of the probability $P_{i,j}$ is performed by equation 11.

$$P_{i,j} = \frac{(1 - \eta_t) \cdot P_{i,j} + \eta_t}{\sum_{k=0}^N P_{i,k}} \quad (11)$$

The other transitions in the MC will have their probabilities updated according to equation 12.

$$P_{a,b} = \frac{(1 - \eta_t) \cdot P_{a,b}}{\sum_{k=0}^N P_{a,k}} \quad (12)$$

The temporal entropy ($H(M)$) is calculated by equation 13, where M is the set of states in the MC.

$$H(M) = - \sum_{i=0}^N \sum_{j=0}^N P_{i,j} \cdot \log_2(P_{i,j}) \quad (13)$$

In order to identify a novelty, at least one of the entropies must surpass its threshold, which is calculated based on a moving average of a historical entropy value Φ (equation 14) and moving standard deviation Ω (equation 15), where γ and δ are weighting factors that can accept values from the interval $[0,1]$. The entropy at moment t is represented by $H_t(\cdot)$.

$$\Phi_t = (1 - \gamma) \cdot \Phi_{t-1} + \gamma \cdot H_t(\cdot) \quad (14)$$

$$\Omega_t = (1 - \delta) \cdot \Omega_{t-1} + \delta \cdot |H_t(\cdot) - \Phi_t| \quad (15)$$

To calculate the threshold of an entropy (τ) (equation 16), the algorithm M-DBScan assumes a Normal distribution on entropy values, adopting a constant parameter θ , which is the number of standard deviation to be defined according to the problem.

$$\tau = \Phi_t + (\Omega_t \cdot \theta) \quad (16)$$

In order to indicate a novelty in the behavior of an adversary, a sequence of novelties has to occur. So, the algorithm M-DBScan requires that a minimum amount of novelty must be identified in an interval of time represented by a sliding window with size k . Every novelty occurrence is registered in the sliding window, where it is necessary to verify if the amount of registered novelties has reached the minimum amount necessary to declare a behavior change. A novelty that occurred in a certain moment can disappear from the window if it has slid k times, considering as the initiation, the moment that this novelty was registered. However, if a behavior change was detected, from this point on every novelty will be ignored for a period of time equal to k . This process is used to discard novelties linked to the same behavior change that have already been identified [19].

3 Related Works

Several related works that apply machine learning techniques in RTS games try to optimize tasks as the order that new units for an army must be constructed and when resources from the environment must be collected. In [11], for example, was described a work that deals with the task of deciding the structures that must be constructed in the game StarCraft, where the algorithm *Continual Online Evolutionary Planing* (COEP) is used. It is important to emphasize that this related work does not analyze the adversary playing style searching for novelty.

There are related works that try to improve an RTS automatic player, but dealing with battles issues. The related work presented in [14], for example, describes the development of a decision module for the game StarCraft, which is based on a neural network with reinforcement learning. Another related work that improves the automatic player for a battle is described in [15], this uses a prediction method that tries to indicate the most likely combination of units produced by the enemy in an RTS game as Warcraft and StarCraft, both developed by *Blizzard Entertainment*. The *Answer set programming* was used to determine the answers of this prediction. The related works [14, 15] differ from the present paper since they do not point out novelties in the playing style and it does not deal with issues of data stream issues. In [17] is described other related work for the RTS game StarCraft II, which uses deep neural network with reinforcement learning trained in a supervised way, by using a dataset of games and playing against itself, this related work does not model the adversary and it does not try to detect novelties in the behavior of the adversary as the present paper proposes. In [16] is presented a method to discover tactics and strategies in the game StarCraft by analyzing the composition of the armies in a dataset of different games. In [20] is described a work that tries to learn strategies in the game StarCraft using data set of games. The main objective of this related work was to create a model that represents the enemy in a supervised way using data

from different players. Different to the present paper the related works [16, 20] do not try to point out any changes in the playing style of the opponent.

Another set of related works are linked to the modeling players of digital games. In [18] a work that models a human player to detect change in the behavior of this player using the algorithm M-DBScan is described. Despite the use of the algorithm M-DBScan, the study presented in this paper aims at applying M-DBScan to indicate the occurrence of changes in the playing style of the adversary, by considering strategic relevant features from StarCraft, using no information related to a human player. In [21], a Bayesian Network is used to create a player model trained over data from a predator/prey game. The aim of this related work is to predict the set of parameters that will produce more interesting games, it does not try to detect changes in the playing style of the adversary. Another related work based on player modeling is described in [13], in which it models the opponent of an RTS game named Spring, it classifies the strategy associated to the behavior of the opponent. In this related work subcategories are also identified in the play style, which occur based on repeated matrix-games, where one notes that this related work does not deal with challenges from a data stream scenario.

There are also related works that deal with novelty detection. The aforementioned related work [18] points out novelties associated to player behavior. Another study, presented in [8], describes a technique to detect novelties in data stream named the *Multi-class Learning Algorithm for Data Stream* (MINAS). Among the contributions put forward by MINAS are the capability of the algorithm to deal with multi-class problems and the use of unknown data for learning new concepts or for adapting those already learned. In [1], the algorithm *Self-Organizing Novelty Detection* (SONDE) is presented, which is a neural network that classifies patterns in a unsupervised way, the activation of a neuron occurs according to the input patterns, and a new neuron can be created if the existing ones are not capable of identifying the new pattern. Another study presented in [3] describes the algorithm Adaptive Windowing (ADWIN) used to detect distribution changes and concept drifts. In [3], the version ADWIN2 which is superior in performance to ADWIN is also presented. The related works [18, 8, 1, 3] have not been used to detect changes in the behavior of the opponent in an RTS game using relevant game features as the present paper does.

4 Novelty Detection in StarCraft

StarCraft is an RTS game released by *Blizzard Entertainment* in 1998 [7], the expansion set StarCraft: BroodWar was released later in the same year, and it has gained a great popularity in AI research, and it is used in this work.

In the game StarCraft players can control one of three races: *Terran*, *Protoss* and *Zerg*. Each race has its own features as weaknesses and strengths. There are differences among the units that compound the army, for example, there are units which its main function is of behaving as a worker, building structures and collecting resources from the environment, others are more appropriate for

combat. The quality of a player army can be evaluated by analyzing each member of its army, since it reflects the power of attack and defense of this player. This study will use information concerning the composition of the opponent's army to point out any significant change in its playing style. Then, this work will deal with data that describes the enemy by use of features, and over this data the M-DBScan algorithm is applied.

Some challenges are noted in RTS games as StarCraft, for example, it is not possible for a player to see what is in areas of the map that have not been explored. There are also challenges related to the data stream scenario, once the data can be obtained from a game in a very high frequency, reflecting the interaction between the game and the player. The uncertainty concerning the time that a game will take makes it impossible to guess how much data can be obtained from the data stream and if this data will suffer any changes. So, all these facts confirm that the use of a real-time technique can provide valuable information in an RTS game.

Since this study aims at indicating changes in the playing style of the opponent, using relevant information concerning the adversary in the game StarCraft, each feature selected was defined using a perspective that provides information on the adversary. The M-DBScan algorithm has been used over a dataset composed of the following features, which represents a perception of the enemy at a moment t of the game:

- Feature 1 (Dead enemy unit): This feature provides the number of deaths in the enemy army until the moment t .
- Feature 2 (Attack power): This feature represents the damage that the enemy army can cause. It is calculated by considering the damage of each unit that exists in the enemy army in a moment t .
- Feature 3 (Defense power): This feature represents the defense power of the enemy army. This feature is calculated considering the armor level of each unit that exists in the enemy army in a moment t .
- Feature 4 (Damage in construction): This feature provides a score that represents the damage that the enemy caused to structures.

Every step of the M-DBScan algorithm is illustrated in a flowchart presented in Figure 1. The algorithm starts by trying to capture a data sample (set of features) from the stream (arrow #1), if this is performed successfully (arrow #2), the data is presented as an input to the online part of the algorithm M-DBScan (arrow #3), as seen in section 2.3. Posteriorly, a decay factor is applied to every p-micro-cluster and o-micro-cluster that has not received the most recent data from the stream (arrow #4). Then the algorithm tries to run a rating verification (arrow #5), which means to perform within a certain interval of time, if it is the moment for running this test (arrow #6), some p-micro-clusters, that do not satisfy the requirements of a p-micro-cluster (see section 2.3), must become an o-micro-cluster. Then some o-micro-cluster can be deleted if it does not meet the requirements of an o-micro-cluster (see section 2.3) (arrow #7). After this verifications, the offline part of M-DBScan is run (arrow #8) (section

2.3), but if it is not the moment to run the rating verification, the offline part of M-DBScan will be run directly after the use of decay factor (arrow #9). In sequence, some calculations are performed: the entropy value is calculated (arrow #10), and this can be the temporal entropy (equation 13) or spatial entropy (equation 10). Then the threshold of the entropy used is calculated (arrow #11) (equation 16). After these calculations, the value of the entropy used is compared to its threshold (arrow #12). If the entropy surpasss its threshold, then a novelty is detected and registered in the sliding window (arrow #13). In sequence (arrow #14), a module to detect behavior change is activated, and in this module is done the verification of the minimum amount of novelties, and it controls the sliding window. Next, verification is made as to if there is a new data sample in the stream (arrow #15). However, if a novelty is not detected (arrow #16), then the existence of a new data sample in the stream is verified directly. If there is no more data to be processed, this algorithm is finished (arrow #17).

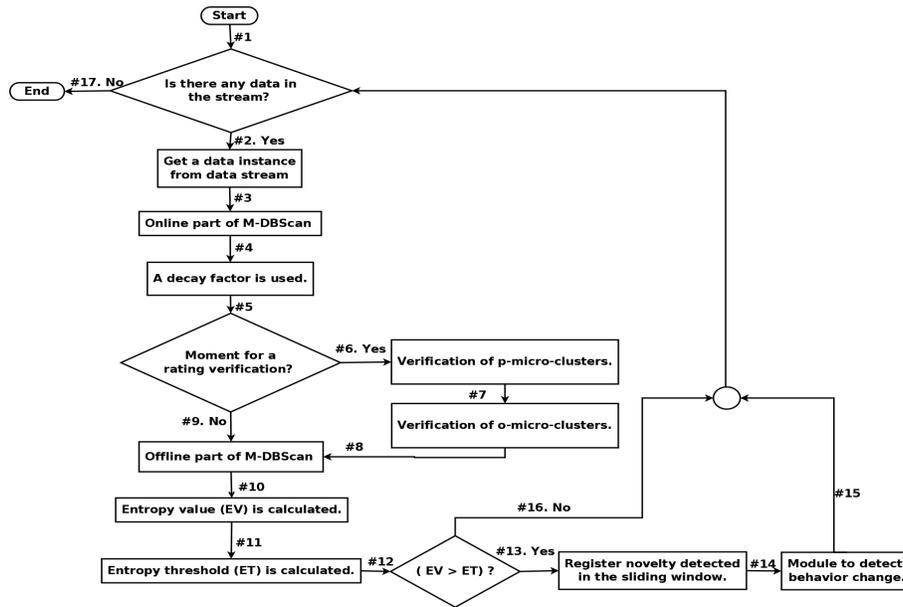


Fig. 1. M-DBScan flowchart.

5 Experiments and Results

In order to show that the M-DBScan algorithm is an effective technique to identify novelties in the playing style of the adversary, a set of test scenarios was created to represent different situation in the game StarCraft.

The results of the experiments are presented using the metrics false positives (FP), false negatives (FN) and true positives (TP). The delay in detecting a novelty will also be pointed out, where a delayed detection is that one which occurs after the arrival of more than 10% up to 30% of data instance after the moment when the novelty occurred. The metrics are presented for each entropy measure (temporal and spatial).

- The parameters used in the clustering process of the algorithm M-DBScan are: μ is 9; β is 0.1; ϵ is 20; λ is 0.03.
- The parameters used in the experiments for temporal entropy are: minimum amount of novelties is 2; size of sliding window is 20; η_t is 0.05; γ is 0.05; δ is 0.03; θ is 3.
- The parameters used in the experiments for spatial entropy are: minimum amount of novelties is 2; size of sliding window is 20; η_s is 0.05; γ is 0.05; δ is 0.002; θ is 3.

The data used in the experiments was obtained from controlled games in StarCraft that permitted to create specific test scenarios, being that each instance in the dataset adds up to 15 seconds of game data. In order to create some complex tests scenarios and know the correct moment that a change occurs, game matches were played and in each of these a specific type of behavior was followed along the whole match. Then, the datasets could be created putting in a sequence data from different matches, so the point of junction of these games is the moment that occurs a behavior change. Every behavior is represented by a sequence of 1000 data instance arriving from the stream before the occurrence of a change. The test scenarios are the following:

- Scenario 1: This is a simple test with one change of behavior, in which the opponent stops creating units, and it focuses on gathering resources and building new structures.
As noted on Table 1 both entropies were capable of identifying the expected novelty, and in no situation did a detection delay occur. The precision to identifying novelties in this test, for each entropy, is equal to 1 as well as the recall measure, so the F1 score is 1 indicating a perfect result.
- Scenario 2: In this test there occur three changes inside the category of aggressive behavior. These changes consist of increasing the attacking power and consequently the number of kills increases.
The results obtained with test scenario 2, presented on Table 1, shows that M-DBScan when using the spatial and temporal entropies was capable of identify the 3 behavior changes expected in this test, and none identified any change that occurred with delay. The precision of identifying novelties in this test, for each entropy, is equal to 1 as well as the recall measure, so the F1 score is 1 which is a perfect result.
- Scenario 3: The opponent creates mostly units for battles, and then there is an upgrade that increases its defense skill (first change). After this, the units will have a new upgrade (second change) that will increase its attack skill that will be reflected as damage caused by the units.

As seen on Table 1 both entropies were capable of identifying the expected novelty, but using the temporal entropy there was false positive identification, and in no situation did there occur a detection delay. The precision of identifying novelties in this test, using spatial entropy, is equal to 1, as was the recall measure, so the F1 score is 1, which is a perfect result. Although, the precision using temporal entropy is about 0.66, the recall measure is 1, and the F1 score is 0.8, which is still a good result indicating an acceptable accuracy.

- Scenario 4: In this scenario the opponent will alternate two behaviors B1 and B2. B1 will focus on attack units increasing the number of kills, and the second behavior B2 consists of attacking structures, thus increasing the damage to structures. Once B2 is concluded, B1 will be repeated followed by B2. This process is repeated 3 times, a total of 6 behavior changes (Idle behavior \rightarrow B1 \rightarrow B2 \rightarrow B1 \rightarrow B2 \rightarrow B1 \rightarrow B2).

In test scenario 4, the results presented on Table 1 show that no entropy was capable of identifying all the behavior changes in this test. The temporal entropy identified 3 changes with no delay and there was one false positive detection. By noting the values of the temporal entropy measure, the changes identified were the first three. The spatial entropy identified the first three changes and the last change from behaviors B2 to B1. Noteworthy was that once the MC had the states already defined, the data that arrived was not sufficient to overpass the entropies threshold in order to identify the other changes.

The precision of identifying novelties in test 4, using spatial entropy, is equal to 1, the recall measure is about 0.66, and the F1 score is 0.8, which is very good result, showing that the spatial entropy is still reliable for the detection of behavior changes even in a more complex scenario. The precision using temporal entropy is 0.75, the recall measure is 0.5, and the F1 score is about 0.6, which indicates an accuracy that maintains the use of the temporal entropy as a good alternative for detecting novelties.

Table 1. Results of test scenarios

	Temporal Entropy			Spatial Entropy		
	FP	FN	TP	FP	FN	TP
Test scenario 1	0	0	1	0	0	1
Test scenario 2	0	0	3	0	0	3
Test scenario 3	1	0	2	0	0	2
Test scenario 4	1	3	3	0	2	4

By analyzing the experiments one notes that when the spatial entropy has an increase in its value, it indicates a heterogeneity growth in the data that arrives from the data stream. If the spatial entropy value starts to reduce, it indicates that the data samples have greater similarity, and if this scenario occurs for

a long period of time, it will have effects over the groups formed. Due to the similarity of the data, a few groups or even one group will receive the data samples, and it may cause those groups that do not receive a data sample for a long period of time to vanish.

Low temporal entropy values indicates that the data samples have not been distributed between different groups, indicating that for a given interval of time the data samples are concentrated in a small quantity of groups. However, if the temporal entropy has an increase in its value, this indicates that different transitions in the MC have been activated, so the data has been distributed between different groups, and this occurs due to the low level of similarity in the data.

6 Conclusion and Future Works

This paper introduces an approach that aims at detecting changes in the playing style of an opponent in an RTS game, where the game StarCraft was used as the study environment. So, the aims of this work are to show that the M-DBScan algorithm is appropriate for detecting the changes in adversary behavior. This information can be useful when adapting the strategy used by the player that faces this adversary.

The experiments run in this work reproduced different scenarios illustrating a change in the behavior of the adversary, and the results showed that the temporal and spatial entropies, used by the M-DBScan, achieved good results in the experiments. Noteworthy was that the spatial entropy presented better results in the most complex test scenarios, but even though the results presented by the temporal entropy are good enough to maintain its usage.

Future works will consist of creating a player agent for StarCraft that uses the behavior change detection module presented in this work, in order to make it capable of adapting its strategy in function of the changes detected. It is also important to identify the semantic meaning in the change so the adaptation in strategies can be coherent.

References

1. Albertini, M.K., de Mello, R.F.: A self-organizing neural network for detecting novelties. In: Proceedings of the 2007 ACM symposium on Applied computing. pp. 462–466. ACM (2007)
2. Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. In: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 1–16. ACM (2002)
3. Bifet, A., Gavaldà, R.: Learning from time-changing data with adaptive windowing. In: Proceedings of the 2007 SIAM international conference on data mining. pp. 443–448. SIAM (2007)
4. Cao, F., Estert, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise. In: Proceedings of the 2006 SIAM international conference on data mining. pp. 328–339. SIAM (2006)

5. Churchill, D., Buro, M.: Build order optimization in starcraft. In: *AIIDE*. pp. 14–19 (2011)
6. Churchill, D., Saffidine, A., Buro, M.: Fast heuristic search for rts game combat scenarios. In: *AIIDE*. pp. 112–117 (2012)
7. Entertainment, B.: *Starcraft: Brood war*. PC. Level/area: Episode IV, mission **2** (1998)
8. Faria, E.R., Gama, J., Carvalho, A.C.: Novelty detection algorithm for data streams multi-class problems. In: *Proceedings of the 28th annual ACM symposium on applied computing*. pp. 795–800. ACM (2013)
9. Faria, E.R., Gonçalves, I.J., de Carvalho, A.C., Gama, J.: Novelty detection in data streams. *Artificial Intelligence Review* **45**(2), 235–269 (2016)
10. Gama, J., Gaber, M.M.: *Learning from data streams: processing techniques in sensor networks*. Springer (2007)
11. Justesen, N., Risi, S.: Continual online evolutionary planning for in-game build order adaptation in starcraft. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. pp. 187–194. ACM (2017)
12. Krawczyk, B., Minku, L.L., Gama, J., Stefanowski, J., Woźniak, M.: Ensemble learning for data stream analysis: A survey. *Information Fusion* **37**, 132–156 (2017)
13. Schadd, F., Bakkes, S., Spronck, P.: Opponent modeling in real-time strategy games. In: *GAMEON*. pp. 61–70 (2007)
14. Shao, K., Zhu, Y., Zhao, D.: Cooperative reinforcement learning for multiple units combat in starcraft. In: *Computational Intelligence (SSCI), 2017 IEEE Symposium Series on*. pp. 1–6. IEEE (2017)
15. Stanescu, M., Čertický, M.: Predicting opponent’s production in real-time strategy games with answer set programming. *IEEE Transactions on Computational Intelligence and AI in Games* **8**(1), 89–94 (2016)
16. Synnaeve, G., Bessiere, P., et al.: A dataset for starcraft ai & an example of armies clustering. In: *AIIDE Workshop on AI in Adversarial Real-time games*. vol. 2012 (2012)
17. The AlphaStar team: AlphaStar: Mastering the real-time strategy game starcraft ii (2019), <https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/>
18. Vallim, R.M.M.: *Mineração de fluxos contínuos de dados para jogos de computador*. Ph.D. thesis, Universidade de São Paulo (2013)
19. Vallim, R.M., Andrade Filho, J.A., De Mello, R.F., De Carvalho, A.C.: Online behavior change detection in computer games. *Expert Systems with Applications* **40**(16), 6258–6265 (2013)
20. Weber, B.G., Mateas, M.: A data mining approach to strategy prediction. In: *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*. pp. 140–147. IEEE (2009)
21. Yannakakis, G.N., Maragoudakis, M.: Player modeling impact on player’s entertainment in computer games. In: *International Conference on User Modeling*. pp. 74–78. Springer (2005)
22. Yannakakis, G.N., Spronck, P., Loiacono, D., André, E.: Player modeling. In: *Dagstuhl Follow-Ups*. vol. 6. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2013)

Packet2Vec: Utilizing Word2Vec for Feature Extraction in Packet Data

Eric L. Goodman, Chase Zimmerman, and Corey Hudson

Sandia National Laboratories, Albuquerque, NM, USA

Abstract. One of deep learning’s attractive benefits is the ability to automatically extract relevant features for a target problem from largely raw data, instead of utilizing human engineered and error prone hand-crafted features. While deep learning has shown success in fields such as image classification and natural language processing, its application for feature extraction on raw network packet data for intrusion detection is largely unexplored. In this paper we modify a Word2Vec approach, used for text processing, and apply it to packet data for automatic feature extraction. We call this approach *Packet2Vec*. For the classification task of benign versus malicious traffic on a 2009 DARPA network data set, we obtain an area under the curve (AUC) of the receiver operating characteristic (ROC) between 0.988-0.996 and an AUC of the Precision/Recall curve between 0.604-0.667.

1 Introduction

An appealing aspect of many deep learning approaches is the ability to automatically extract features from largely unprocessed data. In Krizhevsky et al. [13], one of the seminal works that started the popularization of convolutional neural networks applied to images, they show that the learned early convolutional kernels displayed a range of image filters, similar to hand-crafted features from more traditional vision processing approaches such as SIFT [15] and SURF [4].

For text processing, Word2Vec approaches [17, 18] create a vectorized representation of words, called embeddings, where similar words (e.g. King and Queen) are close distance-wise in the embedded space. Vector operations also make intuitive sense, such as $King - Man + Woman = Queen$, meaning that the vector representation of *King* minus the vector for *Man* plus the vector for *Woman* creates a vector where the closest word embedding is the one for *Queen*. This feat is achieved on a large corpus of raw text with little to no-preprocessing. The deep learning approach is able to create these word embeddings just based on the text itself without human-engineered feature extraction.

Cyber data and intrusion detection is an area ripe for exploration of how deep learning can automatically extract features from raw packet data. However, most of the current work applying deep learning to intrusion detection relies upon the features already being extracted from packet data [12, 27]. Many researchers choose to use data sets such as NSL-KDD [24, 26] or the original 1999 KDD data set, both of which have 41 features to represent the network packet data.

Instead of creating hand-crafted features for each packet, the approach we take is to pass the raw packet data through a Word2Vec approach to create a vectorized representation for each packet, and then perform classification of the packet based on that representation.

Specifically, our approach has the following steps:

- **N-grams:** Word2Vec requires a sequence of tokens. Packet data has no clear analog. To address this, we take each packet and transform it into a sequence of n-grams. This forms our sequence of words, similar to the presentation of text. We purposefully throw out IP and port information, as we want the representation of the packet to be based on content, not who sent it.
- **Embeddings:** Once we have a sequence of n-grams, applying Word2Vec is straightforward, and we create a vectorized representation for each frequent n-gram (vocab size is a hyperparameter).
- **Feature Vectors:** To perform classification on each packet, we need a fixed-size vector representation for each packet. We take the simple approach of averaging the word embeddings for all of the n-grams in a packet, i.e.

$$v(p) = \frac{\sum_{t \in p} e(t)}{|p|} \quad (1)$$

where p is a packet, $t \in p$ are the n-grams of p , $|p|$ is the number of n-grams found in p , and $e(t)$ returns the embedding for n-gram t .

- **Learning and Classification:** Once we have each packet translated into fixed-size feature vectors, we then pass those feature vectors to a supervised machine learning approach for training and then testing on unseen data.

Intrusion detection is an important area of research, vital for protection of national infrastructures, intellectual property, financial systems, privacy, and safety; however, the problem is a moving target, an arms race between defenders and attackers, along with constant evolution of the underlying technologies. There is evidence of growing sophistication among malicious actors. Symantec reports that the number of targetted attack groups, i.e. groups that are professional, highly organized, and target specifically rather than indiscriminately, grew at a rate of 29 groups a year between the years of 2015 to 2017, from a total of 87 to 140 [25]. Also, as evidence of constant change in the cyber arena, the number of IoT (Internet of Things) attacks grew by 600%, an increase of 54% of mobile malware variants, and an 80% increase in Mac malware.

We view our contribution as a way to increase the rate that defenders can evolve their methods to protect networks and infrastructure. Instead of manually hand-crafting features, which is error prone and difficult to determine impact, we can rely upon our Packet2Vec approach to automatically calculate features of interest.

The rest of this paper is organized as follows: Section 2 describes our approach in detail, including the steps we took to parallelize our solution. Section 3 presents the results of using our approach on a large cyber data set. Section 4 covers related work. Section 5 concludes.

2 Approach

In the introduction, we presented our approach at a high-level. However, applying Word2Vec on cyber data is challenging due to amount of information. In particular, we examined the DARPA 2009 data set [10]. This data set spans a period of 10 days, from November 3rd to November 12th, 2009. It is broken up into files that are just over 1 billion bytes (954 MBs), where each file represents 1-6 minutes worth of traffic. In this work we examined the first day, which is roughly 15.5 hours (it starts after 8:30 am) and comprises 558.8 GBs in total packet data. Due to the size of the data, we needed to create an iterative process for training our model.

Our solution is a combination of C++ code that is then exposed to python using Boost python [2]. We developed most of our implementation in C++ for performance, but then exposed it to python so that we could integrate with the Tensorflow library [1] for creating the embeddings for the n-grams, and the Scikit-learn library [6] for the classifier models to make predictions on whether the packets are benign or malicious. We also took efforts to parallelize the code using standard C++ features such as `std::thread` to manually instrument the code. As we discuss the implementation, we will highlight the parallelization. Also, in Section 3.1, we will discuss the parallel performance of the code.

Figure 1 gives an overview of the iterative approach. The first phase (pseudocode found in Algorithm 1), creates a dictionary, mapping n-grams to integer identifiers. The first phase begins by iterating through all pcap files used for training, n-gramming each packet, and incrementing the counter for each n-gram. After obtaining counts for each n-gram found in all the training files, identifiers are assigned for the top $|V|$ n-grams, where $|V|$ is the size of the vocabulary, a hyperparameter. Concerning memory utilization, we only load one pcap file at a time. Also, the dictionary is limited by the number of found n-grams. We used 2 byte n-grams, which at most has 2^{16} possible values.

The actual implementation of Algorithm 1 is a bit more nuanced as we structured it in such a way to enable parallelization. We first iterate in parallel over all packets and n-gram them. This is embarrassingly parallel and requires no inter-thread coordination. The end result is a vector of vector of n-grams. Then we flatten the vector of vector of n-grams into a single vector of n-grams, again in parallel. Finally, we hand the single vector of all n-grams to the dictionary, which updates the frequency counts for each n-gram. This is the only loop that requires coordination between threads, as two threads can potentially try to update the count for the same n-gram; however, adding mutexes around the update routine makes it thread safe. After all files have been processed, we also parallelize the implementation of lines 15 - 18. We need the dictionary for later phases, so we write it out to disk on line 19.

The second phase (Algorithm 2) utilizes the dictionary created from the first phase to translate the pcap files into integers. We iterate through each pcap file (line 1), creating two data structures for each pcap file. One data structure is a list of integers (line 2), which is the pcap file translated into integers using the dictionary. There is also a vector of vector of integers (line 3), which is the

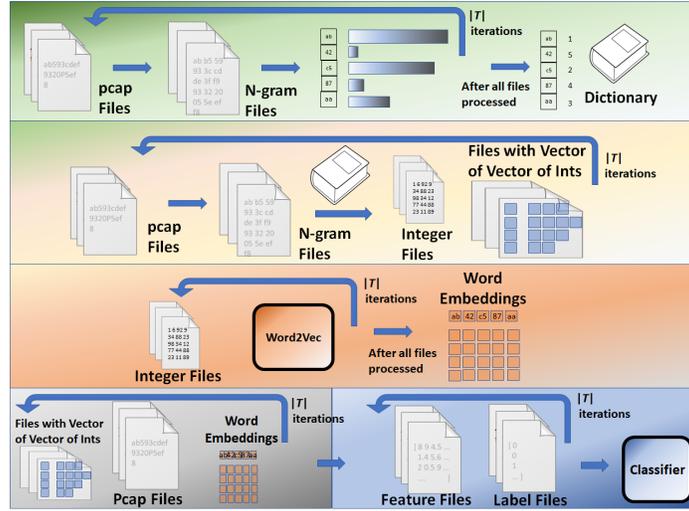


Fig. 1: Implementation of iterative pcap processing approach. The first phase creates a dictionary, mapping n-grams to integer identifiers. The dictionary is utilized in the second phase to transform the raw pcap data into integer vectors which are saved on disk. In the third phase, a Word2Vec approach is applied to the 1D integer vectors to create the n-gram embeddings. These embeddings are used in conjunction with the 2D integer vectors to create feature vectors (fourth phase) which are then used for training in the final phase.

Algorithm 1 Training Phase 1: Creating the Dictionary

```

1:  $T$ , the set of pcap files used for training.
2:  $D$ , a dictionary mapping from n-grams to integers.
3:  $|V|$ , the size of the vocabulary.
4: for all  $f \in T$  do
5:   for all  $packet \in f$  do
6:      $ngrams \leftarrow ngram(packet)$ 
7:     for all  $ngram \in ngrams$  do
8:        $D[ngram] \leftarrow D[ngram] + 1$ 
9:     end for
10:  end for
11: end for
12:  $keys \leftarrow sort(D)$  ▷ Keys sorted by decreasing frequency
13:  $D.clear()$  ▷ Clear out counts
14:  $i \leftarrow 1$ 
15: while  $i \leq |V|$  do
16:    $D[key[i]] \leftarrow i$ 
17:    $i \leftarrow i + 1$ 
18: end while
19:  $write(D)$ 

```

same as the integer list, but now indexed by packet. After processing a pcap file, we write out the list of integers (line 13) and the vector of vector of integers to disk (line 14). This again allows us to process all of the large pcap files without exceeding memory limits. We also parallelize the for loop of line 1. Each of the packets can be handled independently, so it is embarrassingly parallel.

The third phase is where we create the word embeddings, i.e. vectorized representations for each n-gram in the vocabulary. The process is described in Algorithm 3 in high level pseudocode. We iterate over all the integer files (pcap files translated by the dictionary into a single sequence of integers). On the first iteration we create an embedding model based on the first integer file using a standard word2vec approach. This creates a matrix of size $|V| \times \text{embedding_size}$, where each row corresponds to the learned vector representation of an n-gram. This first embedding matrix serves as the starting point for the next iteration of applying word2vec to another integer file. We continue in this manner until all integer files have been processed.

Algorithm 2 Training Phase 2:
Translating Pcap Files

```

1: for all  $f \in T$  do
2:    $l$ , a list of integers
3:    $vv$ , a 2D vector of integers
4:   for all  $packet \in f$  do
5:      $v$ , a vector of integers
6:      $ngrams \leftarrow ngram(packet)$ 
7:     for all  $ng \in ngrams$  do
8:        $l.push\_back(D[ng])$ 
9:        $v.push\_back(D[ng])$ 
10:    end for
11:     $vv.push\_back(v)$ 
12:  end for
13:   $write(l)$ 
14:   $write(vv)$ 
15: end for

```

Algorithm 3 Training Phase 3:
Creating Word Embeddings

```

1:  $L$ , the set of files with lists of integers
2:  $first\_run \leftarrow True$ 
3: for all  $l \in L$  do
4:   if  $first\_run$  then
5:      $E \leftarrow create\_model(l)$ 
6:      $first\_run = False$ 
7:   else
8:      $E \leftarrow update\_model(l, E)$ 
9:   end if
10: end for
11:  $write(E)$ 

```

The method for training the model is a standard word2vec approach. We use the skip-gram model [18] with noise contrastive estimation [11]. The basis of this approach is for the network to predict the context given a target word. However, with noise contrastive estimation, it becomes a logistic regression problem where the network is making a binary classification for each word in the vocabulary of whether or not it came from the distribution of context words or from the noise distribution (unrelated words). The hyperparameters associated with this approach include the following. In parenthesis we specify the value we used in our experiments. *Batch size* (128): The number of words considered at one time. *Skip window* (1): How big of a context window to consider. A value of one selects words to the left and right of the target word. *Num skips* (2): The *batch size* is divided by *num skips* to determine the number of skip windows. *Embedding size* (128): The size of each embedding vector. *Num negative* (64):

The number of negative examples used per batch. *Num steps* (100000): How many batches to create and from which to train.

The fourth phase utilizes the word embeddings in conjunction with the two dimensional integer vectors to create the feature files. Each feature file is a matrix where each row represents the features derived for a packet. On line 3 we iterate over the two dimensional integer vector files, VV . On line 6 we iterate over each vector, v , within the two dimensional integer vector, vv . v is a vector of integers, representing the n-grams of the original packet translated using D , the dictionary from Algorithm 1. To create a single representation for the entire packet, we use the simple strategy of averaging the embeddings (lines 9 - 12). In the end, we write out each feature matrix, X , to disk (line 15).

There is also another process for producing labels for the data. The DARPA-2009 dataset has a spreadsheet with labels; however, the labeling is not at the individual packet level. It lists times, IP addresses, and ports used by malicious traffic. Thus, to create labels, we read in the original pcap files and evaluate each packet, checking if the parameters of the packet match those of an entry in the label spreadsheet.

Algorithm 4 Training Phase 4:
Create Feature Vectors

```

1:  $VV$ , set of files with 2D vector of
   integers
2:  $E$ , the word embeddings indexed
   by integer identifier
3: for  $i \leftarrow 1$  to  $|VV|$  do
4:    $vv \leftarrow VV[i]$ 
5:    $X$ , a matrix of features
6:   for  $j \leftarrow 1$  to  $|vv|$  do
7:      $v \leftarrow vv[j]$ 
8:      $x$ , a vector of features
9:     for all  $integer \in v$  do
10:       $x \leftarrow x + E[integer]$ 
11:     end for
12:      $x \leftarrow x/|v|$ 
13:      $X[j] \leftarrow x$ 
14:   end for
15:    $write(X)$ 
16: end for

```

Algorithm 5 Training Phase 5:
Train Classifier

```

1:  $X_{files}$ , the list of feature files.
2:  $y_{files}$ , the list of label files.
3:  $n_{est}$ , the number estimators per
   file.
4:  $rfc \leftarrow RFC(warm\_start =$ 
    $True, n_{est})$ 
5:  $i \leftarrow 0$ 
6: for  $i \leftarrow 1$  to  $|X_{files}|$  do
7:    $X \leftarrow X_{files}[i]$ 
8:   if  $X$  has positive then
9:     if  $i \neq 0$  then
10:       $rfc.n_{est} += n_{est}$ 
11:     end if
12:      $y \leftarrow y_{files}[i]$ 
13:      $rfc.fit(X, y)$ 
14:   end if
15: end for
16:  $write(rfc)$ 

```

The last phase of training is to train an actual classifier. After phase 4, we finally have the data in a format that can be ingested by a standard machine learning algorithm. We have a set of files that contain the feature vectors for each packet, and we have another corresponding set of files that have a binary label indicating a benign/malicious packet. Algorithm 5 outlines the iterative approach to learning. In particular we show pseudocode related to the Random Forest Classifier [5], but it can be easily generalized to other machine learning algorithms. An important point to note here is the *warm_start* parameter on line

4. Since we are training in batches over many files, we need to maintain what was learned from earlier files. The *warm_start* parameter of Scikit-learn [6] is used when multiple calls to the *fit* function are used. In the case of the Random Forest Classifier, a number of estimators (trees) are created per file. However, this doesn't work if a file does not contain any malicious examples. On line 8 we skip any files that do not have malicious packets. What *warm_start* means differs depending on the classifier used. For example, with neural networks we would initialize the model with the weights learned from training on previous files.

3 Results

In this section we discuss two aspects of performance: 1) the throughput achieved when applying a trained classifier, and 2) the classifier performance in detecting malicious network activity. The system we used for our experiments was a DGX [22], a supercomputer designed for accelerating deep learning applications with powerful GPUs. However, except for the Packet2Vec portion that creates embeddings, our code primarily uses the CPU. The CPU is a dual Intel Xeon 20-core E5-2698 v4 2.2 GHz processor with 512 GBs 2133 MHz DDR4 memory. There is some variability to the timing of runs as other users are also using the system concurrently.

We tested our implementation on the DARPA-2009 data set [10]. DARPA-2009 is a generated data set covering a period of time from November 3-10, 2009. Traffic is simulated between a /16 local subnet that goes through a cisco router to the Internet. There are a variety of protocols (e.g. HTTP, SMTP, DNS) and malicious activities (e.g. DDoS, Phishing, port scans, spam bots). For this work, we treat all the malicious categories as single class so the problem is binary classification: malicious or benign. We evaluated our approach on the first day's worth of data (about 15.5 hours because the data starts around 8:30 am). In total for the first day there are 600 pcap files, each 1 billion bytes (954 MBs). Groundtruth labels are provided in the form a spreadsheet specifying the IPs, ports, and a bounding time window of when an attack occurred. For the portion we used, malicious activity accounted for 0.46% of the the total packets.

3.1 Processing Time

In this section we report on the processing time for applying a trained classifier on unseen data. It is important that our approach be able to keep pace with data creation. While application of a trained machine learning model is generally not a concern - testing is often orders of magnitude faster than training - our approach does have significant preprocessing steps. To classify unseen data, we need the following as input: 1) a pcap file, 2) the dictionary from n-grams to integers (created during Algorithm 1 and written to disk on line 19), 3) the n-gram embeddings (created from Algorithm 3 and written to disk on line 11),

and 4) the trained classifier (created during Algorithm 5 and written to disk on line 16).

The overall process of applying a trained classifier to unseen data is described below. We will make note of which portions are serial, serial but could be parallelized, and already parallelized.

1. Read pcap object: We read in a pcap object. Unless there is parallel I/O, this is largely a serial operation and cannot be parallelized.
2. N-gram the packets: For each of the packets in the pcap object, we n-gram them. This step has been parallelized.
3. Translate the n-grams into integers: Using the dictionary, we translate each vector of n-grams into a vector of integers. This step has been parallelized.
4. Create the feature matrix: This step takes the translated packet data of integers and converts them into embedding vectors, averages the embeddings, and then fills a matrix that has all the feature vectors. This step should be parallelizable but since we use a python object within C++ as the feature matrix, we run into issues with the Python global interpreter lock only allowing one thread. This should be surmountable, but will require a deeper dive into Boost python [2] and the NumPy C-API, which is C-based API for manipulating NumPy data structures (the feature matrix is a *NumPy.ndarray*).
5. C++ to python overhead: The function to create the feature matrix is written in C++ but we added a python interface. The python function reports on average 13.6 seconds more than the corresponding C++ implementation. We hypothesize this may be due to memory transfer costs. Regardless, this will be difficult to optimize without a deep exploration into Boost python.
6. Making predictions on the feature matrix: Here we apply the trained classifier to the now prepared feature matrix. We use the Scikit-learn library [6] for the machine learning models. This step could also be parallelized using one of the python libraries for parallel execution, but we have not taken that step yet.

To evaluate the parallel performance of the pipeline to apply a trained classifier to unseen data, we trained a Random Forest Classifier [5] on one pcap file and then tested it on another pcap, varying the number of threads. Figure 2 gives the overall time while Figure 3 provides the relative speedup as we increase the thread count. As expected, the parallel portion's total time decreases as we increase the number of threads, though the overall speedup plateaus around 10 threads.

Since we have good understanding of which portions of the program are parallel and which are serial, using Amdahl's law we can estimate the maximum achievable speedup: $Speedup(t) = \frac{1}{(1-p) + \frac{p}{t}}$, where we can think of t as the number of cores applied to the program and p is the proportion of the code that benefits from parallel execution. As $t \rightarrow \infty$, the equation becomes just $Speedup(t) = \frac{1}{1-p}$. Table 1 shows the maximum theoretical speedup based upon the times from using one thread. The *Current* row shows the times for the parallel and serial portions for our current implementation. Based on those numbers,

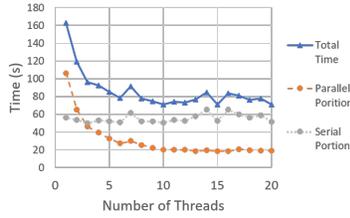


Fig. 2: Time for Testing One File: We apply a trained Random Forest Classifier to unseen data and report the times. The portion of the code that has been parallelized shows improvement up to ten threads.

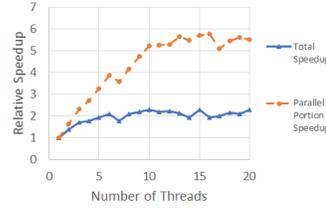


Fig. 3: Relative Speedup: Same data as Figure 2 but now showing relative speedup of the overall testing phase and the parallel portion.

our maximum speedup is about 2.9. Experimentally, we achieved a 2.3 speedup with ten threads. If we parallelized steps four and six, which certainly seems possible, then the maximum speedup is close to 9.2. Of course this is only single node speedup, and we can obtain greater aggregate throughput on a distributed system. If pcap data is ingested on multiple nodes, the task of classifying network traffic is embarrassingly parallel once the dictionary, embeddings, and trained classifier have been distributed.

Table 1: Theoretical Speedup

	Parallel Portion	Serial Portion	Max Speedup
Current	106.4	56.5	2.9
Future	145.2	17.7	9.2

Table 2: Testing Throughput

	Num Files	Time (hours)	Size (GBs)	Rate (MB/s)
Data creation	600	15.5	559	10.3
RFC 10	600	14.5	559	11.0
RFC 820	300	25.5	279	3.1
Naive Bayes	300	7.6	279	10.5

We also did some longer runs of applying a classifier to large sets of pcaps to gauge average throughput. Table 2 summarizes the results. For the simpler models, we can classify at about 10.5-11 MB/s while packet data is created at an average rate of 10.3 MB/s. The original data (the first day of DARPA-2009) comprises 15.5 hours and 600 files. We ran the Random Forest Classifier trained on one pcap on the entire data set. We also ran another Random Forest Classifier that was trained on 300 files and tested it on the other 300 files. Similarly, we trained a Naive Bayes classifier on 300 files and tested on the other half. For all the runs we utilized ten threads.

The difference between the two Random Forest Classifiers is that the one trained on one pcap file has ten estimators while the one trained on 300 files has 820 estimators. The difference comes from the fact that in order to incorporate

knowledge from other files to an existing Random Forest Classifier, we had to increase the number of estimators, essentially creating additional trees for each file. Thus, the Random Forest with 820 estimators has a much lower throughput because the longer predictions times (about 6 seconds versus 214 seconds). In the future, we plan to parallelize the prediction for loop which will likely make the difference in throughput less drastic. The Random Forest Classifier with ten estimators and the Naive Bayes were able to keep pace with the data creation rate.

3.2 Classifier Performance

We tested out two classifiers, the Random Forest Classifier [5] and Gaussian Naive Bayes [7]. We split the first day of DARPA-2009 into two sets of 300 files, one for training and one for testing. We listed all 600 files and gave training the even files and testing the odd files. This gave both sets representative data throughout the day.

We report two metrics, the area under the curve (AUC) for both the Receiver Operating Characteristic (ROC) curve and the Precision/Recall curve. The ROC curve plots true positive rate against the false positive rate as the threshold is varied. A perfect score for the AUC is 1.0. The ROC is known to provide overly optimistic results when data skew is present, as is with DARPA-2009.

The Precision/Recall curve emphasizes how good the predictions are for the minority class (i.e. malicious traffic). Precision is defined as the true positives divided by the true positives and false positives. So it is the fraction of results that are correct returned by the model: $Precision = \frac{TP}{TP+FP}$. Recall is defined as the number of true positives divided by the true positives plus the false negatives: $Recall = \frac{TP}{TP+FN}$. This gives you the fraction of the entire target class that are being returned by the model.

Table 3 gives an overview of both classifiers and both metrics. The AUC ROC metric gives a somewhat optimistic impression of the classifier’s skill, with values between 0.988 and 0.996, while the AUC or the precision/recall curve range between 0.604 and 0.667. The AUC of the precision/recall curve is probably more useful as it gives an idea of how good the classifier does at predicting the minority class. Figures 4 and 5 present the ROC and precision/recall curves for the Random Forest Classifier, respectively, while Figure 6 and 7 are for Gaussian Naive Bayes.

Table 3: Classifier performance

	AUC ROC	AUC Precision/Recall
Random Forest Classifier	0.996	0.604
Gaussian Naive Bayes	0.988	0.667

In both cases, there is a significant change in the precision/recall curve when recall is about 0.94. For Gaussian Naive Bayes, the plot is a little deceptive as

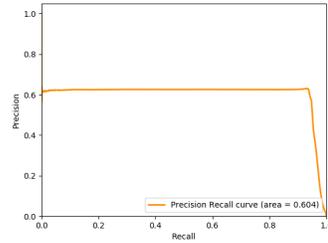
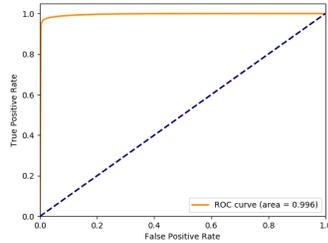


Fig. 4: Random Forest Classifier - Receiver Operating Characteristic Fig. 5: Random Forest Classifier - Precision/Recall

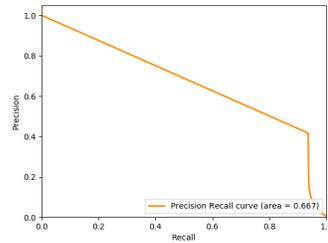
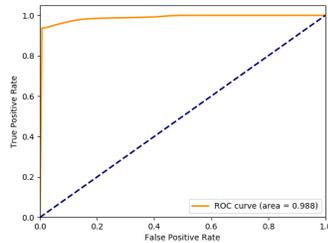


Fig. 6: Gaussian Naive Bayes - Receiver Operating Characteristic Fig. 7: Gaussian Naive Bayes - Precision/Recall

the first point from the data is $(0.937, 0.417)$ with a threshold of 1, meaning that any prediction less than one was considered benign. The point at $(0, 1)$ is by definition. We believe there is a large class of malicious behavior, likely the DDoS traffic, that both classifiers have a relatively easy time predicting. The transition at $recall = 0.94$ is likely for the other classes of malicious behavior.

Tables 4 and 5 provide some points along the precision/recall curve for the two classifiers, along with the corresponding F1 score. This is to give an idea of the tradeoff between finding malicious behavior and dealing with false positives. For instance, the first row of Table 4 shows that the Random Forest Classifier can find 98.8% of the malicious traffic, but you have to deal with about 94% of the returned results being false positives. If that is too many, one could use the threshold of the third line, where about half of the returned results are actually malicious and you still catch 95% of the total malicious behavior.

4 Related Work

A work similar to our own is that of Lotfollahi et al. [14] and their approach called *Deep Packet*. They focus on two problems, traffic characterization (e.g. identifying peer-to-peer traffic) and application identification (e.g. identifying traffic emanating from Skype or Tor), and use raw packet data as their data source. Like our approach, they avoid hand crafted features, but instead of a

Table 4: Random Forest Classifier - Precision Recall

Precision	Recall	Threshold	F1
0.060	0.988	0.006	0.113
0.108	0.981	0.009	0.195
0.504	0.951	0.029	0.659
0.630	0.930	0.285	0.751

Table 5: Gaussian Naive Bayes - Precision Recall

Precision	Recall	Threshold	F1
0.050	0.963	2.0e-134	0.095
0.010	0.947	1.31e-118	0.181
0.417	0.937	0.999	0.577

Word2Vec-based approach, they directly feed the packet bytes into a deep learning architecture. Packets are truncated or padded to be 1500 bytes long, and then fed into either a 1D convolutional neural network or a stacked autoencoder.

There are several papers that use deep learning, but they apply the network to already derived features. For the most part they test out deep learning strategies on either KDD or NSL-KDD [24, 26]. KDD is a challenge dataset from 1999 with artificially generated network data. The data was composed of benign and malicious connections, with each connection comprising 41 features. NSL-KDD is a modification of the original KDD data set to remove redundant records.

Javaid et al. [12] use Self-taught Learning [23] on NSL-KDD. Self-taught learning is an approach where you first use an unsupervised machine learning technique to create another representation of the data. For example, Javaid et al. use an autoencoder to translate the NSL-KDD feature set into a smaller representation. This new representation is then used as the basis for classification in a supervised training algorithm. Yin et al. [27] also employ deep learning, this time with recurrent neural networks, but they also test their approach on NSL-KDD. We agree with the conclusions of Malowidzki et al. [16], that many of the labeled public datasets are outdated, including NSL-KDD.

In terms of work that has examined the same data set, Moustafa and Slay [19] ran *tcptrace* on the first 30 files of DARPA-2009 to create flow-based features from which they filter down to 11 features in total. It is somewhat difficult to compare their work with ours as they are doing classification at the flow level, rather than at the packet level as we do. Also, they only examine 30 files, of which they report that 99.995% of the malicious activity is related to DDoS, while our 600 files covers a much broader range of categories of malicious activity. Also, they report that malicious flows account for 45.5% of their data set. It may be a difference between flows and packets, but we found malicious packets to account for far less: 0.46%. Their best recorded model was a decision tree, that missed 10 positive examples (there were 12 total non DDoS flows) and had no false positives.

Ackerman et al. [3] also examines DARPA-2009. They divide up the data into temporal chunks of one minute each, resulting in 13,835 chunks over the ten days, with 1,848 being malicious (if any malicious activity occurred during the time period) and 11,987 benign. They then selected 25 features that were aggregate computations over the time intervals. They used diffusion maps [8] for dimensionality reduction. Then from a single initial point in the new feature

space, they expand to find all similar points by recursively adding ones that are within a certain distance of an existing point. They do not report precision/recall numbers, but from what they do state we calculated an average precision of 0.03 and an average recall of 0.08, both of which are considerably lower than our results. However, they obtain their results from a single example. for finding other instances of malicious behavior in unlabeled data.

Part of the allure of deep learning is the ability to extract relevant features. Other work that focuses on feature extraction include Ngyuen et al. [21], where they use sketches [20] to approximate values in the stream of network data and Field-programmable gate arrays (FPGA's) to increase throughput, achieving a rate of 21.25 Gbps. Das et al. [9] also use an FPGA-based approach and a Feature Extraction Module (FEM) based on sketches.

5 Conclusions

We have presented a novel application of Word2Vec, called Packet2Vec, that translates packets into vectorized representations. We have demonstrated promising results, with classifiers achieving an AUC of the ROC between 0.988-0.996 and an AUC of the Precision/Recall curve between 0.604-0.667. The method can be used on raw packet data and does not require any domain expertise to extract relevant features.

There are many possible avenues for future work: **Temporal phenomenon:** We completely ignored temporal information. Many detection strategies utilize temporal information to distinguish between human actors and bots. How to incorporate temporal information within a deep learning strategy for cyber data is unexplored to our knowledge. **Aggregating predictions:** We made classification at the packet level. However, to a human analyst, it is likely more useful to roll up predictions to the level of a flow, or an IP, or a domain. **Existing features:** While we rely upon the deep learning model to extract relevant features, augmenting with existing approaches could be a fecund avenue to explore.

We believe that deep learning has much to offer cyber analysis, and that this work is just an initial step into discovering solutions for pressing security problems.

Acknowledgment

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energys National Nuclear Security Administration under contract DE-NA0003525.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A.,

- Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), <https://www.tensorflow.org/>, software available from tensorflow.org
2. Abrahams, D., Grosse-Kunstleve, R.W.: Building hybrid systems with boost.python (2003)
 3. Ackerman, D.A., Averbuch, A., Silberschatz, A., Salhov, M.: Similarity detection via random subsets for cyber war protection in big data using hadoop framework (2015)
 4. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (surf). *Computer Vision and Image Understanding* **110**(3), 346–359 (2008). <https://doi.org/https://doi.org/10.1016/j.cviu.2007.09.014>, <http://www.sciencedirect.com/science/article/pii/S1077314207001555>, similarity Matching in Computer Vision and Multimedia
 5. Breiman, L.: Random forests. *Machine Learning* **45**(1), 5–32 (Oct 2001). <https://doi.org/10.1023/A:1010933404324>, <https://doi.org/10.1023/A:1010933404324>
 6. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. pp. 108–122 (2013)
 7. Chan, T.F., Golub, G.H., LeVeque, R.J.: Updating formulae and a pairwise algorithm for computing sample variances. In: Caussinus, H., Ettinger, P., Tomassone, R. (eds.) *COMPSTAT 1982 5th Symposium held at Toulouse 1982*. pp. 30–41. Physica-Verlag HD, Heidelberg (1982)
 8. Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., Zucker, S.W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences* **102**(21), 7426–7431 (2005). <https://doi.org/10.1073/pnas.0500334102>, <https://www.pnas.org/content/102/21/7426>
 9. Das, A., Nguyen, D., Zambreno, J., Memik, G., Choudhary, A.: An fpga-based network intrusion detection architecture. *IEEE Transactions on Information Forensics and Security* **3**(1), 118–132 (March 2008). <https://doi.org/10.1109/TIFS.2007.916288>
 10. Gharaibeh, M., Papadopoulos, C.: Darpa-2009 intrusion detection dataset report. Tech. rep., Colorado State University (2014)
 11. Gutmann, M.U., Hyvärinen, A.: Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *J. Mach. Learn. Res.* **13**(1), 307–361 (Feb 2012), <http://dl.acm.org/citation.cfm?id=2503308.2188396>
 12. Javaid, A., Niyaz, Q., Sun, W., Alam, M.: A deep learning approach for network intrusion detection system. In: *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (Formerly BIONETICS)*. pp. 21–26. BICT’15, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium (2016). <https://doi.org/10.4108/eai.3-12-2015.2262516>, <http://dx.doi.org/10.4108/eai.3-12-2015.2262516>

13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc. (2012), <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
14. Lotfollahi, M., Zade, R.S.H., Siavoshani, M.J., Saberian, M.: Deep packet: A novel approach for encrypted traffic classification using deep learning. *CoRR* **abs/1709.02656** (2017), <http://arxiv.org/abs/1709.02656>
15. Lowe, D.G.: Object recognition from local scale-invariant features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. vol. 2, pp. 1150–1157 vol.2 (Sept 1999). <https://doi.org/10.1109/ICCV.1999.790410>
16. Maowidzki, M., Berezinski, P., Mazur, M.: Network intrusion detection: Half a kingdom for a good dataset (04 2015)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *CoRR* **abs/1301.3781** (2013), <http://arxiv.org/abs/1301.3781>
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc. (2013), <http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf>
19. Moustafa, N., Slay, J.: Creating novel features to anomaly network detection using darpa-2009 data set. In: *14th European Conference on Cyber Warfare and Security (2015)*
20. Muthukrishnan, S.: Data streams: Algorithms and applications. *Found. Trends Theor. Comput. Sci.* **1**(2), 117–236 (Aug 2005). <https://doi.org/10.1561/0400000002>, <http://dx.doi.org/10.1561/0400000002>
21. Nguyen, D., Memik, G., Memik, S.O., Choudhary, A.: Real-time feature extraction for high speed networks. In: *International Conference on Field Programmable Logic and Applications, 2005*. pp. 438–443 (Aug 2005). <https://doi.org/10.1109/FPL.2005.1515761>
22. Nvidia dgx-1 datasheet (2017), <http://images.nvidia.com/content/technologies/deep-learning/pdf/Datasheet-DGX1.pdf>, accessed: 2017-08-18
23. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.Y.: Self-taught learning: Transfer learning from unlabeled data. In: *Proceedings of the 24th International Conference on Machine Learning*. pp. 759–766. ICML '07, ACM, New York, NY, USA (2007). <https://doi.org/10.1145/1273496.1273592>, <http://doi.acm.org/10.1145/1273496.1273592>
24. Revathi, S., Malathi, A.: A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection. *International Journal of Engineering Research & Technology (IJERT)* **2**, 1848–1853 (01 2013)
25. Symantec: Internet security threat report (2018)
26. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 data set. In: *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. pp. 1–6 (July 2009). <https://doi.org/10.1109/CISDA.2009.5356528>
27. Yin, C., Zhu, Y., Fei, J., He, X.: A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access* **5**, 21954–21961 (2017). <https://doi.org/10.1109/ACCESS.2017.2762418>

Handbags Classification Model via Deep Learning

Dieinison J. F. Braga¹, Leodecio Braz S.S.¹, Críston Pereira de Souza¹, and Ticiania L. Coelho da Silva¹

Federal University of Ceara, Brazil

{dieinison, leodeciosegundo}@alu.ufc.br, {cristonsouza, ticianalc}@ufc.br

Abstract. Handbags are essential items in the fashion world, becoming indispensable in a person’s wardrobe. Such items have led a significant financial growth for important companies in the industry. Recognizing specific model information from a handbag posted either from magazines or other web pages such as blogs and Facebook can help in many applications as multimedia retrieval, fashion recommendation, and fashion search. Recently, deep convolution neural networks (CNNs) have been widely used to extract image features and construct models capable of accurately classifying objects from images. In this work, we aim at exploring the use of CNNs based on proposed architectures in the ImageNet competition. We chose Inception as a neural network to train a classifier capable of learning features from different handbags models and brands. Our experiments were conducted on a real data set, and show the accuracy and the performance of our method for handbags brands/models detection. Besides, we evaluate the effect on the performance and models’ accuracy by including a smart cropping phase, which is a pre-processing step that crops the handbag region of interest of the input images before performing the model training or prediction.

Keywords: Deep Learning · Object Detection · Handbag Recognition.

1 Introduction

Handbags are essential items in the fashion world, becoming indispensable in a person’s wardrobe. Such items have led a significant financial growth for important companies in the industry. Recognizing specific model information from a handbag posted either from magazines or other web pages such as blogs and Facebook can help in many applications as multimedia retrieval, fashion recommendation, and fashion search. Another important application in this context is to analyze comments associated with a handbag image, which can be easily obtained from social networks. This application might help fashion companies marketing to understand the opinion of their clients.

Several proposals have been published for object recognition in images [29], [30], [14], [7], [19], [26]. However, handbag recognition belongs to the category of fine-grained object recognition which is a challenging problem even for human beings [29]. There are four main challenges in handbag recognition problem:

1. Depending on the material of the bag, bags may have a distorted shape;
2. *Intraclass* low similarity, that is, bags of the same model have different colors or shades;
3. Bags of the same model may become visually distinct due to lighting effects;
4. Different handbag brands can be very similar visually, which yields challenging to build a classifier capable of differentiating such brands/models.

Recently, deep convolution neural networks (CNNs) have been widely used to extract image features and construct models capable of accurately classifying objects from images. At 90% or more of the applications, there is no need to worry about proposing and building a CNN [1]. Instead of creating an architecture for a specific problem, it is possible to initially check some architecture already proposed on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competition that performs well, download a pre-trained model and adjust it to the problem’s dataset. It is rarely necessary to train a CNN from scratch or create an architecture from scratch [1]. The focus of this work is to explore the use of CNNs based on proposed architectures in the ImageNet competition to train a classifier capable of learning features from different handbag models and brands.

There exist several methods [21], [22], [23] to generate potential bounding boxes for bags in an image. In general, such methods output bounding boxes covering pre-defined objects (such as bags, person, airplanes, animals, among others) from the images. This bounding box extraction can increase the accuracy of object detection as showed in [2], [28]. In this paper, we consider only the bag cropped with the highest confidence in each image, and we call this phase in our proposal as smart cropping.

We present an extensive experimental evaluation conducted on a real-world dataset that assesses the accuracy of our proposed deep CNN models for handbag brands detection. Indeed, we evaluate the effect on performance and models’ accuracy by including the smart cropping phase. We produce the bounding box by running promising methods for object detection: Faster R-CNN [23], YOLOv3 [22], Single Shot MultiBox Detector (SSD) [18], among others. We observe a trade-off between such methods concerning accuracy and performance. We also provide some insights on how to choose proper values for model parameters which is a challenging problem on the machine learning domain.

The remainder of the paper is structured as follows: Section 2 presents our proposed solution. Section 3 presents the experimental evaluation. Finally, Section 4 presents the related works, and Section 5 draws the final conclusions.

2 Handbag Recognition

Figure 1 shows our proposed pipeline for training a handbag recognition classifier and predict the handbag model from an input image. Before training a classifier, there is a need to collect lots of images already classified. In this paper, we aim at training a model by using more than 4,000 images collected from more than 200 different handbags models that belong to around 40 brands. Our solution

cropping the handbag region of interest (ROI) of the input images before performing the model training or prediction. We call this phase *smart cropping*, which is a proper step since in real cases the handbag is not centered on the image, i.e., there are lots of external information that would potentially mislead the classification model.

Therefore, we propose smart cropping as a pre-processing procedure, to locate and extract the ROI in which the handbag is located on a given input image. There exist several object detection methods, which were design to find ROIs on an image. [10] studies the trade-off between efficiency and effectiveness in some state-of-the-art object detection methods, i.e., Faster R-CNN[23], R-FCN[5], and SSD[18] by looking at Mean Average Precision (mAP) metric using MS COCO dataset [16]. Overall, [10] noticed that SSD is faster than Faster R-CNN, on the other hand, Faster R-CNN provides better results for mAP.

In this paper, we study five different solutions for object detection, which were used on [10]: Faster R-CNN[23] with Inception Resnet V2[24], SSD[18] with MobileNet[9] and R-FCN[5] with Resnet 101[8]. Apart from those, we also investigate the widely used YOLOv3 [22] and RetinaNet [15]. In the experiment section, we discuss the trade-off between detection accuracy and execution time. Moreover, we also observe a gain regarding model’s accuracy and loss metrics when such model is training with images after the smart cropping compared with the model trained using the original images.

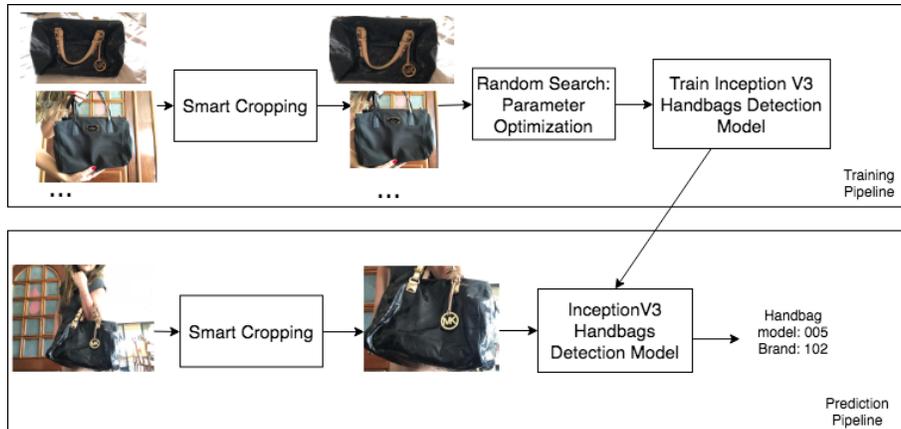


Fig. 1: Overview of the proposed handbag recognition solution

CNNs are at the core of most state-of-the-art computer vision solutions for a wide variety of classification tasks. ImageNet[6] challenge leveraged many successfully deep learning models appropriate to apply to a larger variety of computer vision tasks, for example to object-detection, segmentation, human pose estimation, video classification, object tracking, and super-resolution [25].

A typical CNN is usually composed of three main types of layers: Convolutional Layer, Pooling Layer, and Softmax Layer. The convolutional layers generate feature maps by applying consecutive convolution between a group of trainable convolutional kernels and the input. From this, the network can learn filters that activate when some type of visual feature or a specific pattern such as an edge of some orientation appears. The pooling layers are utilized to progressively reduce the spatial size of the representation to reduce the dimension of feature maps and computation in the network, and hence to also control overfitting. The softmax layers output the normalized probability of each label and in general, are used at the end of a CNN [1].

The paper [4] provides an informative view comparing the trade-off between accuracy and computational cost for different deep learning architectures proposed on the ImageNet challenge, such as VGG16, VGG19, Inception-v3, Resnet-50, among others. Overall, ResNet and Inception architectures surpass all other architectures.

It has also been shown that Inception is appropriate in big-data scenarios, where a huge amount of data needed to be processed at reasonable cost or scenarios where memory or computational capacity is inherently limited, for example in mobile vision settings [25]. This is a strong reason why we chose to build our solution upon Inception architecture in this paper. The need to find the handbag model or brand is a common demand in mobile apps with the widespread adoption of smartphones and the popularity of e-commerce.

However, to achieve better model accuracy, there is a need to search for the best parameter values. There exist several model parameters to tune while training a CNN model. In this work, we perform a widely used method, called random search [3], to choose appropriate values for learning rate and weight regularization. We divide the random search into two main steps: coarse search and fine search.

Basically, in the coarse search, we randomly sample a set of pairs of learning rate and weight regularization according to a pre-defined range of values. These pairs are trained over a number of epochs. Next, we rank the pairs according to the achieved model accuracy. Then, we start the fine search by looking at the range where the best-ranked values for learning rate and weight regularization fall within. We pick out new pairs from such range, and we train the model again over more epochs. After that, the pair of learning rate and weight regularization which provides the highest accuracy is the best configuration. In Section 3, we provide more details about how we found the model parameter values for our problem. We also refer the reader to [3] for further details.

3 Experimental Evaluation

In this section, we assess the quality and efficiency of the results by performing our proposed approach. We use "accuracy" as the metric to compare the models. Accuracy is the proportion of correctly classified images, i.e., the number of correctly classified images divided by the total number of images.

3.1 Handbag Images Dataset

The dataset contains more than 4,000 images collected from more than 200 different handbags models non-uniformly distributed to around 40 brands. The dataset contains professional and non-professional images, which some of them were taken at a photographic studio with a white background and others were taken on streets or squares, for instance. The collected dataset presents challenges to this problem as also shown in Figure 2: (i) different handbag brands can be very similar; (ii) images with different size; (iii) images with bags not in evidence or not centralized; (iv) bags of the same model may become visually distinct due to lighting effects. Thus, to correctly recognize the handbag models, our approach tries to overcome such challenges.



Fig. 2: Image samples to motivate our problem.

While conducting the experiments, we construct four datasets, and we named them: *Handbags-I*, *Handbags-II*, *Handbags-III* and *Handbags-IV*.

From the collected images, *Handbags-I* dataset contains only classes (brands) with the number of images above than a threshold. We chose to set the threshold as the average of images per class, that is equivalent to fourteen (14). The classes with less than 14 images were removed. *Handbags-II* is derived from the previous dataset after performing the smart cropping phase applying *Faster R-CNN* network (this network presents the best accuracy as reported afterward). *Handbags-III* is derived from the *Handbags-I*, however, considering only images that show the foreground handbag region. *Handbags-IV* is derived from the *Handbags-II*, however, considering only images that show the foreground handbag region. All the datasets contain the same number of classes (handbag brands). We perform data augmentation for all datasets. We adopt the following parameters for the data augmentation: maximum 20 degrees rotation range, maximum 20% zoom range, random perspective skewing (25% probability), and random horizontal flip.

All the datasets contain 88 classes (handbag brands). The data augmentation phase resulted in 17,729 images for *Handbags-I*, 17,729 images for *Handbags-II*, 14,837 images for *Handbags-III*, and 14,833 images for *Handbags-IV*. For each dataset, the images were divided into three different sets: 65% for training, 15% validation, and 20% for test. We ensure that the images from these three sets

are the same in *Handbags-I* and *Handbags-II*, of course after the smart cropping phase in the later dataset. The same applies to *Handbags-III* and *Handbags-IV* for validation and test sets. It’s worth to mention that the sizes of the images range from 82 x 62 to 3092 x 2576, due to this variation the images were resized to 224 x 224.

3.2 Choice of model parameters

As we mentioned before, we found proper model parameters using random search. In these experiments, we picked out 100 values for the pair learning rate and weight regularization during the coarse search and fine search. We train the model for five epochs in the coarse search, and 30 epochs in the fine search. After the coarse search, we rank the pairs according to the achieved validation accuracy and identify a smaller range of value containing the best pairs. After the fine search, we chose the pair of learning rate and weight regularization which provides the highest validation accuracy.

During the coarse search and fine search, the chosen values for learning rate and weight regularization fall within the ranges described in Table 1. As suggested in [1], we set the range of values for learning rate and weight regularization on a log scale. Notice that during the fine search, for each dataset, we found different ranges of values for the analyzed parameters. Furthermore, the best configuration values are different for each dataset.

Table 1: Results of hyper-parameter optimization

Dataset	Coarse Search range		Fine Search range		Best configurations	
	Learning rate	Reg. weight	Learning rate	Reg. weight	Learning rate	Reg. weight
Handbags-I	$(10^{-6}, 10^{-3})$	$(10^{-5}, 10^5)$	$(10^{-6}, 10^{-4})$	$(10^{-5}, 10^{-2})$	$5.9 \cdot 10^{-5}$	$8.8 \cdot 10^{-3}$
Handbags-II	$(10^{-6}, 10^{-3})$	$(10^{-5}, 10^5)$	$(10^{-6}, 10^{-4})$	$(10^{-5}, 10^{-3})$	$9.7 \cdot 10^{-5}$	$8.7 \cdot 10^{-5}$
Handbags-III	$(10^{-6}, 10^{-3})$	$(10^{-5}, 10^5)$	$(10^{-6}, 10^{-4})$	$(10^{-5}, 10^{-3})$	$6.1 \cdot 10^{-5}$	$4.5 \cdot 10^{-5}$
Handbags-IV	$(10^{-6}, 10^{-3})$	$(10^{-5}, 10^5)$	$(10^{-6}, 10^{-3})$	$(10^{-5}, 10^{-2})$	$1.4 \cdot 10^{-4}$	$7.2 \cdot 10^{-5}$

Another relevant parameter is the optimizer. We set Adam [12] as the optimizer, which is a simple and computationally efficient algorithm for gradient-based optimization of objective functions.

3.3 Smart cropping trade-off analysis

The difference between *Handbags-I* and *Handbags-II* is the smart cropping phase under *Handbags-II* construction. The same applies to *Handbags-III* and *Handbags-IV*. Therefore, we can evaluate the effect of smart cropping phase on the execution time and accuracy of the classification model.

As mentioned before, we consider five state-of-the-art object detection models to perform the smart cropping: Faster R-CNN, SSD, R-FCN, YOLOv3, and RetinaNet. These models have a significant variation on the execution time and

detection quality, thus allowing us to evaluate which one provides the best trade-off to our problem. To perform Faster R-CNN, SSD, R-FCN, we use the TensorFlow Object Detection API [10]. Indeed, these models were pre-trained with the MS COCO dataset. Since *handbag* is one of the labels of MS COCO, we have decided to use the pre-trained models of Faster R-CNN with Inception Resnet V2, SSD with MobileNet, and R-FCN with Resnet 101. In the experiments, we use ImageAI API[20] to perform YOLOv3 and RetinaNet.

First of all, we executed each smart cropping model (Faster R-CNN, SSD, R-FCN, YOLOv3, and RetinaNet) for the test set of *Handbags-I*. Table 2 reports in seconds the execution time to perform such processing per image. Moreover, we also collected if the model assigned the image to its correct label. The same steps apply for *Handbags-III* and *Handbags-IV* reported in Table 3.

From these experiments, we notice that the accuracy is higher when the input images were cropped using Faster R-CNN. However, Faster R-CNN model takes longer to crop an image than the other models. On the other hand, YOLOv3 runs faster to crop a given image, but the accuracy decreases. Thus, there is a trade-off between quality and performance. Imagine a real mobile application which detects handbag brands using our proposal. A user uploads a handbag image, and the app provides the handbag brand. In this case, using YOLOv3 can be more efficient since it is a real-time application. However, applications where the efficiency is not an issue, the best choice is to crop using Faster R-CNN.

Tables 2 and 3 also report the classification time per image. For each image cropped by a smart cropping model, we collected the execution time to classify such image using the model trained over *Handbags-II* (Table 2) and *Handbags-IV* (Table 3). We can observe, in general, that the classification time slightly decreases when the image is cropped by a proper smart cropping model. Thus, to improve the performance and the accuracy of the model classification, there is a need to choose the smart cropping model with the best accuracy. Faster R-CNN presents the best accuracy comparing to the others, and it helps to accelerate the classification time. Thus, we decide to adopt it in the experiments performed in the next section.

Table 2: Execution time and accuracy obtained from classification model trained using *Handbags-II*

Detection Models	Smart Cropping (seconds/image)	Classification (milliseconds/image)	Top-1 accuracy
Faster R-CNN	6.126	10.219	89.2%
R-FCN	3.830	10.183	87.1%
SSD	2.143	12.668	86.6%
RetinaNet	1.510	12.425	86.2%
YOLOv3	0.325	13.811	85.9%

Table 3: Execution time and accuracy obtained from classification model trained using *Handbags-IV*

Detection Models	Smart Cropping (seconds/image)	Classification (milliseconds/image)	Top-1 accuracy
Faster R-CNN	6.070	8.968	94.08%
R-FCN	3.802	8.982	92.77%
SSD	2.162	10.889	91.79%
RetinaNet	1.503	11.031	92.57%
YOLOv3	0.327	13.070	91.99%

3.4 Model result analysis

Table 4 provides the accuracy and loss of our proposed model over the test set for *Handbags-I*, *Handbags-II*, *Handbags-III* and *Handbags-IV*. It is worth to mention that after the parameters optimization described in Section 3.2, the validation images were inserted into the training set and the model was retrained. This procedure improved the model generalization and the accuracy in the test set.

Notice that the model trained using *Handbags-II* outperforms the one trained using *Handbags-I*. We found the same pattern when comparing *Handbags-IV* and *Handbags-III*. The model trained using *Handbags-IV* surpasses the accuracy of the model trained using *Handbags-III*. These results reinforce that the smart cropping phase in our proposal improves the model’s accuracy and decreases its loss.

As the training and test set of *Handbags-I* and *Handbags-II* are the same (of course, before the smart cropping in the latter dataset), we can only fairly compare the model trained using *Handbags-I* with the model achieved from *Handbags-II*. The same intuition applies for *Handbags-III* and *Handbags-IV* since both datasets present the same images on training and test set (again, before the smart cropping phase in the latter dataset).

Table 4: Accuracy and loss for the test set for each dataset. In bold are the best results.

Dataset	Top-1 Accuracy	Top-1 Loss
Handbags-I	84.09%	1.438
Handbags-II	89.54%	0.598
Handbags-III	90.37%	0.508
Handbags-IV	94.08%	0.346

4 Related Work

Fine-grained object recognition is a challenging problem and has recently received much attention in computer vision. On one hand, with the publication of

several datasets with detailed annotations such as birds [27], dogs [11], cars [13] datasets, among others. On the other hand, several proposals with innovative studies which achieved remarkable performance [29], [30],[14],[7],[19], among others. These works on fashion items mainly target on clothing retrieval [17], [19], or branded handbag recognition [29], [30], among other applications.

[14] proposes a machine learning system to compose fashion outfits automatically. [7] presents two fashion recommendation systems: one that suggests an item that matches existing components in a set to form a stylish outfit, and another that generates an outfit with multimodal (images/text) specifications from a user.

[19] introduce DeepFashion, a large annotated clothes dataset. Moreover, [19] proposes a deep learning classification model which learns clothing features by jointly predicting clothing attributes and landmarks. [17] tackles the problem of given a human photo captured occasionally on the street, finding similar clothing from online shops. [17] proposes a solution including two key components, i.e., human/clothing parts alignment to handle human pose variation and bridging cross-scenario discrepancies with an auxiliary daily photo dataset.

[29] proposes a handbag recognition framework. The framework consists of a proposed CNN detection model trained over a not public handbag dataset. [29] incorporates in its solution the symmetry property of the handbag for extracting handbag proposals. Then each proposal is fed into the CNN detection model. [30] also deals with handbag recognition. It tackles the problem of inter-class style similarity and the intra-class color variation, i.e., the first challenge means that different handbag brands can be very similar visually, and the second challenge means that bags of the same model can present different colors or shape. As a solution, [30] developed discriminative representations algorithms of handbag style and color.

The problem tackled in this paper is related to the [29] and [30] problems. However, our work differs in the deep learning architecture used as a solution and the smart cropping step presented on the training and test phases.

5 Conclusion

In this work, we investigated the use of Inception V3 as a deep network classifier capable of learning features from different handbags models that belong to different handbag brands. Our experiments were conducted on a real data set, and show the accuracy and the performance of our method for handbags brands/models detection. We also evaluated the effect on the performance and models' accuracy by including a smart cropping phase, which is a pre-processing step that crops the handbag region of interest of the input images before performing the model training or prediction. We concluded that performing smart cropping phase improves the model's accuracy and accelerates the classification time. As future work, we aim at investigating other deep neural networks as Inception V4 and ResNet. Another future direction is to develop a large scale handbags dataset.

References

1. Convolutional Neural Networks for Visual Recognition. (2018 (accessed November 5, 2018)), "<http://cs231n.github.io/>"
2. Angelova, A., Zhu, S.: Efficient object detection and segmentation for fine-grained recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 811–818 (2013)
3. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. *Journal of Machine Learning Research* **13**(Feb), 281–305 (2012)
4. Canziani, A., Paszke, A., Culurciello, E.: An analysis of deep neural network models for practical applications. arXiv preprint arXiv:1605.07678 (2016)
5. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems. pp. 379–387 (2016)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 248–255. Ieee (2009)
7. Han, X., Wu, Z., Jiang, Y.G., Davis, L.S.: Learning fashion compatibility with bidirectional lstms. In: Proceedings of the 2017 ACM on Multimedia Conference. pp. 1078–1086. ACM (2017)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
9. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
10. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: IEEE CVPR. vol. 4 (2017)
11. Khosla, A., Jayadevaprakash, N., Yao, B., Li, F.F.: Novel dataset for fine-grained image categorization: Stanford dogs. In: Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC). vol. 2, p. 1 (2011)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 554–561 (2013)
14. Li, Y., Cao, L., Zhu, J., Luo, J.: Mining fashion outfit composition using an end-to-end deep learning approach on set data. *IEEE Transactions on Multimedia* **19**(8), 1946–1955 (2017)
15. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence* (2018)
16. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
17. Liu, S., Song, Z., Liu, G., Xu, C., Lu, H., Yan, S.: Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 3330–3337. IEEE (2012)

18. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
19. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1096–1104 (2016)
20. Olafenwa, M., Olafenwa, J.: Imageai. <https://github.com/OlafenwaMoses/ImageAI/> (2018)
21. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. arXiv preprint (2017)
22. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
23. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
24. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI. vol. 4, p. 12 (2017)
25. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
26. Tangseng, P., Yamaguchi, K., Okatani, T.: Recommending outfits from personal closet. In: 2017 IEEE International Conference on Computer Vision Workshop (ICCVW). pp. 2275–2279. IEEE (2017)
27. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
28. Wang, X., Yang, T., Chen, G., Lin, Y.: Object-centric sampling for fine-grained image classification (2014)
29. Wang, Y., Li, S., Kot, A.C.: Deepbag: Recognizing handbag models. IEEE Transactions on Multimedia **17**(11), 2072–2083 (2015)
30. Wang, Y., Li, S., Kot, A.C.: On branded handbag recognition. IEEE Transactions on Multimedia **18**(9), 1869–1881 (2016)

Long Short-Term Memory-based Multi-Period Price Prediction for Portfolio Management

Zhengyong Jiang^{1,2,4} and Frans Coenen³

¹ Department of Electrical and Electronic Engineering, University of Liverpool based in Xi'an Jiaotong-Liverpool University, Suzhou, China

² Department of Mathematical Sciences, Xi'an Jiaotong-Liverpool University, Suzhou, China

³ Department of Computer Science, University of Liverpool, Liverpool, UK

⁴ Authors to whom correspondence should be addressed
zhengyong.jiang@xjtlu.edu.cn

Abstract. In this paper, we present a novel, machine learning-based approach for price prediction to portfolio construction in the context of multi-period trading. We use a combination of recurrent neural network (RNN) and a long short-term memory (LSTM) network for predicting the future prices and to perform constrained optimization on the portfolio update.

Evaluation using a series of back-test on a number of datasets obtained from Shanghai Stock Exchange (SSE) and National Association of Securities Deal Automated Quotations (NASDAQ) sources show that our proposed approach outperforms the most common conventional portfolio management technique, namely Robust Median Reversion strategy, on a number of different metrics. In back-test experiment, our proposed method offers an average of 148% returns over 360 trading periods with 100 stocks, compared to 124% returns using conventional technique, over the same period and number of stocks.

Keywords: Portfolio Management, Recurrent Neural Network, Long Short-Term Memory

1 Introduction

In financial context, the term “portfolio” refers to any combination of financial assets such as stocks, bonds and cash which are held by individual investors or managed by financial professionals, hedge funds, banks and other financial institutions [30]. Portfolio management is the decision-making process of allocating a specified amount of fund to a set of different assets, with the aim of maximizing the return under the same level of risk [5, 9]. This can be seen as a constrained optimization problem. To this end, the mean-variance model and the principle of efficient frontier are two most common techniques used to construct portfolios [30].

Owing to the underlying mathematical model, the mean-variance model is only suitable for single-period trading, and extending that to update multiple

periods results in computationally intractable solution. Moreover, nearly all of the tradings in the stock markets are multi-period [13]. As such, techniques based on mean-variance model are less useful in the context of multi-period trading.

In the recent past, machine learning-based approaches have become a popular choice for solving a number of financial, particularly portfolio management, problems [34], in addition to solving a number of problems from the signal processing domain [38, 40]. One of the common principles of machine learning is to supervise the machine to learn by examples — supervised machine learning. An underpinning aspect of learning here, in conventional methods, is identifying the features to learn. This process, often referred to as feature engineering, is central to most of the supervised learning. Among different machine learning-based approaches, deep learning has become a popular method for addressing a number of problems. Deep learning methods rely on neural networks and their variants to learn features themselves along with the task. As such, deep learning-based techniques have become an attractive choice for a number of tasks both for classification problems and for regression problems.

In the context of price prediction, the historical stock prices are treated as a time series and future prices are predicted by learning from the past. The novel technique we propose here in this paper combines deep learning method and constrained optimization for handling the portfolio construction problem in the context of multi-period trading. In particular, we use a variant of neural network called Recurrent Neural Network (RNN) along with another variant called Long Short Term Memory network (RNN-LSTM) to predict the future price of each asset and to update the concerning portfolio. The combination of networks we use in our method not only learns the features that are responsible for the fluctuations in price, but also remembers the past due to its long-term memory. The key contributions of this paper are two fold:

1. We propose a novel price prediction based on a combination of RNN-LSTM network that offers far superior results than the conventional Robust Median Reversion (RMR) method; and
2. We perform a thorough evaluation of our approach and validate its effectiveness by performing back-test on 100 real-world stocks. We compare our strategy against other strategies such as Passive Aggressive Median Reversion (PAMR) and Confidence Weighted Median Reversion Strategy (CWMR).

The rest of the paper is organized as follows. The background researches about portfolio construction is discussed in Section 2. In Section 3, we discuss our approach in solving portfolio management problem. The results of back-tests are then discussed in Section 4. Finally, we conclude the paper with directions for further research in Section 5.

2 Background

2.1 Problem Statement

For a given financial market, suppose that we are interested in the investment of d assets for n trading days altogether. At the beginning of t^{th} trading day, our investment for the d assets is denoted by the portfolio vector $\mathbf{b}_t = [b_t^1, \dots, b_t^d]^T$ where $b_t^j \in [0, 1]$ represents the proportion of wealth invested in the asset $j \in \{1, 2, \dots, d\}$ where $b_t^1 + b_t^2 + \dots + b_t^d = 1$. Following the investment, let vector $\mathbf{p}_t = [p_t^1, p_t^2, \dots, p_t^d] \in \mathbb{R}_+^d$ represent the close price of all d assets at the end of t^{th} trading days. The vector $\mathbf{x}_t = [x_t^1, \dots, x_t^d]^T \in \mathbb{R}_+^d$ gives the ratio of current close price to the previous close price for each asset $j \in \{1, 2, \dots, d\}$ at time t , i.e., $x_t^j = p_t^j / p_{t-1}^j$. At the end of t^{th} trading day, we achieve a period return $S_t = \mathbf{b}_t^T \mathbf{x}_t = \sum_{j=1}^d b_t^j x_t^j$. The aim of portfolio management is to design a strategy for determining the portfolio vector \mathbf{b}_t at the beginning of t^{th} trading day so as to maximize the final cumulative portfolio wealth $S_n = S_0 \prod_{t=1}^n (\mathbf{b}_t^T \mathbf{x}_t)$ where S_0 is the initial wealth at the beginning of trading. The strategy shall be measured based on the final cumulative portfolio wealth and other metrics which are introduced later in this paper.

In this research, we shall assume that the market is in a perfect liquidity, with zero impact cost situation, and the price of each stock is independent from one another. These assumptions are not trivial. The first assumption ensures that we could invest our capital in each asset with any possible proportion. The second assumption ensures that we could obtain price information immediately at any time nodes without any cost. The third assumption enables us to predict future price of each stock independently of others.

2.2 Related work

There are two main mathematical principles in the portfolio management problem, namely the efficient frontier principle [30] and the Kelly investment principle [18]. The mean-variance model, which is based on the efficient frontier principle, trades in the market according to expected return of the stock and risk (i.e., variance of the price of the stock) [3], is suitable for single-period portfolio management. The Kelly investment principle, which targets to maximize the expected return, focuses on multiple-period sequential portfolio management.

Traditional multiple-period sequential portfolio management methods can be classified into four categories, namely, the *Follow-the-winner*, *Follow-the-loser*, *Pattern-Matching* and *Meta-Learning* [24]. The first two categories are based on existing financial models such as mean reversion model and exponential gradient model. The *Follow-the-winner* algorithm is inclined to invest stocks in an upward trend while the *Follow-the-loser* algorithm is inclined to invest stocks in a downward trend. They may also be assisted by some online learning techniques, e.g., statistical techniques to improve the

performance [20] and to tune the parameter in them [23] [7]. The performance of these methods depends on the validity of models on different markets such as stock markets, futures markets or cryptocurrency markets. The *Pattern-Matching* algorithm selects part of historical data similar to the current situation for optimizing the portfolio based on some assumptions of the behavior of the market [12]. The last category, the *Meta-Learning* method, attempts to combine different categories to achieve better performance [37] [8].

Recently, a novel price prediction based strategy called Robust Median Reversion (RMR) is proposed [15]. The strategy uses the L1-median-estimate [15] on historical prices of assets to predict the future prices of assets. This is followed by an update on the portfolio using these predicted prices. However, there some factors unaccounted for such as financial crisis which cause the prices of assets to often fluctuate drastically [22]. As a result, the L1-median-estimate algorithm may fail to perform satisfactorily. In view of this, machine learning methods are applied to portfolio management in recent years.

Deep learning allows a system to automatically discover the representations needed for feature detection or classification from data [35], has had impressive performances in several areas such as image classification [38], speech recognition [40], sentiment analysis [2], machine translation [6], advertising [16] and urban design [33]. It exceeds 99% accuracy in MINST dataset for image classification and achieves 95% accuracy for speech recognition. Deep learning methods can also be applied to help investment managers to manage the portfolio by using historical market data [14] [28] or predict future price [31] [39]. However, these existing deep learning methods can only predict the stock price or obtain a efficient frontier of portfolio for a single time period. They have not been used in trading directly. In contrast, our research work is novel since we are the first to propose a strategy which combines price prediction and portfolio updating so as to output a unique portfolio vector to be directly used for multiple period trading.

3 Methodology

The RNN-LSTM approach proposed consists of two steps, namely, the price prediction step and portfolio updating step. In the first step, price prediction is carried out. It is a form of a regression problem in which historical stock prices are treated as a observations indexed by time. In our price prediction, the historical close price of each asset is seen as a time series. This is followed by portfolio updating in which the portfolio vector is updated based on the predicted price obtained earlier. Fig 1 gives a flow chart to describe how our strategy is carried out.

3.1 First Step: Price Prediction

We use RNN-LSTM to predict the close price of daily trading data. Fig 2 gives the architecture of the RNN-LSTM. The input vector of neural network is

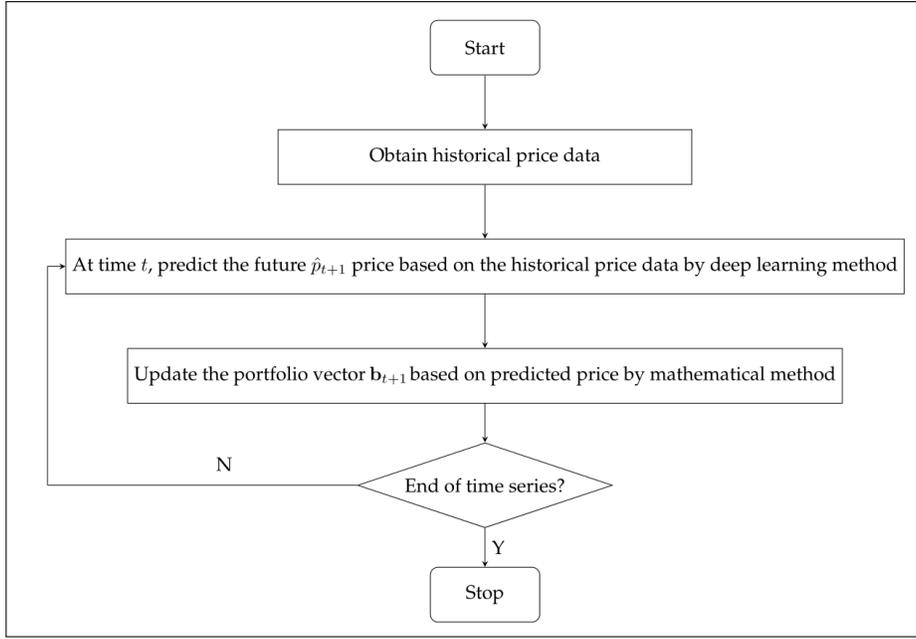


Fig. 1: A schematic description of the proposed RNN-LSTM technique. The first step of the approach predicts the future price for each single asset, and the second step updates the portfolio vector according to the predicted price.

a moving window $\mathbf{p}_t^j = [p_{t-i+1}^j p_{t-i+2}^j \dots p_t^j]^T \in \mathbb{R}_+^i$ containing the most recent i daily close price for the j^{th} single asset. In this research, we shall set i to be equal to 6 after hyper-parameter tuning. In Fig 2, the output vector \mathbf{h}_t^j is such $\mathbf{h}_t^j = [\hat{p}_{t-i+2}^j \hat{p}_{t-i+3}^j \dots \hat{p}_{t+1}^j]^T \in \mathbb{R}_+^i$ and the element \hat{p}_{t+1}^j is the predicted close price of the j^{th} single asset at time $t+1$ which be used for portfolio updating. The output window length is also set to 6.

Table 1 contains relevant information regarding the LSTM networks which is used in the training of a single asset after the parameter tuning. Therefore we have 100 neural network models for training 100 stocks. Also, to avoid the problem of over-fitting, the L1 regularization was added in our loss function and our loss function is defined as $\sum_{i=7}^n (\hat{y}_i - y_i)^2 + 0.01 \sum_{c=1}^m |\omega_c|$ where \hat{y}_i is the predicted value, y_i is the real value and $\sum_{c=1}^m |\omega_c|$ represents the sum of the absolute value of m weighting parameters for the hidden layer of the neural network.

Length of window	6
Hidden Unit	8
Learning rate	0.0006
Batch Size	8
Number of iterations	50
Optimizer	ADAM
Period of Training Data	2008/8/6 - 2014/1/9
Period of cross validation Data	2014/1/10 - 2014/8/28
Period of Back-test Data	2014/8/29 - 2016/3/9

Table 1: The parameters of LSTM used in the experiment. These hyper-parameters are the same for each training neural network of 100 stocks in total.

3.2 Second Step: Portfolio Updating

In the second step, constrained optimization is used for the portfolio updating based on the predicted price p_{t+1}^j for single asset j obtained at time t . The price relative vector

$$\hat{\mathbf{x}}_{t+1} = [\hat{x}_{t+1}^1 \hat{x}_{t+1}^2 \dots \hat{x}_{t+1}^d]^T \in \mathbb{R}_+^d$$

is a vector contains all d assets at time $t + 1$ whose j^{th} element is $\hat{x}_{t+1}^j = \frac{\hat{p}_{t+1}^j}{p_t^j}$ in which \hat{p}_{t+1}^j is the predicted close price of the j^{th} single asset at time $t + 1$ and p_t^j is the close price of the j^{th} single asset at time t , and the parameter d is the total number of assets used in the experiment. The optimization problem to obtain the optimal portfolio can be formulated as [25]

$$\mathbf{b}_{t+1} = \begin{cases} \arg \min_{\mathbf{b} \in \Delta_d} \frac{1}{2} \|\mathbf{b} - \mathbf{b}_t\|^2, & \text{such that } \mathbf{b}^T \cdot \hat{\mathbf{x}}_{t+1} \geq \varepsilon & (1) \\ \mathbf{b}_t, & \text{otherwise} & (2) \end{cases}$$

where ε denotes the minimal return we wish to obtain in next trading day. In this research, we select ε to be equal to 1.05. The vector \mathbf{b}_t is the portfolio weight vector which represents the proportion of our capital which we invest in each single asset in time t , $\Delta_d = \left\{ \mathbf{b} : b^j \geq 0, \sum_{j=1}^d b^j = 1 \right\}$ and $\|\cdot\|$ denotes the Euclidean norm. The inequality in (1) is used to decide whether we should update our portfolio vector; if its constraint is satisfied, that is, the expected return is higher than the expected minimal return ($\mathbf{b}^T \cdot \hat{\mathbf{x}} \geq \varepsilon$), then the resulting portfolio equals to previous portfolio ($\mathbf{b}_{t+1} = \mathbf{b}_t$). However, if the constraint is not satisfied, then the formulation will update a new portfolio such that the expected return is higher than the expected minimal return, while the new portfolio is not far from previous portfolio. Since the proportion which we invest in any single asset cannot be a negative number, we shall constrain our portfolio vector to be non-negative. The solution of the optimization problem is first analytically obtained without considering the

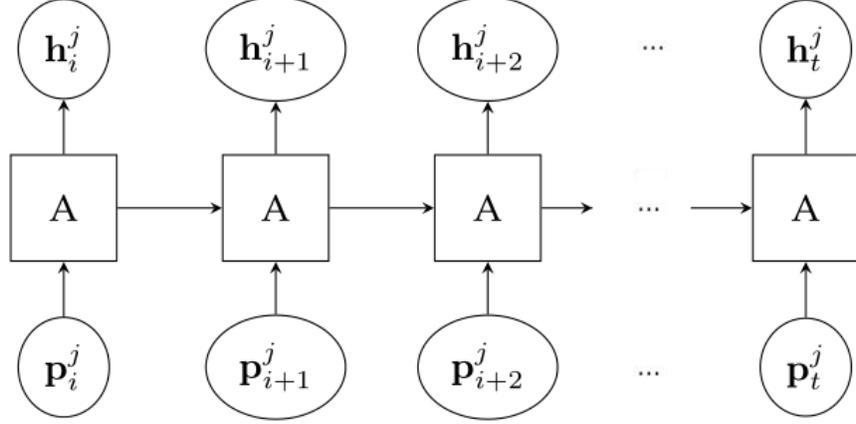


Fig. 2: The architecture of RNN used in the experiments for each single asset where $\mathbf{p}_t^j = [p_{t-i+1}^j p_{t-i+2}^j \dots p_t^j]^T$ is the input vector and \mathbf{h}_t^j is such $\mathbf{h}_t^j = [\hat{p}_{t-i+2}^j \hat{p}_{t-i+3}^j \dots \hat{p}_{t+1}^j]^T$ is the output vector. Each block A here represents identical hidden layers of RNN with LSTM block.

non-negatively constraint:

$$\mathbf{b}'_{t+1} = \mathbf{b}_t - \alpha_{t+1} (\hat{\mathbf{x}}_{t+1} - \bar{x}_{t+1} \mathbf{1})$$

where $\bar{x}_{t+1} = \frac{1}{d}(\mathbf{1} \cdot \hat{\mathbf{x}}_{t+1})$ denotes the average predicted price relative, d is the number of assets and α_{t+1} is the Lagrangian multiplier due to the inequality in (1) according to the method of solving the Lagrangian multiplier in a inequality constraint [27]. The Lagrangian multiplier can be calculated as

$$\alpha_{t+1} = \min \left\{ 0, \frac{\mathbf{b}_t^T \cdot \hat{\mathbf{x}}_{t+1} - \varepsilon}{\|\hat{\mathbf{x}}_{t+1} - \bar{x}_{t+1} \mathbf{1}\|^2} \right\}. \quad (3)$$

By combining (1) and (3), we can update our portfolio vector as follows:

$$\mathbf{b}'_{t+1} = \mathbf{b}_t - \min \left\{ 0, \frac{\mathbf{b}_t^T \cdot \hat{\mathbf{x}}_{t+1} - \varepsilon}{\|\hat{\mathbf{x}}_{t+1} - \bar{x}_{t+1} \mathbf{1}\|^2} \right\} (\hat{\mathbf{x}}_{t+1} - \bar{x}_{t+1} \mathbf{1}). \quad (4)$$

Note that it is possible that the resulting portfolio vector \mathbf{b}'_{t+1} in (4) contains negative elements since the non-negativity constraint is not considered. Thus, to ensure that the portfolio is non-negative, the the resulting portfolio vector \mathbf{b}'_{t+1} in (4) undergoes Euclidean projection to a non-negative domain [10]. The resulting portfolio \mathbf{b}_{t+1} of the projection can be proved to be the vector with the shortest Euclidean distance from vector \mathbf{b}'_{t+1} in non-negative domain [10].

4 Results and Discussion

There are 100 stocks in total from Shanghai Stock Exchange (SSE) or National Association of Securities Dealers Automated Quotations (NASDAQ) which are used in our experiments. These trading records of stocks can be downloaded in Yahoo Finance for free. The detailed information of these stocks are in the appendix. These data are divided into three parts based on time sequence, the first part is training set which is used to train a RNN-LSTM network to predict price; the second part is cross validation set which is used to tune the hyper-parameters of the neural network. The other is test set which is used in the back-test. For each stock, there are 1620 trading records used to train the neural network, 180 trading records used to hyper-parameter tuning and 360 trading records used in the back-test experiment.

4.1 Results of Back-tests

Performance Measures The following financial metrics shall be used to measure the performance of each portfolio management strategy in this paper.

1. **Final Cumulative Portfolio Wealth.** Final cumulative portfolio wealth is the portfolio value in the last time step, it can reflect how much money will be make or lost in the whole trading period. The higher the final value, the better the result of the strategy become.
2. **Positive Days.** Positive days is the proportion of trading periods which have the positive return ($\frac{p_{t+1}}{p_t} > 1$).
3. **Max Drawdown [29].** The drawdown is the measure of the decline from a historical peak in some variable (typically the cumulative profit or total open equity of a financial trading strategy). For example, if $X = (X(t), t \geq 0)$ is a random process with $X(0) = 0$, the drawdown at time T , denoted $D(T)$, is defined as:

$$D(T) = \max \left\{ 0, \max_{t \in (0, T)} X(t) - X(T) \right\}$$

The maximum drawdown (MDD) up to time T is the maximum of the Drawdown over the history of the variable. The formula is:

$$M(T) = \max_{\tau \in (0, T)} \left[\max_{t \in (0, \tau)} X(t) - X(\tau) \right]$$

It can be understood as the proportion of money one will lose in the worst situation during the trading period so the lower the max drawdown, the better the result of the strategy.

4. **Sharpe Ratio [36] [32].** In finance, the Sharpe ratio (also known as the Sharpe index, the Sharpe measure, and the reward-to-variability ratio) is a way to examine the performance of an investment by adjusting for its risk. The ratio measures the excess return (or risk premium) per unit of deviation

in an investment asset or a trading strategy, typically referred to as risk. The Sharpe ratio is defined as:

$$s_a = \frac{\mathbb{E}[R_a - R_f]}{\sigma_a} = \frac{\mathbb{E}[R_a - R_f]}{\sqrt{\text{Var}[R_a - R_f]}}$$

where R_a is the asset return, R_f is the risk-free return. $\mathbb{E}[R_a - R_f]$ is the expected value of the excess of the asset return over the benchmark return, and σ_a is the standard deviation of the asset excess return. The Sharpe ratio characterizes how well the return of an asset compensates the investor for the risk taken so the higher value, the better the result of the strategy become.

Results of Back-tests and Discussion The initial value of the back-test experiment is set to be 1000000 and the commission fee is set to be 0.05% in this experiment. The commission fee is the money which will be cost in each transaction, 0.05% commission fee means 0.05% amount of transactions should be paid as commission fee in each transaction. After running the back-test experiment based on 100 stocks, the performance of RNN-LSTM based strategy is compared to several well-known or recently published strategies based on several metrics as discussed in this section. Also, we try to compare the result of our strategy to the performance of the market. Buy and Hold strategy, a strategy which spread the total capital equally into the preselected assets and holding them without making any purchases or selling until the end, can represents the performance of the market. Also, Uniform Constant Rebalanced Portfolios, a strategy invest all assets in average in the first time step and keep the capital in each assets equal in the following trading periods [21], can represents the performance of the market as well.

Most of the strategies compared in this work were surveyed by Li and Hoi [24] including Online Moving Average Reversion Strategy (OLMAR) [23], Passive Aggressive Median Reversion Strategy (PAMR) [26], Online Newton Selection (ONS) [1], Exponentiated Gradient (EG) and Anticor [4], Kernel-Based Strategy (BK) [11], Confidence Weighted Median Reversion Strategy (CWMR) [25] except Robust Median Reversion Strategy (RMR) [15]. Table 2 shows performance of 100 stocks according to the metrics Final Value, Max Drawdown, Positive Days and Sharpe Ratio of 100 stocks in the back-test with 0.05% commission fee.

Table 2 shows back-test result of 100 stocks according to the metrics Final Value, Max Drawdown, Positive Days and Sharpe Ratio of 100 stocks in the back-test with 0.05% commission fee. The value which has bold font represents the best result of these strategies, the value which has underline represents the second better result of these strategies. The final value of Buy and Hold strategy after 360 trading periods is about 100% which shows the overall trend of the market is stable relatively. It can be seen that RNN-LSTM strategy has the highest return (148%) and CWMR strategy obtains the second best return(124%) in the back-test experiment. RNN-LSTM strategy also achieves the best result in Sharpe Ratio and achieves the second best result in Max Drawdown. RNN-LSTM strategy outperforms than RMR strategy, which is the benchmark in this

	Final value	Max Drawdown	Sharpe Ratio	Positive Days
ANTICOR	106.2414%	0.234585	0.409714	0.51676
BAH	99.9219%	0.199235	0.120446	0.511173
BK	103.0641%	0.147922	0.31184	0.541899
CRP	107.7313%	0.165124	0.96716	0.513966
CWMR	<u>124.6668%</u>	0.21815	<u>1.417289</u>	<u>0.555866</u>
EG	116.6944%	0.125781	1.237309	0.519553
OLMAR	101.4545%	0.415075	0.08132	0.485101
ONS	86.86563%	0.296049	-0.7179	0.472067
PAMR	120.174%	0.227172	1.390486	0.558659
LSTM	148.4522%	<u>0.139873</u>	1.687191	0.527933
RMR	111.3425%	0.178053	0.780039	0.5
UP	108.4251%	0.141192	0.930596	0.519553

Table 2: Performance of our strategy and other strategies in back-test of all 100 stocks with 0.05% commission fee. The performance metrics are Final Portfolio Value, Max Drawdown, Sharpe Ratio. The other strategies in the table are Buy and Hold (BAH), Uniform Constant Rebalanced portfolio (CRP) [21], Online Moving Average Reversion Strategy (OLMAR) [23], Robust Median Reversion Strategy (RMR) [15], Passive Aggressive Median Reversion Strategy (PAMR) [26], Online Newton Selection (ONS) [1], Exponentiated Gradient (EG), Kernel-Based Strategy (BK) [11], Confidence Weighted Median Reversion Strategy (CWMR) [25] and Anticor [4]. The value which has bold font represents the best result of these strategies, the value which has underline represents the second best result of these strategies.

research, in all four metrics. RNN-LSTM strategy does not achieve the best two result only in the Postive Days metric. It means the stability of RNN-LSTM strategy may not be remarkable enough.

Figure 3 gives the plot of the change of final value against time of the RNN-LSTM strategy, Buy and Hold strategy and Uniform Constant Rebalanced Portfolios. The Buy and Hold strategy and Uniform Constant Rebalanced Portfolios can represent the performance of the whole market. The x -axis represents the time t and the y -axis represents the final value of the each strategy. RNN-LSTM strategy performs better than Buy and Hold strategy and UCRP strategy in final value throughout the back-tests although it cannot perform better in every time period. It is because the profitability of our strategy depends on the accuracy of the price prediction. The price prediction can not be accurate every time and the money will be lose if the prediction is not accurate enough in that trading period. However, RNN-LSTM strategy still obtains a lower value in Max Drawdown than Buy and Hold strategy or

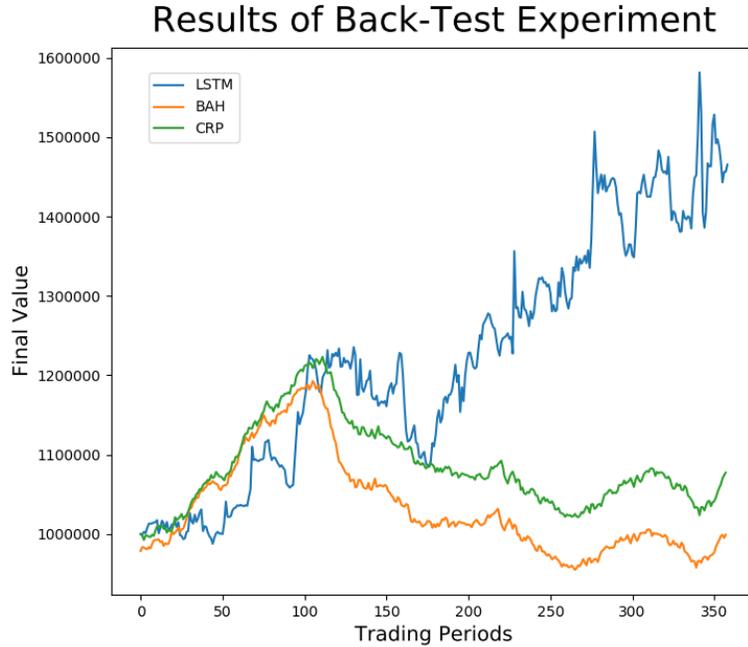


Fig. 3: Final value of the Back-test for RNN-LSTM strategy Buy, Hold strategy and Uniform Constant Rebalanced Portfolios in each trading day.

Uniform Constant Rebalanced Portfolios which means that the risk of our strategy is lower than the average performance of the market in terms of the whole 360 trading periods.

5 Conclusions and Future Work

In this paper, we propose a novel multiple period on-line portfolio selection strategy which based on the stock price prediction by Recurrent Neural Network (RNN) which has Long Short-Term Memory (LSTM) block. The profitability of our strategy surpasses most of common portfolio management strategies, as demonstrated in the paper by the average result of back-test over 100 stocks in a stock market. In the experiment, RNN-LSTM strategy outperformed RMR strategy, which is seen as benchmark in this research for all four metrics. The satisfying performance of our strategy confirm the effectiveness of prediction of RNN-LSTM. Also, our strategy may can initiate a new direction of portfolio management research which combines deep learning method to constraint optimization method.

There are several research directions we may take in the future. The first is multiple comparisons with the best (MCB) [19]. We can divide these stocks

into several small datasets and MCB method may be used in these datasets to obtain a more compelling conclusion in comparison with our strategies and other strategies. Ranking test can reveal that whether the excellent performance of our strategy is not due to chance but owed to the strategy principle. Next, we shall try to upgrade our strategy to control the volatility of the portfolio. Our strategy do not achieve the best result of Max drawdown and Positive Days in the back-test experiment which means the stability of our strategy is not remarkable enough. In additional, we can put risk-free asset such as T-bills into consideration to avoid the risk which all prices of stocks are falling down. The volatility of portfolio should also be controlled when risk-free asset is put into consideration. Lastly, we shall continue to explore the effectiveness of our strategy in high frequency trading data such as half hour trading data or 5 minutes trading data and compared our strategy with other machine learning strategy such as the deep reinforcement learning strategy [17] in the future work.

References

1. Amit Agarwal, Elad Hazan, Satyen Kale, and Robert E. Schapire. Algorithms for portfolio management based on the newton method. In *International Conference*, pages 9–16, 2006.
2. Oscar Araque, Ignacio Corcuera-Platas, J. Fernando Sanchez-Rada, and Carlos A. Iglesias. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications An International Journal*, 77(C):236–246, 2017.
3. Michael J. Best and Jaroslava Hlouskova. The efficient frontier for bounded assets. *Mathematical Methods of Operations Research*, 52(2):195–212, 2000.
4. A. Borodin, R. Elyaniv, and V. Gogan. Can we learn to beat the best stock. *Journal of Artificial Intelligence Research*, 21(1):579–594, 2011.
5. Robert G. Edgett Cooper. New product portfolio management:. *Journal of Product Innovation Management*, 16(4):333–351, 1999.
6. Marta R. Costa-jussa, Alexandre Allauzen, Loic Barrault, Kyunghun Cho, and Holger Schwenk. Introduction to the special issue on deep learning approaches for machine translation. *Computer Speech & Language*, 46, 2017.
7. T. M. Cover. Universal data compression and portfolio selection. In *Symposium on Foundations of Computer Science*, page 534, 1996.
8. Puja Das and Arindam Banerjee. Meta optimization and its application to portfolio selection. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1163–1171, 2011.
9. Davelos, L Anita, Kinkel, L Linda, Samac, and A Deborah. *Modern investment theory*. Prentice Hall,, 2001.
10. John C. Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the. In *International Conference*, pages 272–279, 2008.
11. Gyorf, Lugosi, Gabor, Udina, and Frederic. Nonparametric kernel-based sequential investment strategies. *Mathematical Finance*, 16(2):337–357, 2010.
12. Laszlo Gyorf, Gabor Lugosi, and Frederic Udina. Nonparametric kernel-based sequential investment strategies. *Mathematical Finance*, 16(2):337C357, 2006.
13. N. Ozsoylev Han and Shino Takayama. Price, trade size, and information revelation in multi-period securities markets. *Journal of Financial Markets*, 13(1):49–76, 2010.

14. J. B. Heaton, N. G. Polson, and J. H. Witte. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1), 2017.
15. Dingjiang Huang, Junlong Zhou, Bin Li, Steven C. H. Hoi, and Shuigeng Zhou. Robust median reversion strategy for on-line portfolio selection. In *International Joint Conference on Artificial Intelligence*, volume 28, pages 2006–2012, 2013.
16. Hsien De Huang, Chia Mu Yu, and Hung Yu Kao. Data-driven and deep learning methodology for deceptive advertising and phone scams detection. 2017.
17. Zhengyao Jiang and Jinjun Liang. Cryptocurrency portfolio management with deep reinforcement learning. *Arxiv*, 2017.
18. Kelly, L. J., and Jr. A new interpretation of information rate. *Bell System Technology Journal*, 2(3):917–926, 1956.
19. Alex J. Koning, Philip Hans Franses, Michele Hibon, and H. O. Stekler. The m3 competition: Statistical tests of the results. *International Journal of Forecasting*, 21(3):397–409, 2005.
20. John R. Koza. Genetic programming as a means for programming computers by natural selection. *Statistics & Computing*, 4(2):87–112, 1994.
21. Suleyman S. Kozat and Andrew C. Singer. Universal constant rebalanced portfolios with switching. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages III–1129 – III–1132, 2007.
22. Jan Lansky. Analysis of cryptocurrencies price development. *Acta Informatica Pragensia*, 2016.
23. Bin Li and Steven C. H. Hoi. On-line portfolio selection with moving average reversion. *Papers*, page 173C190, 2012.
24. Bin Li and Steven C. H. Hoi. Online portfolio selection: A survey. *Acm Computing Surveys*, 46(3):1–36, 2012.
25. Bin Li, Steven C. H. Hoi, Peilin Zhao, and Vivekanand Gopalkrishnan. Confidence weighted mean reversion strategy for online portfolio selection. *Acm Transactions on Knowledge Discovery from Data*, 7(1):1–38, 2013.
26. Bin Li, Peilin Zhao, Steven C. Hoi, and Vivekanand Gopalkrishnan. Pamr: Passive aggressive mean reversion strategy for portfolio selection. *Machine Learning*, 87(2):221–258, 2012.
27. Mengmou Li. Generalized lagrange multiplier method and kkt conditions with application to distributed optimization. *IEEE Transactions on Circuits & Systems II Express Briefs*, PP(99):1–1, 2018.
28. Shanghong Li, Jiayu Zhang, and Yan Qi. Predicting s & p500 index using artificial neural network. In *International Conference on Computer and Computing Technologies in Agriculture*, pages 173–189, 2015.
29. Zhiyong Li. Maxdrawdown: Stata module to calculate the maximum drawdown of a stock, fund or other financial product. *Statistical Software Components*, 2016.
30. Harry Markowitz. Portfolio selection. *Journal of Finance*, 7(1):77–91, 1952.
31. Phayung Meesad and Srikhacha Tong. Stock price time series prediction using neuro-fuzzy with support vector guideline system. In *Acis International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/distributed Computing*, pages 422–427, 2008.
32. Christoph Memmel. Performance hypothesis testing with the sharpe ratio. *Social Science Electronic Publishing*, 27(3):299–306, 2003.
33. Vahid Moosavi. Urban morphology meets deep learning: Exploring urban forms in one million cities, town and villages across the planet. 2017.
34. Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1, 2015.

35. Foundations Trends In Signal Processing. Deep learning: Methods and applications. *Foundations & Trends in Signal Processing*, 7(3):197–387, 2014.
36. William F. Sharpe. The sharpe ratio. *Journal of Portfolio Management*, 21(1):49–58, 1994.
37. V. Vovk and C. Watkins. Universal portfolio selection. In *Eleventh Conference on Computational Learning Theory*, pages 12–23, 1998.
38. Jiajun Wu, Yinan Yu, Chang Huang, and Kai Yu. Deep multiple instance learning for image classification and auto-annotation. In *Computer Vision and Pattern Recognition*, pages 3460–3469, 2015.
39. Yoshiyuki Yabuuchi and Junzo Watada. *Building Fuzzy Autocorrelation Model and Its Application to the Analysis of Stock Price Time-Series Data*. Springer Berlin Heidelberg, 2013.
40. Dong Yu and Li Deng. *Automatic Speech Recognition: A Deep Learning Approach*. Springer, 2014.

6 Appendix

The appendix shows the detailed stock code or logogram of stocks which are used in the back-test.

1. **SSE**. There are 50 stocks from SSE, their stock codes are 600000, 600001, 600004, 600015, 600028, 600031, 600060, 600249, 600546, 600848, 600104, 600109, 600119, 600485, 600893, 601198, 601377, 601800, 601985, 601998, 600016, 600036, 600111, 600519, 600585, 601006, 601088, 601318, 601328, 601601, 600048, 600050, 600089, 600104, 600282, 600348, 600547, 601857, 601899, 601939, 600019, 600362, 600383, 600489, 600518, 600887, 601600, 601628, 601788, 601766.
2. **NASDAQ**. There are 50 stocks from NASDAQ, their logograms are CAT, GE, GS, F, CAH, CCL, CCE, DIS, DUK, HAS, AVP, BXP, D, DFS, DVA, IFF, MAS, MO, POM, USB, ALL, BDX, C, CNP, EFX, MSI, NWL, S, TGNA, ZMH, BBT, BBY, BIG, BILL, COP, DRI, GWW, VNO, XEL, XL, AEP, AIV, AN, BMY, CHRW, CL, DNR, HUM, JPM, MTB.

Fraud Detection Using Explainable Machine Learning Algorithms

Luciano C M Andrade^[0000-0002-0420-2078] André C P L F
Carvalho^[0000-0002-4765-6459]

Institute of Mathematics and Computer Science
University of São Paulo
São Carlos, SP, Brazil
{lucianocarli, andre}@icmc.usp.br

Abstract. Artificial Intelligence is increasingly being used by credit and finance companies and has led to most financial transactions being carried out through electronic systems. As a consequence, there has been an increase in fraud transactions. Additionally, fraudsters continuously look for new approaches to commit illegal actions. Frauds usually result in high economic costs. Improvements in fraud detection systems play a key role in reducing losses and improve the reliability of electronic systems. In order to obtain a more effective fraud detection, many researchers have attempted to develop sophisticated anti-fraud approaches by incorporating artificial intelligence, mainly machine learning, techniques. Machine techniques induce models able to distinguish between legitimate and illegitimate transactions, supporting fraud detection handling large volumes of highly complex data. A challenge arising from applying machine-learning techniques to fraud-related datasets is the high imbalance in the data from the different classes. In these datasets, which are usually binary, the class of interest, fraud, usually has much less examples than the non-fraud class. Machine label algorithms based on ensembles have been successfully applied to imbalanced datasets. Additionally, there is a movement towards increasing transparency and interpretability of artificial intelligence, called explainable artificial intelligence. This paper investigates using machine learning techniques able to induce interpretable models for fraud detection tasks. Considering this, algorithms with increasing complexity are applied to three fraud-related imbalanced datasets, associated with credit card transaction fraud, click fraud and retail fraud. Finally, to improve the predictive performance, the parameters of the most complex technique are optimized using two different optimization algorithms. The experimental results show the gains obtained by this approach.

Keywords: Explainable Artificial Intelligence · Fraud detection · Multi-objective optimization.

1 Introduction

Fraud can be defined as any deceitful activity that leads to obtaining unlawful advantage by one part over another or causes unlawful losses [14]. Typically, the

fraudster uses false or misleading representations to disguise his/her activities as long as possible in order to maximize the effects of his/her fraudulent behavior [2]. Adverse effects related to fraud have an impact on all business enterprises [18]. The development of new technologies has also provided further ways in which criminals may commit fraud. Therefore, fraud detection has become an important issue in identifying frauds as soon as they have been committed [10]. It is increasing substantially with the expansion of global communication technologies, resulting in considerable business losses [1].

The Internet Crime Complaint Center (IC3), which is a joint operation between the Federal Bureau of Investigation (FBI) and the National White Collar Crime Center (NW3C), summarizes the number of complaints related to internet crimes received and the corresponding dollar losses [2]. Figure 1 shows an increase from 262,813 complaints in 2013 to 301,580 in 2017, with a corresponding loss from 781.8 million dollars to 1,418.7 million dollars. In 2016, the loss was 1,450.7 million dollars. Therefore, the costs incurred by fraud transferred to the society can be observed causing increased inconvenience to customers, unnecessarily high prices of goods and services, and criminal activities funded by fraudulent gains. Despite significant advances in fraud detection technologies, fraud losses continue to be a significant problem in sectors such as telecommunications, banking and finance, insurance, e-commerce, and many others [18].

Fraud detection is a constantly evolving discipline. As soon as a new detection method is designed, criminals adapt their strategies and try new ones. As new fraudsters are also constantly taking action, they may be barred by fraud detection methods that have been successful in the past. This means that previous detection methods need to be constantly applied, as well as the latest developments [21].

The first layer of protection systems is to prevent fraud. Fraud prevention is the task of preventing fraud from happening in the first place. This can be done by improving technologies and projects. However, the first tier is not always

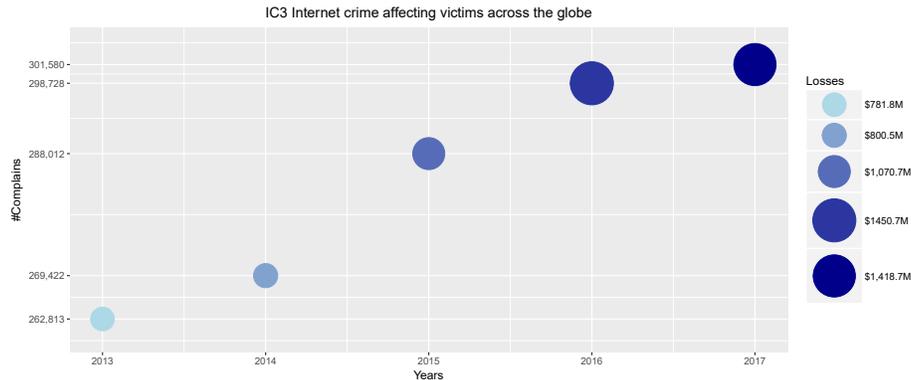


Fig. 1. Number and costs of internet crimes across the globe.

successful and is occasionally penetrated by fraudsters. Fraud detection is the second layer of defense, and it is responsible for detecting and recognising fraudulent activities as they enter into systems and for reporting them to a system manager [2].

It is impossible to be absolutely sure about the intent behind a transaction. However, fraud detection is one of the layers of general fraud control. It automates and helps reduce the manual parts of a verification process to identify a fraudulent transaction. This area has become one of the most established industrial and governmental data mining applications [13]. Given the reality, the best cost effective option is to discover possible evidence of fraud from the available data using automatic algorithms [23].

It is important to mention that there are commercial compromises that have to be reached between the cost of detecting a fraud and the savings to be made by detecting it. Additionally, the adverse publicity of a campaign of fraud detection can be complicated. Revealing that an enterprise is a significant target of fraud, even though many frauds have been detected, does not inspire confidence, and accusing an innocent customer who may be suspected of fraud is detrimental to good customer relations. Thus, a proper fraud detection technique must not only maximize correct predictions, but also maintain incorrect predictions at an acceptable level [19].

Once the costs and challenges that may be caused by these illicit activities are presented, it becomes evident that the development of efficient fraud detection algorithms is essential to reduce these costs. Fraud detection algorithms can rely on advanced Machine Learning (ML) techniques to help fraud investigators. Designing fraud detection algorithms is, however, particularly challenging due to non-stationary data distributions, highly imbalanced class distributions, and continuous transaction flows [15].

Therefore, the main contributions of this paper are: (i) investigating the use of ML techniques for the fraud detection task; (ii) investigating the use of other areas of successful techniques [24] to deal with fraud detection imbalanced datasets; (iii) comparing the use of optimization methods (MOPSO and NSGA-II) determining the best Cascade Random Forest (CRF) parameters for fraud detection; and (iv) a comprehensive criterion to choose the best CRF parameters in case of fraud detection;

The paper is organised as follows: Section 2 presents some aspects of the datasets used in this research. Section 3 presents the methodology adopted. Section 4 presents and discusses the results obtained, and Section 5 draws the main conclusions and points out future work directions.

2 Aspects of Fraud Datasets

In order to evaluate and compare the performance of the ML techniques, three fraud datasets, covering different fraud problems, were used:

- Credit Card Fraud dataset [20];

- Click Fraud Dataset [8];
- Retail Fraud Dataset [22].

Although these datasets have been used in previous publications, the authors did not find any previous work evaluating the performance of different levels of explainable ML [6] using these datasets. Next, the main aspects of each dataset are presented.

2.1 Credit Card Fraud Dataset

This dataset was one of those used in the UCSD-FICO Data mining contest 2009, organised by FICO (Fair Isaac Corporation), a provider of analytics and decision management technology, and the University of California, San Diego (UCSD). It is a real dataset of e-commerce transactions [20].

The competition provided two files, one labeled (for training) and one unlabeled (to evaluate the submitted solutions). For the experiments in this paper, we used only the labeled training dataset. It contains 94682 transactions- from 9811 customers in a period of 2002 days. This dataset has 19 predictive attributes and one target attribute [20], as follows:

- amount - the total amount of the purchase;
- hour1 - the hour of the day of the purchase;
- state1 - the state of the customer;
- zip1 - the zip code of the customer;
- field 1,...,5 - five anonymized fields;
- indicator 1,2 - two anonymized indicators;
- flag - 1,...,4 - four anonymized flags;
- Target - the target to be predicted;

The attributes total and amount, as well as hour1 and hour2 are found to be the same for each transaction, thus we removed the total and hour2. Similarly, state1 and zip1 are also the same information, thus we removed state1. All other attributes are anonymized and consequently, it was decided to keep them as they are. Therefore, the final credit card (CCard) dataset contains 16 attributes—amount, hour1, zip1, field1, domain1, field2, flag1, field3, field4, field5, indicator1, indicator2, flag2, flag3, flag4, flag5 and Target [20].

2.2 Click Fraud Dataset

The click fraud dataset used in the experiments was obtained from the largest Chinese independent platform of big data services. It covers 70% of active mobile devices nationwide. The dataset covers approximately 200 million clicks collected during 4 days [8].

Each row in the click fraud dataset contains a click record, with the following attributes.

- ip - ip address of the click;

- app - app id for marketing;
- device - device type id of user mobile phone (e.g., iphone 6 plus, iphone 7, huawei mate 7, etc.);
- os - os version id of user mobile phone;
- channel - channel id of mobile ad publisher;
- click_time - timestamp of click (UTC);
- attributed_time - if the user downloads the app after clicking on an ad, this is the time the app was downloaded;
- is_attributed - the target to be predicted, indicating the app has been downloaded.

Table 1 summarises the main information found in the click fraud dataset. The total amount of transactions, the amount of frauds and genuine transactions, and the significant information of fraud transaction ratio can be observed. The click fraud dataset transaction ratio is 1:398. This information demonstrates the highly imbalanced characteristics of the data. However, it is important to note that while other fraud detection problems often contain a much larger number of genuine transactions than fraud transactions, when analysing click fraud detection problems, this characteristic is reversed. This particular aspect was identified in the click fraud dataset analysed [8].

2.3 Retail Fraud Dataset

The retail fraud data used is available in [22] and it has been anonymized. Each of the 401,146 rows of the data table includes information on one report by some salesman. This information includes an ID, a product ID, and the quantity and total value reported by the salesman. This data has already gone through some previous analysis. The result of this analysis is shown in Insp attributes, which has the result of the inspection of some transactions by the company [22].

The retail fraud dataset used has the following attributes:

- ID - a factor with the ID of the salesman.
- Prod - a factor indicating the ID of the sold product.
- Quant - the number of reported sold units of the product.
- Val - the reported total monetary value of the sale.
- Insp - three possible values: “ok” if the transaction was inspected and considered valid by the company, “fraud” if the transaction was found to be fraudulent, and “unknown”.

In order to evaluate if a transaction is valid or fraud, the unknown reads were removed to create the final retail fraud dataset used, resulting in a dataset with 15,546 labeled transactions. This settlement were recommended by [22].

2.4 General Aspects of All Datasets

Table 1 presents summarised information about all the datasets used in this research. One important aspect to verify is the high imbalance ratio of all datasets.

The CRF [24] technique was developed to address this characteristic and this research optimizes the CRF [24] parameters to obtain better classification results regarding the majority and minority classes (genuine and fraud).

Table 1. Imbalanced Classification Datasets Used in the Experiments

Name	Number of Examples			Imb. Ratio	#Features
	Total	Genuine	Fraud		
CCard	94,682	97.78%	2.22%	44:1	21
Click	3,698,077	0.25%	99.75%	1:398	8
Retail	15,546	92.28%	7.72%	12:1	5

3 Methods

Genetic algorithms (GA) techniques were recently combined with k-means to optimize credit card fraud detection. This hybrid technique enhanced the classification performance of the minority instances of credit card fraud in the imbalanced dataset [3]. GA were also combined with fuzzy logic to optimize fuzzy rules and determine the most dangerous seller in auction fraud detection [25]. It is important to highlight that fuzzy logic is a predictive model that humans can interpret. Therefore, optimization methods and explainable models have provided improvements in the subject of fraud detection.

Multi-objective optimization has been applied in many fields of science, including engineering, economics and logistics, where trade-offs between two or more conflicting objectives is present. Meta-heuristics find a set of solutions in a search space, which cannot be entirely demonstrated using some forms of stochastic optimisation, where the solution found is conditional on the set of random variables generated. Meta-heuristics search over a large set of possible solutions and find excellent solutions with a minor number of attempts than optimisation algorithms, iterative methods, or simple heuristics. Therefore, meta-heuristics are useful methods for optimisation problems. NSGA-II [5] and MOPSO [17] are the well-known meta-heuristic optimisation methods for non-trivial multi-objective optimisation problems in engineering, economics and logistics. Thus, the objective functions are contradictory, and there are a number of Pareto optimal solutions. In case none of the objective functions can be enhanced in value without the deterioration of other objective values, the solution is called non-dominated Pareto optimal. All Pareto optimal solutions are taken into account equally. The objective is to find a set of Pareto optimal solutions, and to find a single solution that satisfies the criterion of a decision maker (DM) [7].

3.1 3.1 Multi-objective Particle Swarm Optimisation

PSO is a population based method for optimisation [9]. The population of the potential solution is called a swarm. It is formed by a set of particles. The particles in the swarm search their best solution based on their own experience and the other particles of the same swarm. PSO became the most popular swarm intelligence technique shortly after being introduced, but due to its limitation of optimisation for only one single objective, a new concept Multi-Objective PSO (MOPSO) became very popular [17], by which optimisation can be performed for more than one conflicting objective simultaneously. Instead of a single solution, a set of solutions is determined, which is also called a Pareto optimal set [7].

3.2 Non-dominated Sorting Genetic Algorithm-II

NSGA-II [5] is a useful algorithm, which has an improved mechanism based upon the crowding distance and performs constraints using an adapted apparatus of dominance with no penalty functions. At first, a zero level is allocated to all non-dominated individuals. During the elimination of the individuals from the population, the lately non-dominated solutions are allocated level one. This procedure is performed until the time in which all the solutions have been allocated a non-domination level. The NSGA-II uses a binary tournament selection based on the lesser rank and greater crowding distance. Then an offspring is generated from the selected population using crossover and mutation operators. Finally, the present offsprings and population are sorted another time, dependent on the non-domination and just the population size of the best individuals are selected [7].

3.3 Cascade Random Forests

The CRF proposed by [24] minimises the negative effect of the data imbalance by ensembling multiple random forests (RF) [4] in a cascade fashion, where each of them is trained with a balanced training subset. Thus, each trained RF will not be biased toward the majority class and the connected RF [4] are intrinsically an ensemble method, which helps to reduce the negative effects of information loss caused by random sub-sampling [7].

3.4 Cascade Random Forest Parameter Optimisation

The CRF proposed algorithm was introduced to deal with the imbalance of the Protein-protein interaction (PPI) problem [24]. The algorithm determines the 4 best parameter values according to the results obtained during the analyses.

The parameters and their best values for PPI are listed as follows:

- L - the number of layers of RF in the trained CRF equal to the imbalance ratio of the datasets;

- TPR - Prescribed true positive rate for selecting the threshold of each trained RF. The value indicated is 0.95;
- nTree - number of trees in each RF equal to 15;
- minLeaf - minimum number of required examples to form a leaf equal to 400;

However, two questions that arise are:

- Can CRF be used in other research areas with imbalanced data?
- What should the values be of the CRF parameters for imbalanced datasets from another area of study such as fraud detection?

Therefore, this work implements a cost function based on the CRF algorithm [24], with four parameters (L, TPR, nTree and minLeaf) and compares the MOPSO and NSGA-II performance. It also determines Pareto optimal sets in case of tree fraud datasets (credit card, click and retail) to answer these questions. Based on the Pareto optimal sets, the decision maker can choose the best parameters, as presented in the Results section.

4 Experimental Results

There is a strong movement in ML to have predictive models that humans can easily interpret [6]. Due to the importance of transparency in ML tasks, we used ML algorithms that induce explainable models. We used transparent ML algorithms with three levels of transparency: the C4.5 decision tree induction algorithm [16], Random Forests (RF) [4], which create an ensemble of decision trees, and Cascade Random Forests (CRF), which induce a cascade of RFs [24].

The mentioned contributions of the paper are: (i) investigating the use of ML techniques for the fraud detection task; (ii) investigating the use of successful techniques in other areas [24] dealing with fraud detection imbalanced datasets; (iii) comparing the use of optimisation methods (MOPSO and NSGA-II) determining the best Cascade Random Forest (CRF) parameters for fraud detection; and (iv) a comprehensive criterion to choose the best CRF parameters in case of fraud detection;

The ML algorithm used in this work model adopted for the work investigation is the Random Forest [4]. This model was chosen by [24] to be assembled in the CRF. Therefore, the work investigates the use of this ML tool for the fraud detection task.

Originally CRF was successfully used to deal with the imbalance PPI problem [24]. In the case of this work, the proposed method was used for fraud detection imbalanced datasets. Table 2 presents the classification predictive performance for C4.5, RF and the standard CRF (CRF with [24] default parameter values: L=imbalance ratio, TPR=0.95, nTree=15 and minLeaf=400) for fraud dataset majority (Pos Pred Value - PPV) and minority (Neg Pred Value - NPV) classes. The good CRF performance of this task can be clearly observed. However, it is

not possible to ensure if the CRF parameter values used in the PPI problem are the most appropriate to be used in case of fraud detection.

The metrics chosen for comparison are PPV (ratio of the correct classification of the majority class) and NPV (ratio of the correct classification of the minority class) obtained by the confusion matrix of the classification results. In case of imbalanced datasets, these metrics are a trade-off (the increase in one value results in the decrease in the other value and vice-versa). Based on this, MOPSO and NSGA-II multi-objective optimization techniques were chosen to determine the Pareto optimal solutions for this problem.

Table 2. C4.5, RF and CRF default (10-fold cross-validation) (PPV/NPV).

Datasets	C4.5		RF-default		CRF-default	
	PPV	NPV	PPV	NPV	PPV	NPV
C4.5	0.81 ± 0.01	0.82 ± 0.01	0.84 ± 0.01	0.82 ± 0.01	0.88 ± 0.01	0.88 ± 0.01
Click	0.91 ± 0.01	0.81 ± 0.01	0.91 ± 0.01	0.82 ± 0.01	0.93 ± 0.02	0.82 ± 0.02
Retail	0.89 ± 0.01	0.82 ± 0.02	0.90 ± 0.02	0.85 ± 0.02	0.98 ± 0.01	0.89 ± 0.01

Table 3 presents the model performance based on Balanced Accuracy (BAcc) and Geometric mean (Gmean) metrics [12]. One can infer that for all fraud datasets, there is a performance increase from C4.5 to CRF-Default. Additionally, based on the performance results presented in Table 5, it can be concluded that NSGA-II and MOPSO determined the most appropriate parameters and achieved better results when compared with the benchmark models.

Table 3. C4.5, RF and CRF default (10-fold cross-validation) (BAcc/Gmean).

Datasets	C4.5		RF-default		CRF-default	
	BAcc	GMean	BAcc	GMean	BAcc	GMean
C4.5	0.83 ± 0.01	0.83 ± 0.01	0.86 ± 0.01	0.86 ± 0.01	0.89 ± 0.01	0.89 ± 0.01
Click	0.91 ± 0.01	0.91 ± 0.01	0.91 ± 0.01	0.91 ± 0.01	0.88 ± 0.02	0.87 ± 0.03
Retail	0.86 ± 0.02	0.85 ± 0.02	0.90 ± 0.01	0.90 ± 0.01	0.94 ± 0.01	0.94 ± 0.01

Tables 4 and 5 present the average and standard deviation calculated over the Pareto optimal solutions obtained from NSGA-II and MOPSO optimization techniques. The superiority of MOPSO in this task can be observed for all the fraud datasets used regarding all the metrics calculated (PPV, NPV, BAcc and Gmean). Therefore, the work contributes with a comparison between MOPSO and NSGA-II for CRF optimisation to the task of fraud detection classification.

The MOPSO superiority can also be observed by the graphics of the Pareto optimal solutions plotted in Figures 2, 3 and 4. The MOPSO (black curve) includes higher values for both PPV and NPV compared to the NSGA-II (blue

Table 4. Pareto optimal solutions comparison (PPV/NPV).

Datasets	CRF-NSGA-II		CRF-MOPSO	
	PPV	NPV	PPV	NPV
CCard	0.88 ± 0.03	0.85 ± 0.05	0.89 ± 0.04	0.88 ± 0.05
Click	0.93 ± 0.05	0.80 ± 0.06	0.96 ± 0.04	0.82 ± 0.06
Retail	0.97 ± 0.06	0.86 ± 0.01	0.99 ± 0.01	0.87 ± 0.04

Table 5. Pareto optimal solutions comparison (BAcc/Gmean).

Datasets	CRF-NSGA-II		CRF-MOPSO	
	BAcc	Gmean	BAcc	Gmean
CCard	0.87 ± 0.04	0.87 ± 0.04	0.89 ± 0.04	0.89 ± 0.04
Click	0.87 ± 0.05	0.86 ± 0.06	0.89 ± 0.05	0.88 ± 0.05
Retail	0.92 ± 0.03	0.92 ± 0.02	0.93 ± 0.02	0.93 ± 0.03

curve). The graphs also present the standard CRF classification value (red circle). The lower performance can be seen in comparison to one possible MOPSO decision maker choice (green triangle). The decision maker could choose any MOPSO Pareto optimal solution. Therefore, one question still remains, which is about the comprehensive criterion to choose the best CRF parameters in case of fraud detection. Any MOPSO Pareto optimal solution can be chosen. However, it is important to remember that there are commercial compromises that have to be reached between detecting a fraud and accusing an innocent customer who may be suspected of fraud. Therefore, a good balance about the PPV and NPV are imperative to determine the chosen Pareto optimal solution (green triangle).

Table 6 shows the values of PPV, NPV, BAcc and Gmean in case of CRF with MOPSO (the best model). The values were chosen based on the Pareto optimal solution curves (green triangles) in all presented graphs (Figures 2, 3 and 4) and the parameter values associated with them. These results can be compared with Table 2 and the superior performance of the MOPSO technique can be seen in comparison with C4.5, RF, standard CRF and NSGA-II.

Table 6. MOPSO best CRF parameters.

Datasets	CRF with MOPSO							
	L	TPR	nTree	minLeaf	PPV	NPV	BAcc	Gmean
CCard	10	0.92	19	418	0.90	0.90	0.90	0.90
Click	9	0.90	14	403	0.97	0.87	0.92	0.92
Retail	11	0.93	17	428	0.99	0.91	0.95	0.95

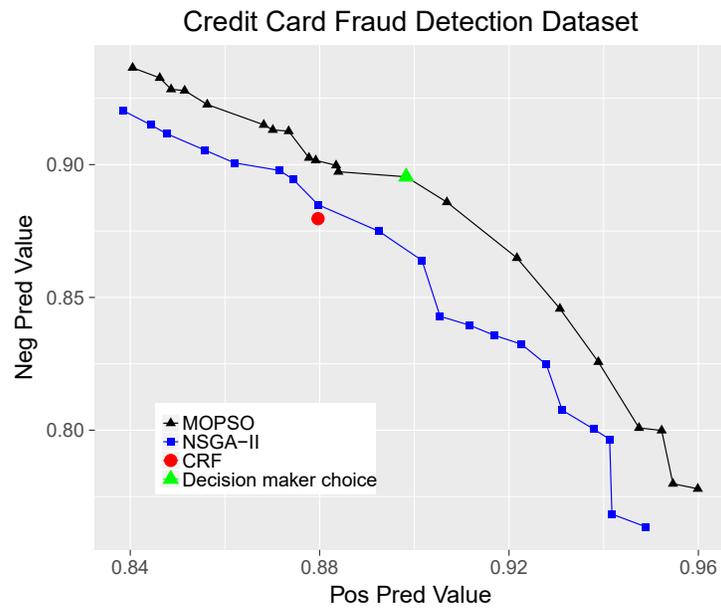


Fig. 2. Pareto fronts obtained for credit card fraud dataset.

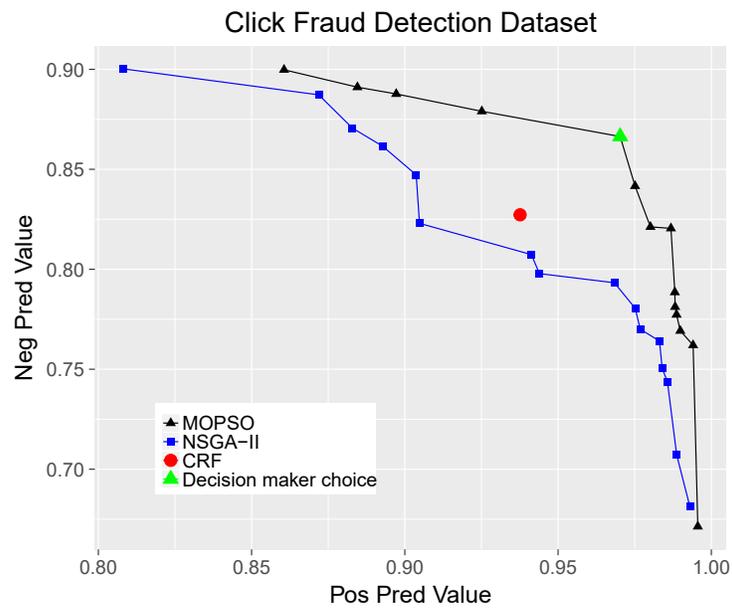


Fig. 3. Pareto fronts obtained for click fraud dataset.

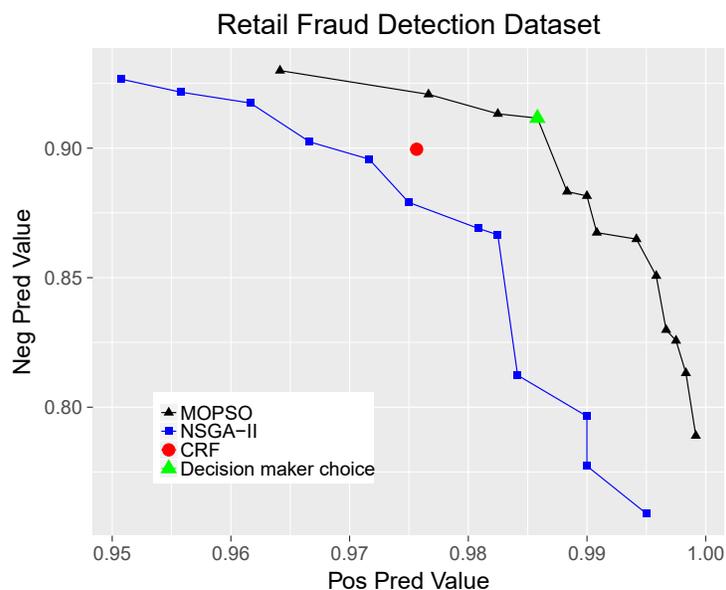


Fig. 4. Pareto fronts obtained for retail fraud dataset.

It is important to mention that the CRF parameters (number of layers of CRF (L), prescribed true positive rate (TPR), number of trees in each RF (nTree), and minimum number of required examples to form a leaf (minLeaf) of each RF) determine how well the CRF can be adapted for data classification. Therefore, the superior performance obtained by the CRF with MOPSO is the result of an efficient optimisation mechanism to find the proper CRF parameters for fraud detection in the case of the three datasets under analysis.

Additionally, Tables 7, 8 and 9 present the order in which the dataset attributes are the most important for the classification process of all the models (C4.5, RF, CRF, NSGA-II and MOPSO) analysed. To determine the importance of the dataset attributes in the case of the decision trees, the ROC curve variable importance was used [11] and, in the case of the random forests the Mean Decrease in Accuracy method was adopted [11].

With an importance average (ImpAgv) of 291.6, the most important attribute in case of the credit card dataset is “hour1”, that is, the hour of the day of the purchase. It can be concluded that the hour of the day when the credit card purchase is made can determine whether the transaction will be legitimate or fraudulent.

In the click fraud dataset, the most important attribute (ImpAgv equal to 143.5) is the “app”, that is, the app id for marketing. This means that the application being used to access the online marketing campaigns determine whether the user will buy the product or just click on the campaign banners without purchasing anything.

Table 7. Ranking of the most important attributes of the Credit Card Dataset.

Attribute	Algorithm					ImpAvg
	C4.5	RF	CRF	CRF-NSGA-II	CRF-MOPSO	
hour1	1	1	1	1	1	291.6
field1	2	2	2	2	2	221.6
field3	3	3	3	3	3	143.4
zip1	5	4	4	4	4	112.8
field4	4	5	5	5	5	102.1

Table 8. Ranking of the most important attributes of the Click Dataset.

Attribute	Algorithm					ImpAvg
	C4.5	RF	CRF	CRF-NSGA-II	CRF-MOPSO	
app	1	1	1	1	1	143.5
ip	2	2	3	3	2	75.0
channel	3	3	3	2	3	67.6
device	4	4	4	4	4	21.8
hour	5	5	5	5	5	6.9

Table 9. Ranking of the most important attributes of the Retail Dataset.

Attribute	Algorithm					ImpAvg
	C4.5	RF	CRF	CRF-NSGA-II	CRF-MOPSO	
Quant	3	1	1	1	1	246.0
Val	4	2	2	2	2	202.0
Prod	1	3	3	3	3	131.4
ID	2	4	4	4	4	128.0

In the retail fraud dataset, the random forests’ most important attribute (ImpAvg equal to 246.0) is the “quant” attribute, that is, the number of reported sold units of the product. In other words, the quantity of products purchased can be decisive for identifying which purchases are legitimate or fraudulent.

5 Conclusions

This paper examined all the information about the comparison between C4.5, RF, standard CRF and CRF optimised by the multi-objective techniques NSGA-II and MOPSO, for the task of classifying imbalanced fraud datasets. The results demonstrated the superiority of MOPSO in comparison with NSGA-II for the task of CRF parameter optimization.

Originally, the CRF was designed to deal with the imbalance in Protein-protein interactions (PPI) problem datasets and proper parameters were deter-

mined for this task. However, one issue that arises in the case of CRF applications in other areas where imbalance datasets are a challenge is which are the most appropriate parameters to be used.

Therefore, with this work one possible Pareto optimal solution could be chosen by the decision maker based on the importance of correct classification in both fraud and legitimate transactions. Thus, the CRF most appropriate parameters could be determined and tabulated for all analysed fraud datasets.

Additionally, it is important to have explainable models, mainly when the decision taken using them affects people. We described the models induced by presenting what they considered as the most important attributes in the datasets. This provides extra information about the attributes that were considered by all the models to determine the legitimate and fraudulent transactions.

Once all the information about the experiments developed in this work has been detailed, it can be concluded that methods used in certain problems such as PPI can be analysed and properly adapted to solve problems in other areas such as fraud detection.

As future work we plan to describe the explanation provided by the models induced by different algorithms and compare them. We also intend to increase the number of fraud-related datasets used.

Acknowledgments

The authors would like to acknowledge the financial support of the Brazilian agencies FAPESP (process numbers 2012/22608-8, 2016/18615-0 and 2017/02859-0), CAPES and CNPq.

References

1. Allan, T., Zhan, J.: Towards fraud detection methodologies. In: 2010 5th International Conference on Future Information Technology. pp. 1–6 (May 2010)
2. Behdad, M., Barone, L., Bennamoun, M., French, T.: Nature-inspired techniques in the context of fraud detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **42**(6), 1273–1290 (Nov 2012)
3. Benchaji, I., Douzi, S., El Ouahidi, B.: Using genetic algorithm to improve classification of imbalanced datasets for credit card fraud detection. In: Khoukhi, F., Bahaj, M., Ezziyani, M. (eds.) *Smart Data and Computational Intelligence*. pp. 220–229. Springer International Publishing, Cham (2019)
4. Breiman, L.: Random forests. *Machine Learning* **45**(1), 5–32 (2001)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (April 2002)
6. Dosilovic, F.K., Brcic, M., Hlupic, N.: Explainable artificial intelligence: A survey. In: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). pp. 0210–0215 (May 2018)
7. Hojjati, A., Monadi, M., Faridhosseini, A., Mohammadi, M.: Application and comparison of nsga-ii and mopso in multi-objective optimization of water resources systems. *Journal of Hydrology and Hydromechanics* **66**(3), 323 – 329 (2018)

8. Kaggle: Talkingdata adtracking fraud detection challenge: [online] (2018), <https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection>
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks. vol. 4, pp. 1942–1948 vol.4 (Nov 1995)
10. Kou, Y., Lu, C.T., Sirwongwattana, S., Huang, Y.P.: Survey of fraud detection techniques. In: IEEE International Conference on Networking, Sensing and Control, 2004. vol. 2, pp. 749–754 Vol.2 (March 2004)
11. Kuhn, M.: Variable importance using the "caret" package: [online] (2007), Available in <http://ftp.uni-bayreuth.de/math/statlib/R/CRAN/doc/vignettes/caret/caretVarImp.pdf>
12. Lopez, V., Fernandez, A., Garcia, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences* **250**, 113 – 141 (2013)
13. Ngai, E., Hu, Y., Wong, Y., Chen, Y., Sun, X.: The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems* **50**(3), 559 – 569 (2011)
14. Pejic-Bach, M.: Invited paper: Profiling intelligent systems applications in fraud detection and prevention: Survey of research articles. 2010 International Conference on Intelligent Systems, Modelling and Simulation pp. 80–85 (2010)
15. Pozzolo, A.D., Caelen, O., Borgne, Y.A.L., Waterschoot, S., Bontempi, G.: Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications* **41**(10), 4915 – 4928 (2014)
16. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
17. Raquel, C.R., Naval, Jr., P.C.: An effective use of crowding distance in multiobjective particle swarm optimization. In: Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation. pp. 257–264. GECCO '05, ACM, New York, NY, USA (2005)
18. Rebahi, Y., Nassar, M., Magedanz, T., Festor, O.: A survey on fraud and service misuse in voice over ip (voip) networks. *Information Security Technical Report* **16**(1), 12 – 19 (2011), next Generation Networks
19. Richhariya, P., Bhopal, Singh, P.K.: A survey on financial fraud detection methodologies (2012)
20. Seeja, K.R., Zareapoor, M.: Fraudminer: A novel credit card fraud detection model based on frequent itemset mining. *The Scientific World Journal* **2014**, 1–10 (2014)
21. Singh, A., Narayan, D.: A survey on hidden markov model for credit card fraud detection (2012)
22. Torgo, L.: *Data Mining with R: Learning with Case Studies*, Second Edition. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, CRC Press (2016), <http://www.dcc.fc.up.pt/~ltorgo/DataMiningWithR/>
23. Wang, S.: A comprehensive survey of data mining-based accounting-fraud detection research. In: 2010 International Conference on Intelligent Computation Technology and Automation. vol. 1, pp. 50–53 (May 2010)
24. Wei, Z., Yang, J., Shen, H., Yu*, D.: A cascade random forests algorithm for predicting protein-protein interaction sites. *IEEE Transactions on NanoBioscience* **14**(7), 746–760 (Oct 2015)
25. Yu, C.H.: A fuzzy genetic approach for optimization of online auction fraud detection. In: Hung, J.C., Yen, N.Y., Li, K.C. (eds.) *Frontier Computing*. pp. 965–974. Springer Singapore, Singapore (2016)

A Fast AEMST Algorithm for High-Dimensional Datasets by Removing Redundant Distance Computations

Xiaochun Wang¹, Xia Li Wang², Xuan Xiong Lin³

¹School of Software Engineering, Xi'an Jiaotong University, Xi'an, 710049, CHINA
xiaocchunwang@mail.xjtu.edu.cn

²School of Information Engineering, Changan University, Xi'an, 710061, CHINA
xlwang@chd.edu.cn

³Xi'an Jiaoda JointSky Corporation, Xi'an, 710065, CHINA
linxuanxiong88@163.com

Abstract Although nearest neighbor search between data points in a given dataset can be conducted fast in Euclidean space, finding Euclidean minimum spanning trees in a linear time is not practical for high-dimensional datasets. Therefore, attention has been paid to the design of efficient minimum spanning tree algorithms for high-dimensional datasets. In this paper, we propose a new fast approximate Euclidean minimum spanning tree algorithm designed especially for high-dimensional datasets by reducing the number of distance computations. To do so, a new distance computation scheme is utilized for distance calculation between two non zero vectors according to the standard inner product formulation. Experiments conducted on sample datasets demonstrate the efficiency of the proposed method.

Keywords: minimum spanning tree, Euclidean distance, Euclidean minimum spanning tree, approximate Euclidean minimum spanning tree, principal component analysis

1 Introduction

In traditional MST problems, an undirected connected and weighted graph with a set of V vertices and a set of E edges is given, for which a minimum spanning tree (MST) is a sub-graph that spans over all the vertices without any cycle and has the minimum sum of total weights among all such sub-graphs. The first minimum spanning tree algorithm was proposed by Otakar Boruvka in 1926 [1]. Since then, optimal exact MST algorithms, fast approximate MST algorithms, distributed MST algorithms [2], and parallel MST algorithms [3] have been developed. Minimum spanning tree algorithms have so far been broadly applied in image segmentation [4][5], clustering [6][7][8], classification [9], etc.

In today's MST tasks, usually, a set of N d -dimensional data points is given and the problem is commonly solved in the Euclidean setting, giving rise to the so-called

Euclidean minimum spanning tree (EMST) problem. In this case, for a complete graph with $V = N$ vertices and $E = N(N-1)/2$ edges, standard MST algorithms, such as Kruskal's [10] and Prim's [11], have a time complexity roughly equal to $O(dN^2)$ and are practical only for small low dimensional datasets. Thus, the computational cost of building an EMST for a high-dimensional dataset is often the bottle neck of efficiency for many practical applications. To solve this problem, fortunately, in many practical applications, an exact EMST can be generally replaced by an approximate one which can be computed more efficiently without degrading the quality of the final application.

To find exact EMSTs quickly, especially on a single modern central processing unit, the goal is to lessen the number of distance computations used to construct an EMST as much as possible by removing redundant distance calculations between two non-zero vectors. Here by redundant distance computations, we mean those distance calculations which can be figured out from other existing distance values, but with less computing resources being consumed, and those neighboring edges in an EMST that can be determined within a much smaller neighborhood instead of a global neighborhood. For the sake of utilizing the Euclidean distance, the proposed method based on the inner product can deal with high-dimensional data and allow us to save the number of distance computations tremendously. Experiments conducted on sample datasets demonstrate the improvement in efficiency of the proposed approximate EMST algorithm in comparison to standard MST algorithms while keeping high accuracy.

The rest of this paper is organized as follows. In Section 2, we review some existing work on MST algorithms. We next present our proposed approach in Section 3. In Section 4, we verify the efficiency of our methods with experiments. Finally, conclusions are made and future work is discussed in Section 5.

2 Related work

For a given connected and weighted graph $G = (E, V)$, Borůvka's algorithm begins with each vertex of a graph being a tree, and for each consecutive iteration, it selects the shortest edge from a tree to another tree and combines them. This process continues until all the trees are combined into one tree [1]. Proposed independently by Jarník [12], Prim [11] and Dijkstra [13] in 1930, 1957 and 1959, respectively, the famous Prim's algorithm first arbitrarily selects a vertex as a tree, and then repeatedly adds the shortest edge that connects a new vertex to the tree, until all the vertices are included. Proposed in 1956, Kruskal's algorithm starts with sorting all the edges by their weights in a non-decreasing order, treats each vertex as a tree, and iteratively combines the trees by adding edges in the sorted order excluding those leading to a cycle until all the trees are combined into one tree [10]. The time complexity of these classic MST algorithms is $O(E \log V)$.

To construct an MST in the Euclidean setting, standard Prim's algorithm requires a quadratic running time. To be more efficient, in 1978, Bentley and Friedman proposed to use a kd-tree in Prim's algorithm to enhance the search for the next edge to

add to the tree, which can reach an $O(N \log N)$ running time for most data distributions [14]. In 1985, Preparata and Shamos gave a lower bound of $\Theta(N \log N)$ for the EMST problem, which has been the tightest known lower bound [15]. In 1993, Well-Separated Pair Decomposition (WSPD) was proposed by Callahan and Kosaraju, and forms the basis of most recent EMST algorithms [16]. The WSPD partitions data points into a set of pairs of tree nodes such that the nodes in any pair are farther apart than the diameter of either node. It can be shown that the WSPD has $O(N)$ pairs of nodes and that an MST is a subset of the edges formed between the closest pair of points in each pair of nodes. In 2000, WSPD was applied to compute neighbors of components for Boruvka's algorithm to find edges of an MST by Narasimhan et al. [17]. However, the constant in the $O(N)$ size of the WSPD grows exponentially with the data dimension and is often very large in practice. In 2010, a new dual-tree algorithm for efficiently computing an EMST [18] was presented by March et al., which is superficially similar to the method in [17] except that the WSPD is replaced by a new dual-tree data structure, and referred to in the following as FEMST algorithm. They used adaptive algorithm analysis to prove the tightest (and possibly optimal) runtime bound for the EMST problem to-date. Experiments conducted demonstrated the scalability of their method on large astronomical datasets.

Being an alternate to exact EMST algorithms, approximate EMST (AEMST) algorithms have been also developed. In 1988, Vaidya [19] employed a group of grids to partition a data set into identical-sized cubical boxes, for each of which, a representative point was determined, and within each of which, points were connected to the representative. Any two representatives of two cubical boxes were connected if corresponding edge length was between two specific thresholds. In 1993, Callahan and Kosaraju [16] proposed to utilize WSPD of a data set to extract a sparse graph from the complete graph and then apply an exact MST algorithm to it. More recently, efficient AEMST algorithms have been developed for clustering. In 2009, Wang et al. [20] employed a divide-and-conquer scheme for AEMST to detect longest edges in an EMST at an early stage for clustering. In the same year, Lai et al. proposed a two-stage Hilbert curve based AEMST algorithm for clustering [21]. In 2013, Wang et al. proposed a fast AEMST algorithm, which was superficially similar to the method in [14] except that an iDistance indexing structure was employed for fast kNN search in high-dimensional datasets [22]. In 2015, Zhong et. al. proposed a fast two-stage AEMST algorithm with theoretical time complexity of $O(N^{1.5})$ [23] which is referred to in the following as FAEMST algorithm. In the first stage, K-means was employed to partition a dataset into $N^{1/2}$ clusters. Then an exact EMST algorithm was applied to each cluster to produce $N^{1/2}$ EMSTs which were connected to form an approximate EMST. In the second stage, the dataset was repartitioned so that the neighboring boundaries of a neighboring pair produced in the first stage were put into a cluster. With these $N^{1/2}-1$ clusters, another AEMST was constructed. Finally, the two AEMSTs were combined to generate a more accurate AEMST.

3 The proposed approximate algorithm

In this section, we first propose an algorithm to remove the redundant distance computations involved in the Prim's EMST algorithms by a more analytical examination on the Euclidean distance definition. Based on this observation, a new fast approximate EMST algorithm is then developed based on PCA (Principal Component Analysis) transformation.

3.1 A simple idea

Given two data points in d -dimensional space, $p=(p_1,p_2,\dots,p_d)$ and $q=(q_1,q_2,\dots,q_d)$, the standard Euclidean distance from p to q , $\text{dist}(p, q)$, or from q to p , $\text{dist}(q,p)$, is given by the following formula,

$$\text{dist}(p, q) = \text{dist}(q, p) = \sqrt{\sum_{i=1}^d (q_i - p_i)^2} \quad (1)$$

The position of a data point in a Euclidean d -space is a Euclidean vector. So, p and q are Euclidean vectors, starting from the origin of the space, and the Euclidean norm, Euclidean length, or magnitude of a vector measures the length of the vector as,

$$\|p\| = \sqrt{p_1^2 + p_2^2 + \dots + p_d^2} = \sqrt{p \bullet p} \quad (2)$$

where the last equation involves the dot product. By dot product format, Equation 1 can be rewritten as,

$$\begin{aligned} \text{dist}(q, p) &= \sqrt{\sum_{i=1}^d (q_i - p_i)^2} = \sqrt{\sum_{i=1}^d (q_i^2 + p_i^2 - 2q_i p_i)} \\ &= \sqrt{\sum_{i=1}^d q_i^2 + \sum_{i=1}^d p_i^2 - 2\sum_{i=1}^d q_i p_i} = \sqrt{\|q\|^2 + \|p\|^2 - 2q \bullet p} \end{aligned} \quad (3)$$

From Equation 3, it can be seen that, if the Euclidean norm for each data vector is pre-computed, the calculation of the Euclidean distance between two vectors can be reduced to the computation of the inner product of two vectors. Further, if at least one of the vectors is highly sparse as shown in the following figure, a considerable amount of distance computations can be saved.

p	0	0	5	0	0	0	0	1	0	1	0
q	0	1	0	0	0	0	0	3	0	0	0

Fig. 1. Two sample vectors.

This observation is promising since, as summarized in Table 1, six benchmark datasets obtained from UCI machine learning repository and extensively used in data mining applications consist of a large amount of zero values [24].

Table 1. Frequency of Zeros in Five UCI Datasets

Data Name	# of Objects/Dimension	% of 0
CorelHistogram	68,040/32	17.64%
IPUMS	88,443/61	36.51%
Coverttype	581,012/55	76.58%
KDDCup1999	4,898,430/42	67.9%
USCensus1990	2,458,285/68	56.97%

Therefore, for highly sparse vectors where only those dimensions for non-zero values need to be considered, not only pairwise distance computations can be reduced to the inner product among a few dimensions, but also the vectors can be represented more efficiently by only remembering the dimensions for which the attribute values are nonzeros, which is especially good for highly sparse high-dimensional datasets. For example, for vector q in Fig.1, only two integers (for the nonzero dimensions) and two float values can be used to represent it in stead of the original eleven float values. Then for inner product computation, only the values of the second and the eighth dimensions of vector p need to be checked since vector q only has nonzero elements in those two dimensions.

3.2 Principal component analysis

With the fast distance computation scheme explained in previous subsection, reduction of computational complexity and further acceleration are possible if unnecessary operations in the calculation of the Euclidean distance for every vector dimension can be eliminated early. However, for high dimensional datasets from other application domains where data consist of a large number of nonzero values, the curse of high dimensionality tends to be a major obstacle in the development of efficient EMST methods. To partially circumvent this problem, we propose a preprocessing method for such early elimination of unnecessary inner product operations.

To provide early elimination of unnecessary inner product operations as a more efficient means for EMST construction, an orthogonal transformation must be used to preserve the distances between vectors. There are various orthogonal transformations. Here we are particularly interested in principal component analysis (PCA, or KL transform) where a basis for the best representation of multidimensional vector fluctuations can be found [25]. In PCA, the eigenvalue decomposition of a covariance matrix is performed, and the eigenvectors are made the new basis. The eigenvector with the largest eigenvalue is called the first principal component, after which comes the second principal component, and so on. The covariance increases with the eigen-

values. Early detection of unnecessary operations can be made more practicable by preliminary transformation of the data so that the coordinates are arranged in the order of principal components. To explain, d -dimensional vector data q_i ($i = 1, 2, \dots, N$) are arranged by matrix Q :

$$Q = [q_1 \ q_2 \ \dots \ q_N] \quad (4)$$

Q is a $d \times N$ matrix, with every column representing one data entry. The i -th row of matrix Q includes the i -th variation of every data entry. In the following, let Q_{ik} denote the element, (i, k) , of the matrix Q , μ_i is defined to be the average of the i -th variation for matrix Q ,

$$\mu_i = \frac{1}{N} \sum_{k=1}^N Q_{ik} \quad (5)$$

Then the covariance, s_{ij} , of the i -th variation and the j -th variation is defined as follows,

$$s_{ij} = \frac{1}{N} \sum_{k=1}^N (Q_{ik} - \mu_i)(Q_{jk} - \mu_j) \quad (6)$$

The matrix $S = [s_{ij}]$ whose elements are the covariances, s_{ij} , is called the covariance matrix. The covariance matrix S is a $d \times d$ matrix including d sets of eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_d$ and eigenvectors u_1, u_2, \dots, u_d . The eigenvalues are arranged in descending order ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$). Now consider the matrix U used to arrange the eigenvectors,

$$U = [u_1 \ u_2 \ \dots \ u_d] \quad (7)$$

Vector elements can be arranged in ascending order of variance by representing vector data q_i in coordinates whose basis is the eigenvectors u_1, u_2, \dots, u_d . Such a representation is equivalent to using the transformed data p_i as follows:

$$p_i = U^T q_i \quad (8)$$

Conventional PCA aims at the reduction of the data dimension, and hence the number of eigenvectors used is smaller than the dimension d of the original data. For our purpose (that is, the early elimination of unnecessary operations in distance calculation), all the eigenvectors are retained and the obtained data after the PCA transformation go through a quantization so as to quantize most small numbers in the new dimensions into zeros.

3.3 The proposed approximate EMST algorithm

Suppose we are given a dataset Q consisting of N objects with a dimension of d and the Euclidean distance in the format of Equation 3 is adopted in the approximate Eu-

clidean minimum spanning tree algorithm in this study. The proposed algorithm is composed of the following three steps.

Step 1: (Preprocess the data by performing PCA followed by a scalar quantization)

1. Conduct PCA on a given dataset consisting of N objects, Q , to produce the KL transformed data Q^T .
2. Sort the absolute values of all the obtained Nd coefficients in ascending order.
3. Determine a threshold value such that (at least) the first $(dN-dN/10)$ out of all the obtained dN coefficients can be set to zero.
4. Obtain a preprocessed data, P , by thresholding Q^T using the value determined in Step 3.

Step 2: (Construct an EMST on P)

5. Apply standard Prim's algorithm to construct an EMST on P . The resulted EMST(P) is contained in two arrays, one for storing the edge weight and the other for remembering the parent index of each node.

Step 3: (Construct an Approximate EMST on Q)

6. Update the EMST on P obtained in Step 2 by replacing the edge weights in P with the exact distances between each node and its parent node calculated in the original Q space to obtain the approximate EMST on Q .

The above algorithm is a local heuristic that runs fast especially for high dimensional datasets. In Step 1, we employ a preprocessing method for data reduction. This method is simple to implement and tends to compress data considerably to within a few dimensions in the transformed data. In Step 2, an exact EMST is constructed on P . In Step 3, with the tree nodes' relationship being obtained in Step 2, an approximate EMST on Q is obtained by updating the tree edge weights with the Euclidean distances computed in the original data space. We would like to mention that, for low dimensional datasets, the proposed method may not outperform state-of-the-art fast EMST algorithms since the saved computations can be marginal.

4 Experiments and results

In this section, we conduct two sets of experiments to compare the effectiveness of the proposed approximate EMST algorithm with state-of-the-art EMST algorithms, including the brute force (that is, the $O(N^2)$ Prim's algorithm), the FEMST algorithm and the FAEMST algorithm, on several real datasets obtained from UCI machine learning repository [25]. Naturally designed for classification and machine learning applications, these datasets have been picked in a way so as to result in considerable variability in terms of the number of attributes.

We implement all the algorithms in C++ and perform all the experiments on a computer with AMD A6-4400M Processor 2.70GHz CPU and 4.00G RAM. The operating system running on this computer is Windows 7. In our evaluation, we focus on the runtime performances of EMST algorithms on different data sets. We use the timer utilities defined in the C standard library to report the CPU time. The results show

that, overall, our proposed algorithm opens a door for superior performance over state-of-the-art EMST algorithm on high dimensional datasets.

4.1 Experiment I

In this set of experiments, we tested the FEMST and the proposed AEMST on several relatively low-dimensional but large-sized real datasets, which are briefly summarized in Table 1. The running time results are shown in Table 2. By performing experiments on this set of datasets, we would like to show that the FEMST algorithm does a very good job on large low-dimensional datasets in comparison with our algorithm.

Table 2. Runtime Performances on Five UCI Datasets

Data Name	FEMST Method (seconds)	OUR METHOD (seconds)
CorelHistogram	72	10 823
IPUMS	8	16 617
Coverttype	160	211 918
KDDCup1999	140 723	905 990
USCensus1990	244	725 218

From Table 2, we can see that the FEMST method is very efficient when running on these five datasets. However, the construction of EMST using our method takes a much longer time. This is because our proposed method relies on the dimensions of the dataset and therefore may not be effective for low dimensional data.

4.2 Experiment II

In this subsection, we compare the effectiveness of the proposed approximate EMST algorithm with state-of-the-art EMST algorithms, including the FEMST algorithm and the FAEMST algorithm, on two high dimensional datasets. The empirical performance on the two datasets is also compared with the brute force algorithm (that is, the $O(N^2)$ Prim's algorithm). The two high dimension data sets from UCI are briefly summarized in Table 3.

Table 3. Description of Datasets

Data Name	Data Size	Data Dimension
ISOLET	7 797	617
MNIST	10 000	784

The ISOLET dataset consists of data extracted from the recorded spoken name of each letter of the alphabet and contains 7797 instances with 617 attributes. To obtain the dataset, 150 subjects spoke the name of each letter of the alphabet twice. Hence, this resulted in 52 training examples from each speaker. The speakers were grouped into sets of 30 speakers each, and were referred to as isolet1, isolet2, isolet3, isolet4, and isolet5. The data appears in isolet1+2+3+4.data in sequential order, first the speakers from isolet1, then from isolet2, and so on. The test set, isolet5, is a separate

file. Fig. 2 displays the (sorted absolute) value distribution of the ISOLET dataset consisting of 7797×617 values before and after PCA. The original value range is $[-1, 1]$. The value range after PCA is $[-10.02, 11.63]$. From the figure, we can see that, in comparison with the original dataset, the transformed data has a much wider value range and most of them are close to zeros.

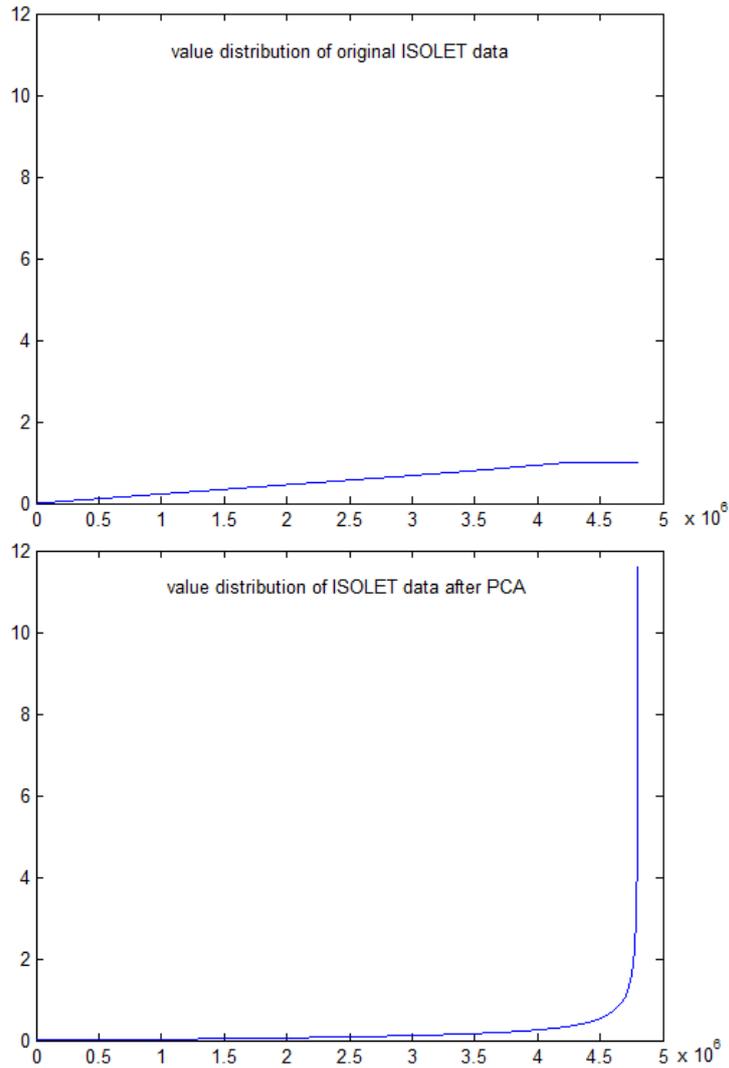


Fig. 2. Value distribution (upper) before and (lower) after KL transformation for ISOLET data.

The MNIST dataset consists of data extracted from the handwritten digits and contains 10000 instances with 784 attributes. The digits have been size-normalized and centered in a fixed-size image. It is a subset of a larger set available from NIST. The

original black and white (bilevel) images from NIST were size normalized to fit in a 20×20 pixel box while preserving their aspect ratio. The resulting images contained grey levels as a result of the anti-aliasing technique used by the normalization algorithm. The images were centered in a 28×28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28×28 field. Fig. 3 displays the (sorted absolute) value distribution of the MNIST dataset consisting of 10000×748 values before and after PCA. The original value range is from 0 to 255. The value range after PCA is $[-1423, 2366]$. From the figure, we can see that, in comparison with the original dataset, the transformed data has a much wider value range and most of them are close to zeros.

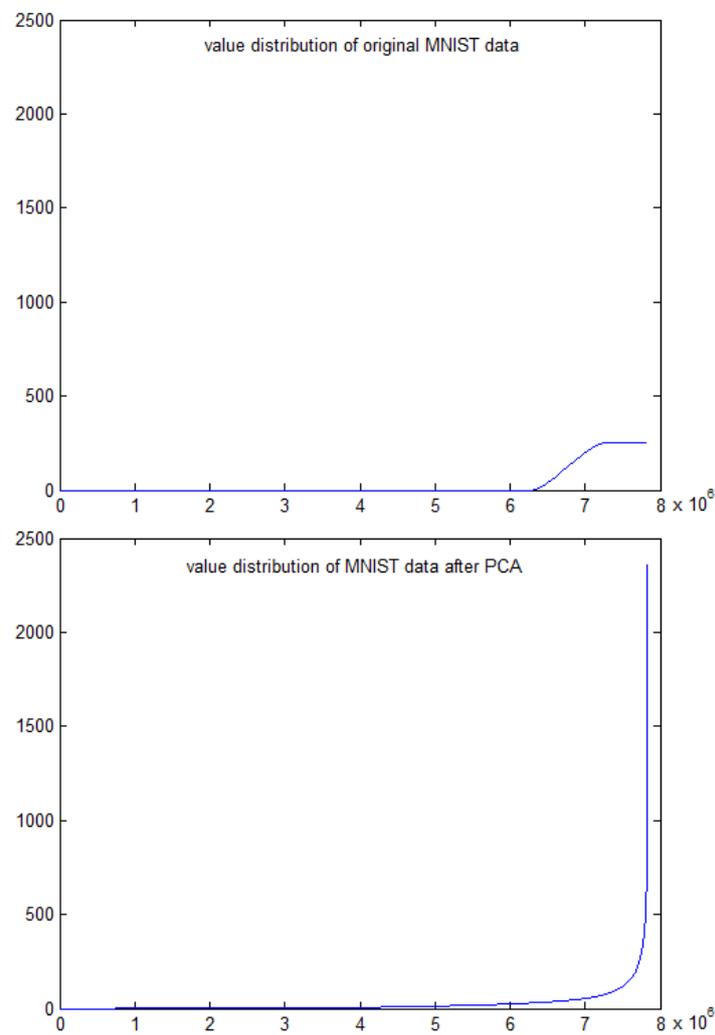


Fig. 3. Value distribution (upper) before and (lower) after KL transformation for MNIST data.

The threshold values for setting $(dN-dN/10)$ and $(dN-dN/100)$ values to zero for both datasets to obtain 10 times faster and 100 times faster over the brute force Prim's algorithm are summarized in Table 4. From the table, it can be seen that, with PCA, our proposed inner product based approach can construct EMST efficiently.

Table 4. Threshold Values from Both Datasets

Data Name	$dN-dN/10$	$dN-dN/100$
ISOLET (before PCA)	1	1
ISOLET (after PCA)	0.2231	1.2366
MNIST (before PCA)	211	254
MNIST (after PCA)	29.7337	214.5969

We show in Table 5 the best experimental results of our method in comparison with Prim's algorithm, FEMST algorithm and FAEMST algorithm. From the table, it can be clearly seen that, overall, our method is the only one that performs best. FEMST performs worse than the Prim's algorithm for high dimensional data. Therefore, overall, our method performs the best for high dimensional data.

Table 5. Runtime Performances

Algorithms	ISOLET (seconds)	MNIST (seconds)
Prim	417	286
FEMST	519	1 135
FAEMST	231	164
OUR METHOD	73	51

4.3 Time complexity analysis

We note that since the Prim's algorithm is used for the construction of approximate EMST in our algorithm, the running time complexity of the proposed method is proportional to the size of the dimensionality of the data (that is, d) instead of the size of the data set (that is, N). The bottleneck operation is in the compression of the major information which is contained in the original data into a few dimensions in the transformed domain. Then in each iteration of the Prim's algorithm which is applied upon the data after the PCA, the number of inner products to be computed is proportional to the size of nonzero dimensions. For this reason, the scalability of the algorithm is proportional to the reduced dimension size of the data. At the same time, since the effect of dimension reduction of the PCA algorithm is essentially data dependent over the data space, it is not possible to propose a reasonably tight estimation of the complexity behavior in closed form.

5 Conclusions

In this paper, we develop a new technique for approximate EMST problems that is especially suited to very high-dimensional data sets. The method works by removing

redundant distance computations involved in the inner product calculations of Euclidean distances and can be enhanced easily by a dimension deduction method, that is, principal component analysis. This technique for approximate EMST has advantages over simple index based FEMST that works remarkably well for large low dimensional datasets but cannot overcome the effects of the dimensionality curse. Experiments conducted on sample datasets demonstrate the efficiency of the proposed method. In our future work, we would like to extend our work to larger-sized high dimensional data since, for larger-sized data, the PCA may not work as easily as for small to medium sized data.

Acknowledgment

The authors would like to thank the Chinese National Science Foundation for its valuable support of this work under award 61473220.

References

1. O. Borůvka. 1926. O jistém problému minimálním (About a Certain Minimal Problem). *Práce moravské přírodovědecké společnosti v Brně. III* (1926) 37-58 (in Czech with German summary).
2. M. Khan and G. Pandurangan. 2008. A Fast Distributed Approximation Algorithm for Minimum Spanning Trees, *Distributed Computing*, 20, 6 (Apr. 2008) 391–402. DOI: <https://doi.org/10.1007/s00446-007-0047-8>
3. S. Pettie and V. Ramachandran. 2002. A Randomized Time-work Optimal Parallel Algorithm for Finding a Minimum Spanning Forest, *SIAM Journal on Computing*, 31, 6 (2002) 1879–1895 DOI: <https://doi.org/10.1137/S0097539700371065>
4. L. An, Q.S. Xiang, and S. Chavez. A Fast Implementation of the Minimum Spanning Tree Method for Phase Unwrapping. *IEEE Transactions on Medical Imaging*, 19, 8 (2000), 805–808. DOI: 10.1109/42.876306
5. Y. Xu, and E.C. Uberbacher. 1997. 2D Image Segmentation Using Minimum Spanning Trees. *Image and Vision Computing*, 15 (1997) 47-57. DOI: [https://doi.org/10.1016/S0262-8856\(96\)01105-5](https://doi.org/10.1016/S0262-8856(96)01105-5)
6. C.T. Zahn. 1971. Graph-theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computer*, C20 (1971) 68-86. DOI: 10.1109/T-C.1971.223083
7. Y. Xu, V. Olman and D. Xu. 2002. Clustering Gene Expression Data Using a Graph-theoretic Approach: an Application of Minimum Spanning Trees. *Bioinformatics*, 18 (4) (2002) 536–545. DOI: 10.1093/bioinformatics/18.4.536
8. C. Zhong, D. Miao, and R. Wang. 2010. A Graph-theoretical Clustering Method Based on Two Rounds of Minimum Spanning Trees. *Pattern Recognition*, 43(3) (2010) 752-766. DOI: <https://doi.org/10.1016/j.patcog.2009.07.010>
9. P. Juszczak, D.M.J. Tax, E. Pełkalska and R.P.W. Duin. 2009. Minimum Spanning Tree Based One-class Classifier. *Neurocomputing*, 72 (2009) 1859-1869. DOI: <https://doi.org/10.1016/j.neucom.2008.05.003>
10. J. B. Kruskal. 1956. On the Shortest Spanning Subtree of A Graph and the Traveling Salesman Problem. In *Proceedings of the American Mathematical Society*, 7, 1 (Feb., 1956). 48-50. DOI: <https://doi.org/10.1090/S0002-9939-1956-0078686-7>

11. R.C. Prim. 1957. Shortest Connection Networks and Some Generalizations. *Bell System Technical Journal*, 36 (1957) 567-574. DOI:10.1002/j.1538-7305.1957.tb01515.x
12. V. Jarník. 1930. O jistém problému minimálním (About a Certain Minimal Problem). *Práce moravské přírodovědecké společnosti v Brně VI* (1930) 57-63 (in Czech).
13. E.W. Dijkstra. 1959. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik I* (1) (1959) 269-271. DOI:10.1007/BF01386390
14. J. Bentley and J. Friedman. 1978. Fast Algorithms for Constructing Minimal Spanning Trees in Coordinate Spaces. *IEEE Transactions on Computers*, 27 (1978) 97-105. DOI:10.1109/TC.1978.1675043
15. F. P. Preparata and M. I. Shamos. 1985. *Computational Geometry*. Springer-Verlag, New York, 1985.
16. P. Callahan, and S. Kosaraju. 1993. Faster Algorithms for Some Geometric Graph Problems in Higher Dimensions. In *Proceedings of 4th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1993, 291-300.
17. G. Narasimhan, M. Zachariasen and J. Zhu. 2000. Experiments With Computing Geometric Minimum Spanning Trees. In *Proceedings of ALENEX'00*, (2000) 183-196.
18. W.B. March, P. Ram, and A.G. Gray. 2010. Fast Euclidean Minimum Spanning Tree: Algorithm, Analysis, and Applications. In *Proceedings of 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*, Washington (2010) 603-612. DOI:10.1145/1835804.1835882
19. P.M. Vaidya. 1988. Minimum Spanning Trees in k-dimensional Space. *SIAM Journal on Computing*, 17 (3) (1988) 572-582. DOI:https://doi.org/10.1137/0217035
20. X. Wang, X.L. Wang and D.M. Wilkes. 2009. A Divide-and-Conquer Approach for Minimum Spanning Tree-based Clustering. *IEEE Trans. Knowledge and data engineering*, 21, 7 (2009) 945-958, DOI:10.1109/TKDE.2009.37
21. C. Lai, T. Rafa and D.E. Nelson. 2009. Approximate Minimum Spanning Tree Clustering in High-dimensional Space. *Intelligent Data Analysis*, 13 (2009) 575-597.
22. X. Wang, X.L. Wang, C. Chen and D.M. Wilkes, Enhancing Minimum Spanning Tree Based Clustering by Removing Density Based Outliers. *Digital Signal Processing*, 23, 5 (2013) 1523 - 1538, DOI:https://doi.org/10.1016/j.dsp.2013.03.009
23. C. Zhong, M. Malinen, D. Miao and P. Fränti. 2015. A Fast Minimum Spanning Tree Algorithm Based on K-means, *Information Sciences*, 295 (C) (2015) 1-17. DOI:https://doi.org/10.1016/j.ins.2014.10.012
24. <http://archive.ics.uci.edu/ml/datasets.html>.
25. S. Ajioka, &S. Tsuge, M. Shishibori and &K. Kita, Fast Multidimensional Nearest Neighbor Search Algorithm Using Priority Queue, *Electrical Engineering in Japan*, 164, 3 (2008) 69-77. DOI:10.1002/eej.20502

Server Failure Prediction Framework for IT Infrastructure Supporting Critical Applications

Mythili Krishnan¹ and Madhan Kumar Srinivasan²

Accenture Solutions Pvt. Ltd., Bengaluru, India

¹mythili.krishnan@accenture.com

²madhan.srinivasan@accenture.com

Abstract. Server downtime be it an application server or a web server or a database server is a cause that plagues the applications that it supports. Continuous availability of servers (in terms of accessibility, recoverability and minimal downtime) and its related applications is critical, and a negation leads to direct as well as indirect costs. IT downtime can cost a lot for companies in terms of dollar as well as productive time. In a typical IT infrastructure environment, all deployed applications across the servers will compete for resources (processor, memory, storage, etc.). When there are issues in availability of these resources, CIs/Servers encounter performance issues, which in turn impact performance of applications running on those CIs/Servers. These kind of performance issues are also encountered in Cloud Virtualization Infrastructure wherein the virtual machines will make use of the underlying hardware infrastructure capabilities of the same physical server using the hypervisor. Among many disruptions in today's enterprises, failure of servers and its related applications is a high potential hazard which has direct impacts on company's production, internal processes, online sales, customer service, etc. This early alert generation method would help managers or technology analysts detect when the servers might fail or cause critical incidents and take anticipatory actions that could help avoid any losses to the business. This research work is not only an exploration of the detection methods that can be utilized to predict when a Server/Configuration Item in a traditional or Cloud Infrastructure will fail but is also a readily available framework that is easily deployable by clients.

Keywords: Prediction, Predictive Analytics, Data Analytics, Configuration Item, Unsupervised Learning, Supervised Learning, Time Series, Cloud Security, eCloudIDS, iCloudIDM, Cloud Computing Security Taxonomies, Cloud Security Framework.

1 Introduction

Infrastructure services and IT operations is an untapped area when it comes to the world of analytics and statistics. With the huge amount of data available, statistical analysis can come to the rescue of Infrastructure services by truly transforming the way operations are run today. The myriad use cases in network services and journey to cloud can make the shift from traditional and classical practices to a new data driven and analytics

infused reality. The Gartner report in Dec 2017 talks about the importance of digital transformation in Infrastructure operations that requires “IT agility and velocity that outstrips classical architectures and practices” and a related report in April 2018 that talks about the importance and necessity of Artificial intelligence for IT operations. In this light, simple automation tools of the past are not sufficient to drive higher business impact. As technologies become more complex and data becomes unmanageable, businesses that simply rely on monitoring software or applications will not be able to catch the bus, detection of problems in advance and providing pro-active automated solutions is the ask of the day.

On average IT downtime costs businesses \$1.55 million every year [1] while 56% portion of Fortune 500 companies who experience weekly average downtime of at least 1.6 hours [2]. In 2011, Japan's Mizuho Bank [3] suffered a system meltdown that shuttered its massive ATM and Internet banking services for several days and delayed processing of 1.16 million transactions, worth a total of about \$10 billion. The company reimbursed customers in a \$2.7 million settlement. While such massive failures influence the revenues directly; minor application disruptions in many organizations reflects in senior executives getting laid off which can be considered more serious and will result in indirect losses to companies strategy and business.

Hence, it is highly critical for IT departments to deliberate the consequences of service outages/failures, especially for those servers and applications which are considered to be highly critical.

2 Related Work

There are some use cases as well as methodology that has been studied and initiate in the areas of fault prediction and predictive maintenance. Typically, there has been work in this area which mostly deals with the method of unsupervised algorithms and anomaly detection with log data. There are few problems that can be encountered in this approach. Firstly, analyzing log data for a diverse range of applications is time consuming because each application will have different layers in addition to the layer specific to the application and the innumerable log file data analysis along with extraction of correct data will become a cumbersome process. Also, in this case the analysis needs to be done for each application separately. In case of analysis done with the server performance data the methods of anomaly detection though fruitful yet is not devoid of certain limitations. Firstly, anomaly detection works on the basic premise of what is normal vs what is aberrant to the norm. Here the definition of what is the norm itself might be varied and dependent on other exogenous factors like software failures, human errors, capacity upgradation issues or other business policies. To provide an example: Normally for server, a CPU utilization of 90% and above would be considered above normal and an aberration but this might be normal because due to business requirements some load was increased on this server on a particularly busy day.

In a related paper RK Sahoo et al [4] authors talk about how time series models can be used to successfully predict system performance parameters. Though several techniques have been proposed, but the large-scale production solution for multiple node

failures and proactive system management is still a lingering question. Successful prediction of a failure node can help in steering jobs away from the failure node. Such predictions can also help in keeping at bay unplanned system outages by proper planning of system maintenance activities. They used different algorithms for prediction and concluded that different algorithms are useful in different scenarios. They used filtering techniques to reduce the data size by disregarding irrelevant information. Based on event log, system activity reporting and other parameters with the help of algorithms like time series, Bayesian and rule-based algorithms. In their future work, they hint at building a hybrid technique comprising all 3 aspects of analysis, prediction and probing to include more system related aspects.

When we consider IT infrastructure for applications, then Cloud becomes an integral part of it [5] [6]. When servers are becoming virtual than physical, problems and failures are scaled to multiple times than ever before. In a logical cloud world, in addition to handling system performance and failures, it also comes with numerous security issues [7]. Madhan Kumar Srinivasan et al designed a secure cloud framework like eCloudIDS [8], a hybrid two tier expert engine in a cloud computing environment. The sub-systems utilize a combination of supervised and unsupervised machine learning algorithms with an option of selecting a single algorithm or a combination of algorithms. The uX engine [9] identifies parameters/activities allowed in a VM and by infusing a behavior analyzer sends the input to sX engine. The sX engine then uses supervised learning algorithms to detect anomalous behavior. This hybrid approach from eCloudIDS and its subsystems design [10] [11] was helpful in forming this solution which uses both unsupervised and supervised machine learning techniques together in a single system.

Similarly, Alessandro et al [12] present a machine learning based framework for failure prediction model. They point out that one significant cause of availability and performance degradation is the accumulation of anomalies of different nature. A large part of these anomalies is often associated with errors and/or sub-optimal implementations of applications, which may lead to the occurrence of, e.g. memory leaks, unterminated threads, unreleased locks, file fragmentation, etc. and as has been observed these accounts to about 40% of all the anomalies. The accumulation of these kinds of anomalies can cause exhaustion of system resources over time, and might lead to incremental loss of performance, or even hang/crash of the hosting system. This model is designed to be used independently of a specific kind of application and type of anomaly. In fact, by changing the set of observed system features and defining a proper condition to be met for considering the system as failed, F2PM can be customized for different systems and applications.

This is further corroborated by Felix et al [13] that classical reliability theory and conventional methods do rarely consider the actual state of a system and are therefore not capable to reflect the dynamics of runtime systems and failure processes. Such methods are typically useful in design for long term or average behaviour predictions and comparative analysis. Both industry and academia realized that traditional fault tolerance mechanisms could not keep pace with the growing complexity, dynamics and flexibility of new computing architectures and paradigms. They talk about the different metrics that should be considered along with different algorithms in place. Sreekumar

Vobugari et al [14] [15] describes the best practices in building performance driven effective complex enterprise architectures in present large IT organizations. Also, Gill et al [16] talk about the various reasons for failures in data centres, most common failure prone devices and network reliability. Revathy P et al [17] [18] [19] work presents the challenges that are being faced by enterprises in their big data or Hadoop infrastructures.

Rosli et al [20] authors present the framework of fault proneness prediction application. This work describes a requirement model to show components interaction and a prototype to verify the proposed fault prediction model. The proposed model result shows that construction of fault proneness prediction using genetic algorithm is feasible, adaptable to object oriented metrics and significant for web applications. The aim of the proposed design model is to develop an automated tool for software development group to discover the most likely software modules in web applications to be high problematic in the future.

The problem is explored by Doug et al [21] and they analyze the hardware sensor data to predict failures in a high-end computer server. Their solution heavily relies on servers that are equipped with sensors and they try to predict the server failures by using the classification technique.

Risto Vaarandi [22] talks about how log analysis can help in system and network management. Log files are an excellent source of determining the health status of the system and centralized log monitoring infrastructure can help achieve that. Common log monitoring techniques like fault detection is not able to detect unknown error messages. But the approach of anomaly detection overcomes this by creating a profile of known system messages and messages outside this profile i.e. unknown messages will be tagged as anomalous. With this approach and using temporal patterns and association rules, the data mining problem has been approached by many. But in this paper, he talks about how association rules can not be directly applied to log files because log files do not have a common format. He then presents a new clustering algorithm called Simple Logfile Clustering Tool using which they detect outliers in log files which can aid in anomaly detection. In this context of how event logs have become important for keeping track of operational status of a computing infrastructure, Makanju et al [23] mentions that the problem of finding frequent event type patterns has become an important topic in the field of automatic log file analysis. This points to the fact that such algorithms which help in mining event type patterns is in vogue. They present a new algorithm through iterative partitioning log mining that is able to discover clusters irrespective of how frequently pattern instances appear in the data. They claim that the use of a pattern support threshold, which is mandatory for other similar algorithms, is optional for IPLoM, running IPLoM without a pattern support threshold provides the possibility that all potential clusters will be found. Murray [24] and others [25] [26] also talks about different machine learning methods and a comparison of the same in the context of hard drive failures. They suggest non-parametric methods for detecting rare events.

Liang et al paper [27] has tackled this challenge by looking at RAS event logs from BlueGene/L over a period of 100 days and used spatial characteristics and temporal distribution of failure events. They found strong correlations between the occurrence of

a failure and several factors, including the time stamp of other failures, the location of other failures, and even the occurrence of non-fatal events. Based on these correlations, three simple yet powerful prediction schemes have been designed, and their effectiveness have been demonstrated through analysis and empirical results.

In all these solutions, few problems that will be encountered is most often the errors are infrequent and due to insufficient data, the prediction accuracy will be low. Likewise, the cost of making a wrong prediction can also be huge because this would mean extra effort that needs to be spent to avert failures that might never occur. But the counter argument for this is the aversion activities deployed by engineers would nevertheless help in improving and preemptive tracking of server health. Because of these challenges, the solutions mostly focus only on the prediction accuracy and various algorithms that can be used for effective prediction. While this is the most critical aspect of the business problem in hand, we also need to focus on the over-all framework, the deployment and usage of the solution and the user experience. This paper tries to focus on all these aspects while also providing an effective solution to the prediction problem.

3 Proposed Solution

In the current scenario, most of the businesses must deal with web applications in their day to day operations and a mechanism which detects the efficacy of these web applications can be very profitable for the infrastructure team. Customers are also becoming increasingly empowered, aware and demanding with respect to the quality of service. In this backdrop, it is extremely critical for any business be it in the technology sector, financial services, pharma or automobile to be up and running 24/7. For a day to day business operation, running of the critical applications successfully is of paramount importance. The methodology discussed further in this paper deals with such business cases and illustrates the example of the failure of a critical business application in the various sectors and how pro-active actions can help avert any such failures.

In view of the above factors, the approach proposed in this paper attempts to create a linkage between the upstream system and the downstream system. Here the upstream is the application layer which is linked with the downstream i.e. the infrastructure layer and unsupervised learning is superimposed on supervised learning to attain the best possible results. The gap between the upstream and downstream systems is bridged with the introduction of configuration items and the linkage map between CI and applications. A new approach has been employed to create a model of early detection of prediction of issues with configuration items in the Infrastructure layer supporting critical applications. The polling data or the performance data that affects the server performance like CPU utilization, file system usage are collated and specific errors which can prove blocking for the servers are derived. These errors and their frequencies are taken as indications of the functioning of the configuration items or servers and in turn the applications hosted in the servers. The methodology used is a combination of supervised and unsupervised learning algorithms which firstly, identifies the relationships/patterns that exist in the various errors that have been extracted from the performance data of the servers and configuration items and secondly, detect the anomalies

that are present by using the error information and the relationships that have been observed.

From the analysis of the performance data it was observed that the pattern and the recurrence of the errors are indicative of the operational capability of the servers and configuration items. Additionally, another outcome of the model is the lead time generated for the early prediction of the failures in the servers/configuration items. This model can also be extended to the Cloud Virtualization Infrastructure where the Virtual machines shares the underlying resources of the physical servers hardware infrastructure through the hypervisor software layer that orchestrates the allocation of resources in the form of shared tenancy as and when it is required.

4 Server Failure Prediction Framework

The paper would focus on the methodology that was deployed in the context of predicting issues across CIs/servers for all major applications across 5 different infrastructure services operating groups. These groups cover different sectors like healthcare, e-commerce, automobile, insurance, quality assurance etc. The inclusion of this diverse set ensures sufficient rationalization and generalization of the data and the underlying correlations. The underlying generic structure between infrastructure and application is given in Figure 1.

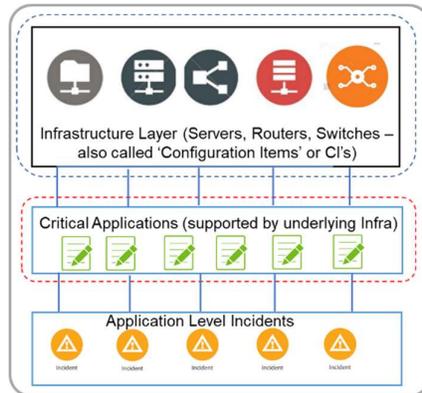


Fig. 1. Relation between Infrastructure and Applications

Deployed Analytical Process.

The process started by studying and understanding the Server Topography and the underlying interrelated systems. This helped us to get an overall as well as a detailed view into the working of various systems and applications. This study is to be restricted at a level so that we can relate the configuration items/Cloud Infrastructure to the various applications. An example is given in Figure 2.

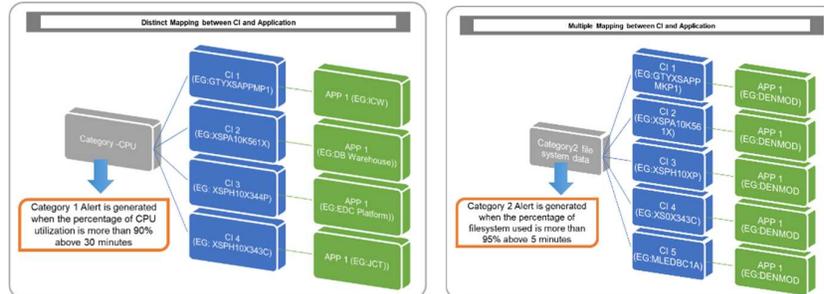


Fig. 2. Distinct Mapping between CI and Applications **Fig. 3.** Multiple Mapping between CI and Applications

In case of many-to-one or one-to-many mapping, all the related incidents need to be considered across CI and across application for 100% capturing of the performance degradation events. The next steps would be to understand the underlying performance files for the servers (physical/virtual)/CIs and all data related to it that can affect the server performance. The data is collected across different domains of infrastructure services operations and then the most representative CIs are considered as targets across all the different data sources. This is the most critical step because correct identification of the target CIs will ensure the achievement of scalability to other unknown/new CIs. Examples of most common representative CI's are: Windows Servers, Linux Servers, Wintel Servers, Oracle Databases, Unix/SQL Servers, Switches/Routers, etc. After we select the target CIs we proceed to the selection of the performance metrics. Three main attributes are key to the performance of the configuration items i.e. how much the CPU is utilized, amount of free space available, number of processes that are currently in progress. 5-8 key performance metrics are selected for these representative target CIs. Examples of the data elements are: CPU Utilization, File System Space Usage, Disc Space Usage, Paging File Usage, Committed Bytes Usage, etc. The resource utilization data will be collated at regular intervals and correlation between these performance metrics and incidents, if any, will be identified. As an example, more than 90% of CPU utilization for more than 20 mins will lead to multiple incidents/performance degradation of the CIs and the related applications. The incident data is to be collected and targets to be created for the time intervals when the incident takes place. An anomaly detection methodology or similar method will be utilized to identify when the CIs and their underlying applications might fail using the utilization data and the incident data. The next critical step is the Master Models Development: Create ~10-20 "master" statistical prediction models, using the data collected from the representative CIs. The models would use various algorithms as suited for the different data types. Post this we would create an Automated Model Solution that will take the 10-20 "master" models and automatically create the 100's of additional statistical models required to cover all the CI's in scope. The model build will be on the top 10 CIs, which will be scalable on all the CIs (~55,000) and ~22,000 servers. Same kind of drivers (resource utilization data) will be effective for all the CIs (~55,000) and ~22,000 servers. The overall model thus will be able to predict the incidents for the individual CIs. There would be

provision for Re-calibration as well with Continuous Learning Model recalibration based on emerging trends. This will be an iterative process and will try to incorporate new emerging patterns and business policies/SME inputs.

Our proposed Server Failure Prediction Framework will rely on 2 key phases:

- i. Build Methodology Phase
- ii. Deployment Methodology Phase

The build phase in Figure 4 will focus on the methodology for building the framework for Server Failure Prediction using historical data, infusing algorithms and predicting the required output.

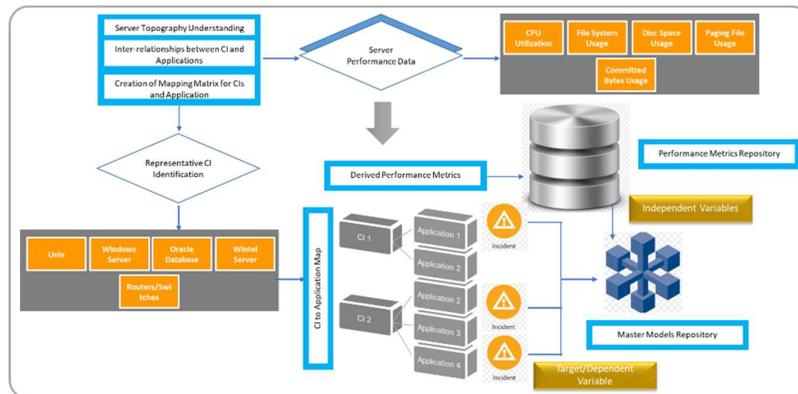


Fig. 4. Server Failure Prediction Framework – Build Methodology

The deployment phase will focus on the actual deployment of the solution i.e. Real time execution of the Server failure prediction framework in a business environment. This is the phase that also determines the user/ business flow of execution of the solution/framework. The framework for deployment is given in Figure 5.

Let's illustrate the framework in greater details. The time series data for the different performance parameters has been considered as the independent variables. In some cases, univariate time series will fit the data (where only one single performance parameter say CPU utilization will be directly affecting the server performance and be the main cause of the incidents). While in most cases a single univariate time series will not work, and we must consider multi-variate time series techniques because multiple variables like CPU Utilization, Paging File Usage, Committed Bytes in conjunction will be the cause for the incidents.

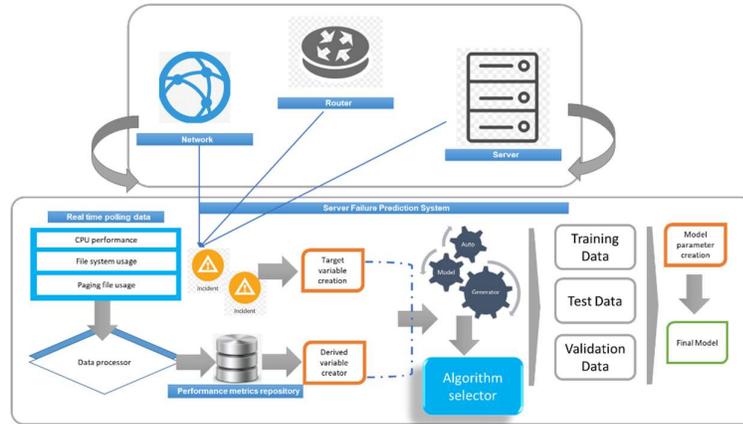


Fig. 5. Server Failure Prediction Framework – Deployment Methodology

Methodology for Target/ Dependent Variable Creation.

The most important part in our analysis is how to relate the downstream application level incidents with the upstream infrastructure related parameters. As we can see in Figure 6, there is a direct relationship between the infrastructure level parameters and the application level incidents. In most cases, the incidents coincide with the time frame, when abnormalities are being observed in the trend of the performance metrics for the configuration items.

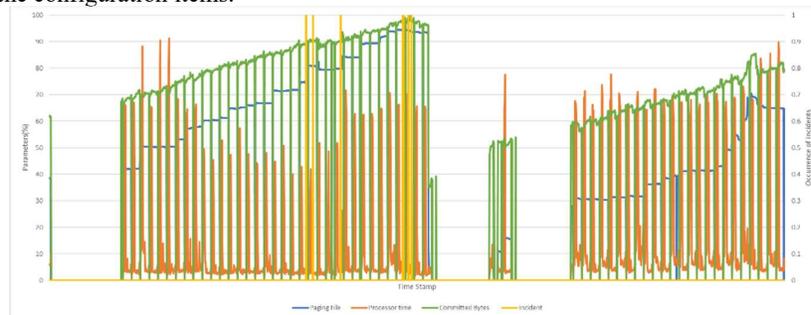


Fig. 6. Relationship between the infrastructure level parameters and the application level incidents

As can be observed from Fig 5, when combination of parameters like paging file, processor time and committed bytes cross normal thresholds, incidents are also generated at the same time window in an interval of 1-2 mins. Hence, the application incident data at 1-2 mins of time interval was chosen as the target variable. Now we will discuss the methodology for independent variable creation.

Methodology for Independent Variable Creation.

We will be forecasting the trend of the independent variables with the help of the time series modelling and these will be the derived independent variables that will be used later.

The forecasting is achieved by first developing a base forecast curve, then developing a factor as weight for curve stitching, adjusting for business policies to arrive at the final forecast. The forecasts are developed for different hours of the day, by different days of the week. The forecasting method is also used to stitch the curve for missing data. For example, we carried out by using the Nearest Neighbor Average that if 4PM data is missing for 4th March then replace the missing values with Average of the 4PM data of 2, 3, 5 & 6th March and for completely missing data, replace with hourly averages computed from the given series of data. Figure 7 is a schematic representation of the method.

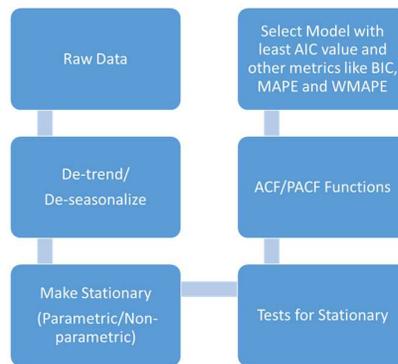


Fig. 7. Methodology for Independent Variable Creation

We considered few time series algorithms like AR, MR, ARIMA, ARMA, Exponential, Double Exponential and Holt Winters models.

There were some pre-criteria for model selection for which the following measures were considered. Pre-criteria like Akaike Information Criteria (AIC), Bayesian Information Criteria (BIC) and Durbin Watson Test were considered. Post criteria like Mean Absolute Percentage Error (MAPE) and Weighted Mean Absolute Percentage Error (WMAPE) were used as measures of forecast accuracies.

Automation of Multivariate Time Series.

Create a repository of derived variables with the time series data (actual and forecasted values). The derived variable creation will be based on different time lags like 10 min, 20 mins, 30 mins, etc. Also, different transformations will be used like log, log-log, log-inverse, etc. This will help to create a series of different features based on past time period patterns.

The use of the forecasted values will be made when we will run the module of automated model selector. Based on the data pattern in current and future time period, it will select the most appropriate model.

For the analysis on what would be normal behavior vs what will be aberrant behavior, each of the performance metric time series is split up into zones, which we call “-normal zone” and “-abnormal zones”. For this the time series data of the independent variables are split into time intervals of 4, 6, 8 hours etc. We then use dynamic time warping and hierarchical clustering to compute the distances between the normal and abnormal zones. This will help us in defining for each time series the abnormalities in the series. We will do this for each of the derived variables created earlier.

Finally, with the help of the derived variables created and the target variables i.e. the incident data we will run supervised regression models. This will help to correctly identify whether the abnormal zones in the time series coincide with the incidents actually happening in the application space.

As an example, one set of master model can be based on Logistic Regression algorithm. For this master model creation, we considered a data period of 1 year and identified the TOP Configuration Items (CIs) causing more number of incidents than other CIs. Incident data was then mapped to identified CI level data based on the Time Stamp and then created the Derived Variables by taking the MIN, AVG and MAX of CPU, Free Space, File System Space, Paging File Usage and Committed Bytes in use for all the CIs. In this phase, we can also introduce other variables that is SME driven and specific to the business to enhance the solution. If multiple incidents happened on the similar hour for multiple CIs, it is considered as single incident to create the Target Variable as a binary variable with Target defined as 1 if incident has occurred else 0. The derived CPU, Free Space, File System Space, Paging File Usage and Committed Bytes in use from the above steps are considered as independent variables. The result of this example model is given in Fig 12.

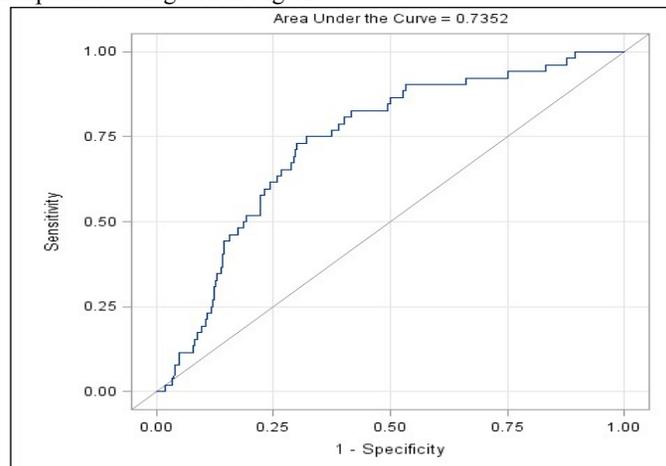


Fig. 12. ROC Curve

5 Conclusion

The use case which is an enigma today as it was yesterday of how to find an effective and efficient solution for failure prediction of servers and applications has been brought under the ambit of a simple framework with the use of analytics and machine learning algorithms. Here the attempt to tackle the problem at the upstream layer is unique in its concept as is the solution provided. As data captured at the infrastructure layer is generic and standard, the solution is far more scalable and has broad based applicability than custom application prediction or related solutions. The linkage and the topography map of configuration item to application is also yet unexplored. Prediction will be done at the CI level and an incident occurring at CI level will affect all the applications related to the CI will be affected. A completely new server data not encountered earlier can be categorized in the specific error categories defined in our solution. The models will be replicated across all the servers based on the issue category they belong to as issues are specific to CI/Server and therefore impacting specific applications, Custom models will be built and subsequently scaled up to all servers using automated model training & implementation methods. Currently, engineers monitor the performance and take actions on such issues in a reactive manner. Ability to correlate these CI/Server metrics and predict issues with a lead time has potential to significantly reduce time spent on incident resolution and improve end user performance. The potential benefits for this solution is also manifold. It will have the operational benefits of reduction in ticket volumes (via early identification and elimination of the cause of ticket), reduction in MTBF (Mean Time Between Failures), increase in MTTR (Mean Time To Resolve), improved availability of CIs (at the infra level the CI will go down even though the app may continue to be up if it is highly available in a cluster form) and Improved SLAs (resolution). In addition to this, there can be strategic and indirect benefits to the businesses as well in the form of Better performance (performance issues when a node goes down – in a cluster), Better capacity planning and Detection of cases where architecture is lacking, and customer needs to revamp – could result in consulting revenue. In the future this work can be extended to consider specific log data of the applications wherever needed and applicable to further enhance the solution.

References

1. Arsenault, R.: Slow is the New Downtime When it Comes to Web Performance. In: Market study report, Aberdeen group (Aug 2015)
2. Oppenkowski, D.: Under Pressure in the Pursuit of Zero Downtime. In: Article in The Data Centre Journal (2016)
3. Codd, P.: Top 10 Software Failures of 2011. In: Business Computing World (Dec 2011)
4. Sahoo, R. K., Oliner, A. J., Rish, I., Gupta, M. Moreira, J.E., Ma, S., Vilalta, R., Sivasubramian, A.: Critical Event Prediction for Proactive Management in Largescale Computer Clusters. pp. 426-435. doi: 10.1145/956750.956799, ISBN: 1-58113-737-0. ACM, Washington, D.C. (2003)

5. Srinivasan, M.K., Sarukesi, K., Revathy, P.: Architectural Design for iCloudIDM Layer-II (iCloudIDM-LII) Subsystem of eCloudIDS Generic Security Framework. DOI: 10.1109/ICACCI.2013.6637432, ISBN: 978-1-4673-6217-7. IEEE, India (2013)
6. Srinivasan, M.K., Sarukesi, K., Revathy, P.: eCloudIDS Tier-1 iCloudIDM Layer-I (iCloudIDM-LI) Subsystem Design and Implementation through User-centric Identity Management Approach for Secure Cloud Computing Environment. DOI: 10.1109/MDM.2013.95. IEEE Computer Society, Italy (2013)
7. Srinivasan, M.K., Sarukesi, K., Rodrigues, P., Saimanoj, M., Revathy, P.: State-of-the-art Cloud Computing Security Taxonomies – A classification of security challenges in the present cloud computing environment. pp. 470-476, DOI: 10.1145/2345396.2345474. ACM, India (Aug 2012)
8. Srinivasan, M.K., Sarukesi, K., Keshava, A., Revathy, P.: eCloudIDS – Design Roadmap for the Architecture of Next-Generation Hybrid Two-Tier Expert Engine-Based IDS for Cloud Computing Environment. In: Thampi S.M., Zomaya A.Y., Strufe T., Alcaraz Calero J.M., Thomas T. (eds) Recent Trends in Computer Networks and Distributed Systems Security. Communications in Computer and Information Science, vol 335. pp. 358-371, doi: 10.1007/978-3-642-34135-9_36. Springer, Berlin, Heidelberg (2012)
9. Srinivasan, M.K., Sarukesi, K., Keshava, A., Revathy, P.: eCloudIDS Tier-1 uX-Engine Subsystem Design and Implementation Using Self-Organizing Map (SOM) for Secure Cloud Computing Environment. In: Thampi S.M., Zomaya A.Y., Strufe T., Alcaraz Calero J.M., Thomas T. (eds) Recent Trends in Computer Networks and Distributed Systems Security. Communications in Computer and Information Science, vol 335. Springer, Berlin, Heidelberg. pp. 432-443, Service Vol. 335, DOI: 10.1007/978-3-642-34135-9_42. (2012)
10. Srinivasan, M.K., Sarukesi, K., Revathy, P.: Design Roadmap for the Phase-I Implementation of Cloud VM/Instance Monitor (CIM-PI) Subsystem of eCloudIDS Security Framework. pp. 520-525, ISBN: 9789351071495. Elsevier (Aug. 2013)
11. Srinivasan, M.K., Revathy, P., Balasundaram, K.: Cloud VM/Instance Monitor Phase-II (CIM-PII) Subsystem of eCloudIDS. In: Saini H., Sayal R., Rawat S. (eds) Innovations in Computer Science and Engineering. Lecture Notes in Networks and Systems, vol 8. Pp. 121-130, ISBN: 978-981-10-3818-1, Doi: https://doi.org/10.1007/978-981-10-3818-1_14. Springer, Singapore (2017)
12. Pellegrini, A., Sanzo, P. D., Avresky, D. R.: A Machine Learning-based Framework for Building Application Failure Prediction Models. In: IEEE International Parallel and Distributed Processing Symposium Workshop. doi: 10.1109/IPDPSW.2015.110, ISBN: 978-1-4673-7684-6, IEE, India (2015)
13. Salfner, F., Lenk, M., Malek, M.: A Survey of Online Failure Prediction Methods. In: ACM Computing Surveys (CSUR). doi: 10.1145/1670679.1670680. ACM, New York, USA (2010)
14. Vobugari, S., Srinivasan, M.K., Somayajulu, D.V.L.N.: Practitioner's guide for building effective complex enterprise architecture in digital transformation: An experience-based industry best practices summary. DOI: 10.1109/ICATCCT.2017.8389158, ISBN: 978-1-5386-1144-9. IEE, India (Dec 2017)
15. Vobugari, S., Srinivasan, M.K., Somayajulu, D.V.L.N.: Roadmap for building effective complex enterprise architecture in digital transformation: An experience-based industry best practices summary. DOI: 10.1109/SmartTechCon.2017.8358640, ISBN: 978-1-5386-0569-1. IEEE, India (Aug 2017)
16. Gill, P., Jain, N., Nagappan, N.: Understanding Network Failures in Data Centers: Measurement, Analysis, and Implications. In: SIGCOMM '11 Proceedings of the ACM SIGCOMM

- 2011, pp.350-361. doi: 10.1145/2018436.2018477, ISBN: 978-1-4503-0797-0. Toronto, Canada (2011)
17. Revathy, P., Mukesh, R.: Analysis of big data security practices. DOI: 10.1109/ICATCCT.2017.8389145, ISBN: 978-1-5386-1144-9. IEEE, India (2017)
 18. Revathy, P., Mukesh, R.: HadoopSec: Sensitivity-aware Secure Data Placement Strategy for Big Data/Hadoop Platform using Prescriptive Analytics. *GSTF Journal on Computing (JoC)*, [S.l.], v. 6, n. 2, p. 5. ISSN 2010-2283. Singapore (2018)
 19. Srinivasan, M.K., Revathy, P.: State-of-the-art Big Data Security Taxonomies. In *Proceedings of the 11th Innovations in Software Engineering Conference (ISEC '18)*. ACM, New York, NY, USA, Article 16, 7 pages. DOI: <https://doi.org/10.1145/3172871.3172886> (2018)
 20. Rosli, M. M., Teo, N. H. I., Yusop, N. S. M., Hasimah, N.: Fault Prediction Model for Web Application Using Genetic Algorithm. In: *IACSIT Press, Singapore* (2011)
 21. Turnbull, D., Alldrin, N.: Failure Prediction in Hardware Systems. In: *CSE221, USA* (2003)
 22. Vaarandi, R.: A Data Clustering Algorithm for Mining Patterns From Event Logs. In: *IEEE*. doi: 10.1109/IPOM.2003.1251233. USA (2003)
 23. Makanju, A.A.O., Zincir-Heywood, A. N., Milios, E. E.: Clustering Event Logs Using Iterative Partitioning. In: *ACM*, pp. 1255-1264, doi: 10.1145/1557019.1557154, ISBN: 978-1-60558-495-9. France (2009)
 24. Murray, J. F., Hughes, G. F., Kreutz-Delgado, K.: Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application. In: *The Journal of Machine Learning Research*, Vol 6, pp. 783-816, ISSN: 1532-4435, EISSN: 1533-7928. ACM (2005)
 25. Balasundaram, K., Srinivasan, M.K., Sarukesi, K.: iReSign – Implementation of Next-Generation Two-Tier Identity Classifier-Based Traffic Sign Recognition System Architecture Using Hybrid Region-Based Shape Representation Techniques. In: Thampi S.M., Zomaya A.Y., Strufe T., Alcaraz Calero J.M., Thomas T. (eds) *Recent Trends in Computer Networks and Distributed Systems Security*. Communications in Computer and Information Science, vol 335. Springer, Berlin, Heidelberg. pp. 408-421, Service Vol. 335, DOI: 10.1007/978-3-642-34135-9_40 (2012)
 26. Balasundaram, K., Srinivasan, M.K., Sarukesi, K.: Implementation of Next-generation Traffic Sign Recognition System with Two-tier Classifier Architecture. pp. 481-487, DOI: 10.1145/2345396.2345476. ACM (Aug. 2012)
 27. Liang, Y., Zhang, Y., Sivasubramaniam, A., Jette, M., Sahoo, R.: BlueGene/L Failure Analysis and Prediction Models. In: *IEE Explore*, doi: 10.1109/DSN.2006.18, ISBN: 0-7695-2607-1. USA (2006)

A New Minimum Spanning Tree Algorithm Constructed in A Backward Fashion

Aozhong Wang¹, Xiaochun Wang², Xia Li Wang³

¹School of Software Engineering, Xi'an Jiaotong University, Xi'an, 710049, CHINA
aozhongwang@stu.xjtu.edu.cn

²School of Software Engineering, Xi'an Jiaotong University, Xi'an, 710049, CHINA
xiaocchunwang@mail.xjtu.edu.cn

³School of Information Engineering, Changan University, Xi'an, 710061, CHINA
xlwang@chd.edu.cn

Abstract Finding a Euclidean minimum spanning tree for a given dataset is a fundamental problem with diverse application domains and many efficient EMST algorithms have been developed. The problem with these algorithms is that today's datasets are often too large to fit into the main memory at once. When working with such massive datasets, memory capacity and, correspondingly, I/O cost, become an important issue. In this paper, we propose a new CPU-based EMST algorithm which can subsequently be adapted to obtain an I/O efficient approximate EMST algorithm and lead to considerable runtime improvements.

Keywords: Minimum spanning tree, Euclidean minimum spanning tree, approximate minimum spanning tree, k nearest neighbors, approximate k nearest neighbors.

1 Introduction

Given an undirected and weighted graph, the problem of minimum spanning tree (MST) is to find a spanning tree such that the total sum of weights is minimized. Being a compact data representation, MST can roughly reflect the intrinsic structure of a data set and has been extensively used in image segmentation [1][2], cluster analysis [3][4][5], classification [6], and manifold learning [7]. Two well-known properties associated with edges in an MST are the cut property and the cycle property. The cut property states that the edge with the smallest weight crossing any 2 partitions of the vertex set must belong to an MST. The cycle property states that the edge with the largest weight in any cycle in a graph can not be in an MST. The first MST algorithm was proposed by Otakar Borůvka in 1926 [8]. Since then, optimal exact MST algorithms, fast and approximate MST algorithms, distributed MST algorithms [9][10][11][12][13][14], and parallel MST algorithms [15][16][17] have been developed.

In today's MST tasks, usually, a set of N d -dimensional data points is given and the problem is commonly solved in the Euclidean setting, giving rise to the so-called Euclidean minimum spanning tree (EMST) problem. In this case, standard MST algorithms, such as Kruskal's [18] and Prim's [19], have a time complexity roughly equal to $O(N^2)$ for a complete graph with $V = N$ vertices and $E = N(N-1)/2$ edges. Thus, the computational cost of building an EMST for a large data set is often the bottle neck of efficiency for many practical applications. Further, for large data sets which can not be loaded into the main memory, external memory-based EMST algorithms have to be developed.

To apply MST-based techniques to large data sets which do not fit into memory, the minimization of the I/O cost is a major concern in the algorithm design. Even though a large number of I/O-efficient graph algorithms have been developed in recent years, they are mainly for MSTs of a given weighted graph. For I/O efficient EMST algorithms where a set of N d -dimensional data points is given, a number of important problems still remain open. In this paper, we present a new CPU-based EMST algorithm. Using the methodology from this in-memory EMST algorithm, we can efficiently compute the longest edges in an EMST for data sets which can not be loaded into the main memory at once, thus overcoming the bottleneck of most external EMST methods. Our objective is to discover all the longest edges with I/O overhead linear to the database size. Throughout this paper, without loss of generality, we assume that no two edges in an EMST have the same weight. To summarize, our first contribution in this paper is a newly proposed CPU-based algorithm for EMST problem. Another important contribution is a new CPU-efficient EMST algorithm with the aid of some indexing structure. A third important contribution is a new I/O efficient approximate EMST algorithm as an adaptation of the proposed in-memory EMST algorithm to cases where datasets can not be loaded into the main memory.

The rest of this paper is organized as follows. In Section 2, we review some existing work on MST algorithms. We next present our proposed approaches in Section 3. In Section 4, an approximate version of the proposed algorithm is developed as an adaptation to cases where datasets can not be loaded into the main memory at once. Finally, conclusions are made and future work is discussed in Section 5.

2 Related Work

2.1 Classic MST Algorithms

Given a connected and weighted graph $G = (E, V)$, Borůvka's MST algorithm begins with each vertex of a graph being a tree. For each consecutive iteration, it selects the shortest edge from a tree to another tree and combines them until all the trees are combined into one tree. The famous Prim's MST algorithm was proposed independently by Jarník [20], Prim [19] and Dijkstra [21] in 1930, 1957 and 1959, respectively. It first arbitrarily selects a vertex as a tree, and then repeatedly adds the shortest edge that connects a new vertex to the tree, until all the vertices are included. Proposed in 1956, Kruskal's algorithm starts with sorting all the edges by their

weights in a non-decreasing order, treats each vertex as a tree, and iteratively combines the trees by adding edges in the sorted order excluding those leading to a cycle until all the trees are combined into one tree [18]. The time complexity of these classic MST algorithms is $O(E \log V)$. If efficient data structures (e.g., Fibonacci heaps [22] and soft heaps [23], etc.) are employed for searching the shortest edges, the computational time can be further reduced.

2.2 Euclidean MST Algorithms

Given a set of N d -dimensional data points, standard Prim's EMST algorithm requires a quadratic running time. To be more efficient, in 1978, Bentley and Friedman proposed to use a kd-tree in Prim's algorithm to enhance the search for the next edge to add to the tree [24]. While their method lacks a formally rigorous bound, it can reach an $O(N \log N)$ running time for most data distributions. In 1985, Preparata and Shamos gave a lower bound for the EMST problem of $\Theta(N \log N)$, which has been the tightest known lower bound [25]. In 1991, Agarwal et al. related the running time of EMST to the bichromatic closest pair (BCP) problem. Given a set of red points and a set of blue points, the bichromatic closest pair is a red point r and a blue point b such that the distance between them is minimized among all such pairs. In 1993, Callahan and Kosaraju proposed Well-Separated Pair Decomposition (WSPD) [26] which forms the basis of most recent EMST algorithms. The WSPD partitions data points into a set of pairs of tree nodes such that the nodes in any pair are farther apart than the diameter of either node. It can be shown that the WSPD has $O(N)$ pairs of nodes, and that the MST is a subset of the edges formed between the closest pair of points in each pair of nodes. In [27], the authors applied WSPD to compute neighbors of components for Boruvka's algorithm by identifying a list of pairs in the WSPD, for which bichromatic closest pair computations are performed to find edges of the MST. However, the constant in the $O(N)$ size of the WSPD grows exponentially with the data dimension and is often very large in practice. In 2010, March et al. presented a new dual-tree algorithm for efficiently computing the EMST [28], which is superficially similar to the method in [26] except that the WSPD is replaced by a new tree data structure. They used adaptive algorithm analysis to prove the tightest (and possibly optimal) runtime bound for the EMST problem to-date. Experiments conducted demonstrated the scalability of their method on astronomical data sets. In 2014, Wang et al. proposed a kNN based fast EMST algorithm [29], which is superficially similar to the method in [26] except that the k-d-tree is replaced by an iDistance indexing structure for kNN search in high-dimensional datasets. The authors argue that the algorithm has an expected $O(N \log N)$ running time, but do not prove this rigorously.

2.3 Approximate EMST algorithms

In addition to exact EMST problem, approximate EMST algorithms have been also proposed. In 1988, Vaidya [30] employed a group of grids to partition a data set into cubical boxes of identical size. For each box, a representative point was determined. Within a cubical box, points were connected to the representative. Any two represent-

atives of two cubical boxes were connected if corresponding edge length was between two specific thresholds. In 1993, Callahan and Kosaraju [26] proposed to utilize well-separated pair decomposition of the data set to extract a sparse graph from the complete graph and then apply an exact MST algorithm to it. More recently, in 2009, Wang et al. [31] employed a divide-and-conquer scheme to construct an approximate EMST, which is superficially similar to WSPD method. Their goal was to detect longest edges in MST at an early stage for clustering. In the same year, Lai et al. [32] proposed another Hilbert curve based approximate EMST algorithm for clustering, which consists of two stages. In the first stage, an approximate MST of a given data set is constructed with Hilbert curve. In the second stage, the data set is partitioned into subsets by measuring the densities of points along the approximate MST with a specified density threshold. In 2015, Zhong et al. proposed a fast two-stage minimum spanning tree algorithm which employs a divide-and-conquer scheme to produce an approximate EMST with theoretical time complexity of $O(N^{1.5})$ [33]. In the first stage, K-means is employed to partition a dataset into $N^{1/2}$ clusters. Then an exact MST algorithm is applied to each cluster and the produced $N^{1/2}$ MSTs are connected to form an approximate MST. In the second stage, the clusters produced in the first stage form $N^{1/2}-1$ neighboring pairs, and the dataset is repartitioned so that the neighboring boundaries of a neighboring pair are put into a cluster. With these $N^{1/2}-1$ clusters, another approximate MST is constructed. Finally, the two approximate MSTs are combined into a graph from which a more accurate MST is generated.

2.4 I/O-efficient MST algorithms

When working with massive graphs, the I/O-efficiency is often the bottleneck. Designing efficient external memory MST algorithms for such graph problems that can thus lead to considerable runtime improvements have received considerable attention. For a general undirected weighted graph $G = (V, E)$, Arge et al. developed an improved I/O-efficient MST algorithm using an efficient contraction algorithm to reduce the number of supervertices to E/B (B is the number of vertices/edges per disk block/page) through $\Theta(\log(VB/E))$ contraction phases [34].

However, these general I/O-efficient MST algorithms designed for massive graphs are insufficient for large metric problems. To this end, Govindarajan et al. presented an external-memory algorithm to compute a well-separated pair decomposition (WSPD) of a given point set S in \mathbb{R}^d in $O(\text{sort}(N))$ I/Os, where N is the number of points in S and $\text{sort}(N)$ denotes the I/O-complexity of sorting N items [35]. The algorithm started by constructing a fair split tree T upon data points recursively. Thereafter, they simulated the internal memory algorithm of Callahan and Kosaraju [26] for constructing pairs in external memory by applying the time-forward processing technique [36]. Unfortunately, they did not directly give I/O efficient solutions to EMST algorithm in their paper.

3 The proposed approach

3.1 An in-memory EMST algorithm

Our CPU-based EMST algorithm follows an observation on the famous Prim's algorithm. To repeat, given a dataset, the Prim's algorithm first arbitrarily selects a vertex as the tree root node, and then repeatedly adds the shortest edge that connects a new vertex to the tree until all the vertices become the tree nodes. Physically, two arrays are used, one to store the index of a data point's parent in the tree and another to store the corresponding edge weight. In other words, for a given dataset of size N , the Prim's algorithm finds an exact EMST by computing the $N(N-1)/2$ number of distances (i.e., distances between every pair of data points) in a forward manner. By the forward manner, we mean that the algorithm takes the first data point in the order they are read into the memory and calculates its distances with the rest of data points not-in-tree yet, and repeat the same process for the next newly selected data point to be in-Tree until the end of dataset is reached.

The Prim's algorithm is simple and elegant given all the data points can be loaded into the memory. However, it has no clue for external memory solutions except that it satisfies the cut and the cycle properties. By contrast, to find $N(N-1)/2$ pairwise distances, there is also a backward manner which forms the core of our proposed algorithm. To illustrate this backward algorithm, we use the sample dataset shown in Fig.1, which consists of 9 data points.

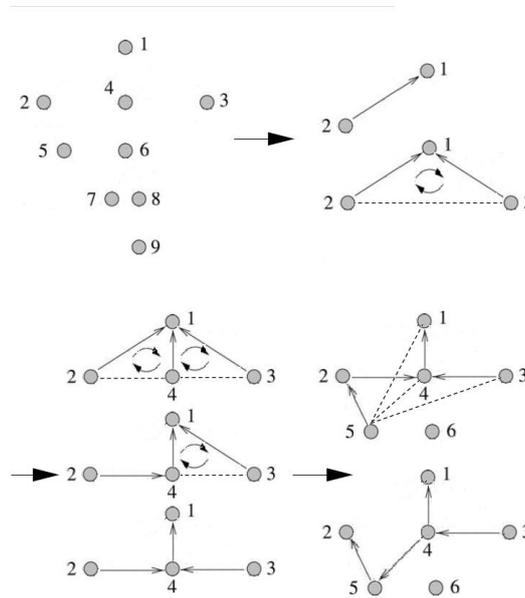


Fig.1. A sample dataset illustrating our new EMST algorithm.

Without loss of generality, our algorithm starts with the first data point in the data sequence, i.e., data point O_1 . Then, for the second data point in the sequence, i.e., O_2 , we calculate its distance with O_1 . They two form a tree with two vertices and one edge, which is denoted by E_{21} . For the third data point in the sequence, i.e., O_3 , we first calculate its distance with O_1 , and a spanning tree is formed. When we next calculate its distance with O_2 , a cycle is formed. By the cycle property of the MST, the longest edge, i.e., E_{32} , should be removed to form a minimum spanning tree. For the fourth data point in the sequence, i.e., O_4 , we calculate its distances with the previous three data points. We first calculate its distances with O_1 , and a spanning tree is formed. We next calculate its distance with O_2 . With the edge E_{42} being added, a cycle is formed upon these three edges i.e., E_{21} , E_{41} and E_{42} , among which E_{21} is the longest edge and should be removed. Finally, we calculate its distance with O_3 . With the edge E_{43} being added, another cycle is formed upon these three edges i.e., E_{31} , E_{41} and E_{43} , among which E_{31} is the longest edge and should be removed by the cycle property of the MST to form a minimum spanning tree. For the fifth data point in the sequence, i.e., O_5 , we first calculate its distances with the previous four data points and form four new edges, E_{51} , E_{52} , E_{53} and E_{54} . Being a little different from the previous four data points, E_{51} is not the shortest edge. To deal with this situation, we first calculate its distances with all the previous data points and sort them in a non-decreasing order. We next add the edge with the smallest distance, which is E_{52} in this case, connecting O_5 to the current minimum spanning tree constructed this far, and no cycle is formed. If the rest edges are added one at a time in the sorted order to the tree, a cycle will always be formed. In each of these cycles, the longest edge should be removed to break the cycle until all edges from O_5 have been added, or the weight of the next edge from O_5 to be added is larger than that of the longest edge currently in the spanning tree. This process continues until the end of the data sequence is reached. By this way, all $N(N-1)/2$ number of distances would be calculated and, by removing the longest edges to avoid the cycles, this process can produce an EMST.

Correspondingly, in comparison to the Prim's EMST algorithm in a forward fashion, there can be a new MST algorithm constructed in this backward fashion. Based on this observation, our backward MST algorithm can be summarized in the following:

1. start with the second data point in the sequence;
2. for each data point i ($1 \leq i \leq N-1$), calculate its distances with all its predecessors, that is, with $0, \dots, i-1$, and sort them in a non-decreasing order;
3. according to the cut property, add the edge with the smallest distance to the MST. For the other edges of the newly calculated $i-1$ ones, if it is added, a cycle is formed. According to the cycle property, the longest edge in the cycle should be removed;
4. $i++$, if $i < N-1$, go to Step 2; otherwise, stop.

The time complexity of our MST algorithm just presented is $O(N^2)$.

4 An I/O Efficient Approximate EMST Algorithm

The above ways of constructing EMSTs are solutions developed for databases which can reside in the main memory as a whole. For today's real-world databases which typically have billions of items with many thousands of fields, resulting in data that range in size into terabytes, it is usually impossible to load them into the main memory at once. Our discussion in this section focuses on large databases that do not fit in memory for which I/O cost becomes a major concern in the algorithm design. To improve the I/O efficiency of solutions of EMST problems, an intuitive way is to reduce the unnecessary scans over the datasets.

The design of a more I/O efficient scheme is motivated by the following observations. According to the working principle of the EMST-based clustering algorithms, a database can be split into partitions by identifying and removing the longest inconsistent edges in the tree. As a result, MST algorithms can be more efficient if the longest edges of an MST can be identified quickly before most of the shorter ones are found. This is because, if the longest edges can be found quickly, the Prim's algorithm can be more efficiently applied to each individual size-reduced cluster. For cases where the number of the longest edges that separate the potential clusters can be much fewer than the number of the shorter edges, this divide-and-conquer scheme will allow us to save the unnecessary scans over the datasets tremendously.

Our objective is to discover all the well-separated clusters with I/O overhead linear to the database size. For datasets too large to fit in main memory, some fetched point should be discarded immediately if its edge weight to some data point is small enough to release some memory space for the rest data points to be read in. Then the question becomes how to define "small enough". Without loss of generality, we assume that the data points should have been stored in a random order, as can be easily achieved by a simple randomization process, and data points form clusters that are well separated. By well separated, we mean the intra-cluster distances are much smaller than inter-cluster distances.

Assuming that each disk page can accommodate b data points, and that the memory has a pages, to start our analysis, let us randomly sample $s = b \cdot a$ data points from the database. In particular, since data points are stored in a random order, we just read in the first $s = b \cdot a$ data points encountered in the scan and construct an EMST using the algorithm presented in Section III. In this local MST, these $s-1$ edges provide the upper bounds for their true $s-1$ edges in the exact true MST on the whole dataset. Among the s data points currently in the main memory, the data points to be discarded should satisfy the following conditions:

- A. the discarding should proceed from the largest cluster through to the smallest cluster so that the small clusters will not be under represented;
- B. they should be discarded in a non-decreasing order of their edge weights;
- C. if the data point to be discarded next is connected to a data point just discarded before loading the next new page of data points and the consecutive

discardings will not result in an edge weight loss that is larger than the b -th smallest discarded edge weight.

As long as s is large enough, b of the s data points currently in the main memory can be discarded to release memory for new b data points to come in, however, without altering the cluster-belonging they are supposed to.

With these observations in mind, we can intuitively think of employing a divide-and-conquer technique to achieve the improvement. Typically, a divide-and-conquer paradigm generally consists of three steps:

1. Divide step. The problem is divided into a collection of subproblems that are similar to the original problem in type but smaller in size.
2. Conquer step. The subproblems are addressed separately, and corresponding subresults are produced.
3. Combine step. The subresults are combined to form the final result of the problem.

In the light of the above divide-and-conquer paradigm, given a loose estimate of minimum and maximum numbers of data items in each cluster, we devise an I/O efficient approximate EMST method as follows:

1. read $s=b$ data points from the database in and construct an EMST using the method presented in Section III;
2. discard b data points according to Conditions A、B and C;
3. read another b data points in;
4. for each newly loaded data point i ($0 \leq i \leq b-1$), calculate its distances with all the data points currently in the memory, and sort them in a non-decreasing order;
5. according to the cut property, add the edge with the smallest distance to the approximate MST. For the rest edges of the newly calculated ones, if it is added, a cycle is formed. According to the cycle property, the longest edge in the cycle should be removed;
6. $i++$, if $i < b-1$, go to Step 4; if $i == b-1$, go to Step 7;
7. discard b data points according to Conditions A, B and C; if the end of file is reached, go to Step 8, otherwise, go to Step 3;
8. according to the approximate EMST just constructed, the longest edges can be identified, if the database is scanned at least twice and the weights of these longest edges do not change, go to Step 9, otherwise, go to Step 1.
9. the longest edges are removed to form partitions; then if needed, our algorithm can be more efficiently applied to each individual size-reduced cluster.

Essentially, given a data set, in the first scan, our algorithm starts with s randomly data points loaded into the main memory, creates an EMST composed with these s

data points. As each page of data object is discarded from and loaded into the main memory in the scanning process, b objects associated with the shortest edges and satisfying Conditions A, B and C are discarded to provide space for another b data points to be loaded into the main memory, our I/O efficient algorithm continues the spanning tree growing process until the end of dataset. In the second scan, each of the s data points retained in the main memory can be updated with a smaller value by a data point it does not meet in the first scan. Such a strategy ensures that points that are close to each other in space are likely to meet with a higher possibility directly depending on the memory size. However, because any data point is closer to its k -nearest neighbors in the main memory (which may not be its true k -nearest neighbors), they can produce false longest edges. Fortunately, such possibilities can be greatly reduced by multiple scans of the database. The upper bound of the expected number of scans through the database to locate longest edges is closely related to the main memory size s and the page size b . Now the memory size represents the possibility that the data points find their approximate nearest neighbors.

For our purpose, after the initial scan, a spanning tree is constructed and each data item in the tree has already had a distance. During the subsequent process, the data items associated with longest edges will have distance computations over multiple scans. Further, we would be more interested in those data points whose distance upper bounds are potential longest edge candidates than those whose distance upper bounds are too small to be given any further consideration. Therefore, after the initialization, we can compute the mean and the standard deviation of the edge weights and use their sum as a threshold value. Then in each step of the spanning tree updates by discarding a page of data points and loading another page, before we calculate a data item to further reduce its nearest neighbor edge weight, if its current distance upper bound is smaller than the threshold, we can ignore it so as to save some spaces. Only when the distance upper bound is larger than the threshold, do we carry out a distance computation to classify it to its closer nearest neighbor and make the corresponding update. In other words, we give the potential longest edge candidates more attention. However, this will give the potential longest edge candidates more opportunities to have a smaller distance upper bound. Only when all the current largest distance values converge to the same value at the same location and have edge weights significantly larger than the average edge weights of two neighboring edges can the procedure stop as completing the verification of the current longest edge candidate. Such a procedure is an efficient method to implement the cycle property. To summarize, we believe that the advantage of our algorithm is that, after multiple scans, by our greedy strategy, each point will be very close to its true nearest neighbor in the data set.

5 Conclusions

In this paper, we propose a new CPU-based EMST algorithm which can subsequently be adapted to obtain an I/O efficient approximate EMST algorithm, which is designed

particularly for datasets that can not be loaded into main memory at once. The CPU-based EMST algorithm works by utilizing the cut and the cycle properties of the MST. The I/O efficient approximate EMST algorithm works by identifying and removing data points associated with smallest edge weights from the main memory so as to release memory space for data points associated with the relatively few largest edge weights to retain in the memory. Our solution is general and can be applied to large high-dimensional datasets. In our future work, we would like to conduct experiments on larger-sized high dimensional datasets to demonstrate the efficiency of the proposed method.

Acknowledgment

The authors would like to thank the Chinese National Science Foundation for its valuable support of this work under award 61473220.

References

1. L. An, Q.S. Xiang, and S. Chavez. A Fast Implementation of the Minimum Spanning Tree Method for Phase Unwrapping. *IEEE Transactions on Medical Imaging*, 19, 8 (2000), 805-808.
2. Y. Xu, and E.C. Uberbacher. 1997. 2D Image Segmentation Using Minimum Spanning Trees. *Image and Vision Computing*, 15 (1997) 47-57.
3. C.T. Zahn. 1971. Graph-theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computer*, C20 (1971) 68-86.
4. Y. Xu, V. Olman and D. Xu. 2002. Clustering Gene Expression Data Using a Graph-theoretic Approach: an Application of Minimum Spanning Trees. *Bioinformatics*, 18 (4) (2002) 536-545.
5. C. Zhong, D. Miao, and R. Wang. 2010. A Graph-theoretical Clustering Method Based on Two Rounds of Minimum Spanning Trees. *Pattern Recognition*, 43(3) (2010) 752-766.
6. P. Juszczak, D.M.J. Tax, E. Pełkalska and R.P.W. Duin. 2009. Minimum Spanning Tree Based One-class Classifier. *Neurocomputing*, 72 (2009) 1859-1869.
7. C L. Yang. 2005. Building k Edge-disjoint Spanning Trees of Minimum Total Length for Isometric Data Embedding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10) (2005) 1680-1683.
8. O.Borůvka. 1926. O jistém problému minimálním (About a Certain Minimal Problem). *Práce moravské pěstirodovědecké společnosti v Brně. III* (1926) 37-58 (in Czech with German summary).
9. R.G. Gallager, P.A. Humblet, and P. M. Spira, 1983. A Distributed Algorithm for Minimum-weight Spanning Trees, *ACM Transactions on Programming Languages and Systems*, 5, 1 (January 1983) 66-77.
10. B. Awerbuch. 1987. Optimal Distributed Algorithms for Minimum Weight Spanning Tree, Counting, Leader Election, and Related Problems, In *Proceedings of the 19th ACM Symposium on Theory of Computing (STOC'87)*, New York City, New York, May 1987.
11. J. Garay, S. Kutten and D. Peleg, 1993. A Sub-Linear Time Distributed Algorithm for Minimum-Weight Spanning Trees (Extended Abstract), In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS'93)*, 1993.

12. S. Kutten and D. Peleg. 1998. Fast Distributed Construction of Small-Dominating Sets and Applications, *Journal of Algorithms*, 28, 1 (July 1998) 40-66.
13. D. Peleg and V. Rubinovitch. 2000. A Near Tight Lower Bound on the Time Complexity of Distributed Minimum Spanning Tree Construction, *SIAM Journal on Computing*, 30, 5 (2000) 1427-1442
14. M. Khan and G. Pandurangan. 2008. Fast Distributed Approximation Algorithm for Minimum Spanning Trees, *Distributed Computing*, 20, 6 (2008) 391-402.
15. K.W. Chong, Y. Han and Lam, Tak Wah. 2001. Concurrent Threads and Optimal Parallel Minimum Spanning Trees Algorithm, *Journal of the Association for Computing Machinery*, 48,2 (2001) 297-323.
16. S. Pettie and V. Ramachandran. 2002. A Randomized Time-work Optimal Parallel Algorithm for Finding a Minimum Spanning Forest. *SIAM Journal on Computing*, 31 ,6 (2002) 1879-1895.
17. D.A. Bader and G. Cong. 2006. Fast Shared-memory Algorithms for Computing the Minimum Spanning Forest of Sparse Graphs, *Journal of Parallel and Distributed Computing*, 66, 11 (2006) 1366-1378.
18. J. B. Kruskal. 1956. On the Shortest Spanning Subtree of A Graph and the Traveling Salesman Problem. In *Proceedings of the American Mathematical Society*. 7, 1 (Feb., 1956), 48-50.
19. R.C. Prim. 1957. Shortest Connection Networks and Some Generalizations. *Bell System Technical Journal*, 36 (1957) 567-574.
20. V. Jarník. 1930. O jistém problému minimálním (About a Certain Minimal Problem). *Práce moravské pěstirodovědecké společnosti v Brně VI* (1930) 57-63 (in Czech).
21. E.W. Dijkstra. 1959. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik I* (1) (1959) 269-271.
22. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein. 2001. *Introduction to Algorithms* (3rd ed.). MIT Press and McGraw-Hill. Chapter 20: Fibonacci Heaps. ISBN 0-262-03293-7.
23. H. Kaplan and U. Zwick. 2009. A simpler implementation and analysis of Chazelle's soft heaps. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics*. (2009) 477-485.
24. J. Bentley and J. Friedman. 1978. Fast Algorithms for Constructing Minimal Spanning Trees in Coordinate Spaces. *IEEE Transactions on Computers*. 27 (1978) 97-105.
25. F. P. Preparata and M. I. 1985. Shamos. *Computational Geometry*. Springer-Verlag, New York, 1985
26. P. Callahan, and S. Kosaraju. 1993. Faster Algorithms for Some Geometric Graph Problems in Higher Dimensions. In *Proceedings of 4th Annual ACM-SIAM Symposium on Discrete Algorithms*, (1993), 291-300.
27. G. Narasimhan, M. Zachariasen and J. Zhu. 2000. Experiments With Computing Geometric Minimum Spanning Trees. In *Proceedings of ALENEX'00*, (2000) 183-196.
28. W.B. March, P. Ram, and A.G. Gray. 2010. Fast Euclidean Minimum Spanning Tree: Algorithm, Analysis, and Applications. In *Proceedings of 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*, Washington (2010) 603-612.
29. X Wang, X L. Wang and J Zhu, A New Fast Minimum Spanning Tree-Based Clustering Technique, In *Proceedings of 2014 IEEE International Conference on Data Mining Workshop*, Shenzhen, China, 14-14 Dec. 2014, 1053-1060.
30. P.M. Vaidya. 1988. Minimum Spanning Trees in k-dimensional Space. *SIAM Journal on Computing*, 17 (3) (1988) 572-582.

31. X. Wang, X.L. Wang and D.M. Wilkes. 2009. A Divide-and-Conquer Approach for Minimum Spanning Tree-based Clustering. *IEEE Transactions Knowledge and Data Engineering*, 21, 7 (2009) 945-958.
32. C. Lai, T. Rafa and D.E. Nelson. 2009. Approximate Minimum Spanning Tree Clustering in High-dimensional Space. *Intelligent Data Analysis*. 13 (2009) 575-597.
33. C. Zhong, M. Malinen, D. Miao and P. Fränti. 2015. A Fast Minimum Spanning Tree Algorithm Based on K-means, *Information Sciences*, 295 (C) (2015) 1-17.
34. L. Arge, G.S. Brodal and L. Toma. 2004. On External-memory MST, SSSP and Multi-way Planar Graph Separation, *Journal of Algorithms* 53, 2 (2004) 186–206.
35. S. Govindarajan, T. Lukovszki, A. Maheshwari, and N. Zeh. 2000. I/O Efficient Well-separated Pair Decomposition and its Applications. In *Proceedings of the 8th European Symposium on Algorithms*, (2000) 220–231.
36. Y.-J. Chiang, M. T. Goodrich, E. F. Grove, R. Tamassia, D. E. Vengroff, and J. S. Vitter. External-memory Graph Algorithms. In *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms*, (January 1995) 139–149.

Analysis of Linear and Non-Linear Classifiers in Imbalanced Data to Predict Diabetes Induced Complications

Tahsinur Rahman, Aniq Zaida Khanom, Sheikh Mastura Farzana, Sharowar Md. Shahriar Khan and Dr. Md. Ashraful Alam

Abstract. This paper presents a comparison of linear and non-linear classifiers and their results in predicting health complications of the Kidney and Heart induced by Diabetes Mellitus based on an imbalanced dataset. Over time Diabetes damages various organs in the body- primarily Kidney, Eyes, Heart and Brain. The onset of these complications can be hard to prevent unless a person is monitored closely. This proposed model uses a time series data of one year that contains 164 features of 779 T2DM patients to predict the risk of Nephropathy and Cardiovascular disease. Methods such as Logistic Regression, Support Vector Machines, Naïve Bayes, Decision Tree and Random Forest have been used to predict the probability of developing the complications. Random Forest produces the best results with 85 trees for Nephropathy with F1 score 0.75. Logistic Regression without oversampling gives the best results for Cardiovascular disease when C is 0.3, F1 score is 0.54.

Keywords: Imbalanced dataset · Complications prediction · Logistic Regression · Random Forest · Oversampling.

1 Introduction

Clinical databases store large amounts of information about patients and their medical conditions that can be used to discover relationships and patterns among clinical and pathological data using data mining techniques [1]. These can be used for early diagnosis by understanding the progression and features of the disease. In most clinical databases, disease cases are fairly rare as compared with the normal populations, hence creating an imbalance. Diabetes Mellitus is one such example of health condition. Several Machine Learning based models exist that deal with Diabetes Mellitus [2]. However, most of these systems only predict the probability of a person having Diabetes in the near future. According to IDF Atlas published in 2017, there are around 424.9 million Diabetes patients around the world aged from 20-79 years, of whom 95% suffer from Type 2 Diabetes Mellitus (T2DM). It is predicted that the number will increase to 628.6 million by 2045 [3]. Diabetes Mellitus can induce other complications like Nephropathy, Cardiovascular disease, Retinopathy and Diabetic Foot disease [4]. In 2017 alone, 4 million people died all around the world due to Diabetes related complications. This paper proposes a prediction model built using linear

and non-linear classifiers that can predict the probability of Nephropathy and Cardiovascular disease onset in T2DM patients. An embedded pipeline has been applied upon an imbalanced dataset containing pathological test results of 779 patients, exploring the effects of imbalance of data in prediction models. Additionally, it is different to the conventional Diabetes predicting systems since it emphasizes on predicting Diabetes related complications. There is a scope to introduce a complete system that can correctly predict onset of complications caused by T2DM using Machine Learning techniques.

2 Related Work

Researchers have found various new aspects in past years discussing the factors which assist Diabetes development and its impact on health that causes different types of complications [5]. Recent development in Deep Learning systems have made it possible to develop a system that diagnoses retinal diseases [6]. AI and Machine Learning are assets that has the ability to help doctors overcome their limitations by improving prediction and diagnosis of diseases [7]. In July 2018, an AI system beat 15 Chinese doctors in a tumor detection competition which is the latest among several similar examples [8]. In several papers, it is described how in the case of diseases like Cancer, Mental health and Cardiovascular conditions, scientists are applying predictive algorithms with satisfactory accuracy [9]. Abundance of Machine Learning models exist that can diagnose if a person has Diabetes or is prone to develop Diabetes [10]. However, models that can predict onset of Diabetes induced health complications are rarer. One such model used a data mining pipeline to predict T2DM related complications using Electronic Health Record data [11]. Dagiati et al. (2018) predicted complications such as Neuropathy, Nephropathy and Retinopathy with an accuracy of up to 0.83 [11]. The researchers used few features to conduct the research which reduced complexity of the model. Furthermore, the model did not predict the complication of Cardiovascular diseases. A recently conducted research presented a new data-driven approach to predict diabetic complications [12]. Another notable work, done by BH Cho et al. in 2008 explains a model which used feature selection to predict diabetic Nephropathy [13]. Tanaka et al. (2013) used a statistical model called Cox regression model to predict the risk of various diabetic complications including Cardiovascular diseases [14]. Nonetheless, apart from these researches, there isn't any significant work that has previously been done regarding the prediction of Diabetes induced Cardiovascular diseases especially using Machine Learning. One of the conceivable reasons might be the lack of information and proper variables since heart complications are related to a lot of other factors and variables of the human body [15]. Another major problem is the presence of imbalanced data in different domains of machine learning and data mining especially in medical science [1, 16]. Mazurowski (2008) presented a paper showing that the classifier performance may deteriorate even with a modest class imbalance in the training data [17]. Even in the field of Diabetes Complication prediction, class imbalance is a concerning issue [11]. To solve the class imbalance problem, data level and algorithm level approaches are usually

taken [18]. Learning from imbalanced data has been a focus of intense research and continuous development for more than two decades. With the expansion of machine learning and data mining, a deeper insight into the nature of imbalanced learning has been gained, adding to the new emerging challenges as well [19]. So even though considering all of these variables can increase the complexity of the model, doing so is very difficult. Regardless, in this paper, these limitations were addressed and focus was put on the imbalance of the dataset in the prediction of risk of Cardiovascular disease alongside Nephropathy.

3 Methodology of Work

The work presented in this paper can be broadly divided into four sections; i) Dataset Collection, ii) Data Pre-processing, iii) Training and Testing Model and iv) Improving the best ML classifier. In dataset collection section, an open source dataset was used. Data imputation, Feature Scaling and Categorical Variable Conversion were done in Data pre-processing. A set of six algorithms were then implemented to find the best possible outcome. After obtaining the best classifier for Nephropathy and Cardiovascular disease, they were further improved by changing a few parameters. Additionally, in case of Cardiovascular disease, oversampling was also implemented in a particular set of experiments discussed later in the paper. Figure 1 represents the methodology of the work done in this paper. LR, SVM, NB, DT and RF stands for Logistic Regression, Support Vector Machines, Naïve Bayes, Decision Tree and Random Forest respectively.

3.1 Dataset

The dataset used in the model is from an open-label, central registration, multi-center, prospective observational study that was conducted at the Tokyo Women's Medical University Hospital in collaboration with 69 other institutions in Japan [20]. It was retrieved from a loyalty free dataset sharing platform and consisted of 779 instances and 164 variables. Out of the 164 variables, 24 were categorical variables and the rest were numerical variables. Many features are time series data of 1 year at several time differences.

Target Variable Although the dataset contains many features there were no feature that could be used to determine if a patient is actually at risk of developing either Nephropathy or Cardiovascular disease. Additionally, target variables are needed to measure model performance. Hence, target variables, Risk of Nephropathy and Risk of Cardiovascular Disease have been synthesized. Methods used by former medical researchers have been re applied for both variables. For the case of Risk of Nephropathy, the conditions (i) Urinary albumin/creatinine ratio greater than 30, (ii) No history of previous renal complication and (iii) GFR less than 60 mL.min-1 have been considered and positive in all of them has been labeled as '1' [21, 22]. On the other hand, for Risk of

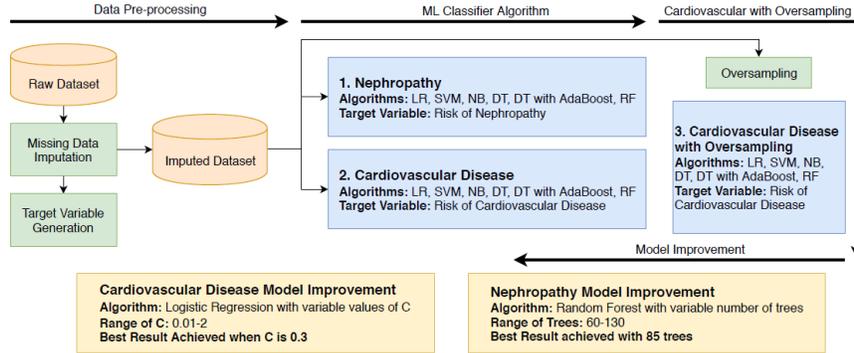


Fig. 1: Methodology of Work

Cardiovascular Disease, (i) History of Diabetes Mellitus, (ii) History of Hypertension, (iii) Hypertriglyceridemia and (iv) History of Dyslipidemia have been considered and positive in all of them has been labeled as '1' [23].

Data Imbalance One of the highlighted features of the above mentioned dataset is that it does not have uniform positive and negative class distribution. For Nephropathy, approximately 25% patients of the dataset were identified as Kidney patients which is an imbalance ratio of 1:4. However, in the case of Cardiovascular disease, the patients having the disease were only about 10% of the set which is an imbalance ratio of 1:10. According to Wong (2009), a ratio as low as 1: 10 can be tough to deal with and is inadequate for building a good model in most cases [1]. To overcome the issue of imbalance, two different approaches are usually used: Data-level approach and Algorithm-level approach. For data-level approach, oversampling was applied on the dataset for Cardiovascular Disease prediction and the performance of the algorithms were compared for balanced and imbalanced dataset. For algorithm-level approach, several classifiers were used including ensemble methods like AdaBoost and Random Forest. The parameters of these classifiers were also tweaked to improve the algorithm and find out the best combination for the imbalanced dataset. Oversampling was implemented by applying the SMOTE algorithm on the training set to make the class ratio 50:50 [25].

3.2 Data Pre-processing

Data Imputation The dataset had values missing at random (MAR) so the missing value of a variable can be predicted from the other variables making it suitable for imputation. Overall, the maximum number of missing columns for a particular instance was 145 and the minimum was 0. Even though traditional methods of imputation like replacing with mean, replacing with 0 and deleting entire instances were considered, they were found to be inadequate. Instead imputation was done using an algorithm called missForest with 100 trees [24]. The model yields an out-of-bag (OOB) imputation error estimate which

consists of the NRMSE (Normalized Root Mean Squared Error) for the continuous numeric variables and PFC (Proportion of Falsely Classified) for categorical variables. NRMSE in this case was 17.89% and PFC was 12.19%.

Feature Scaling and Categorical Variable Conversion Standardization was used as the feature scaling technique. It is particularly important since some classification algorithms like SVM and Logistic Regression do not perform well on unscaled data since variables with higher and lower scale are going to be treated differently. Another important part of the model is the conversion of the categorical variables to their numerical counterparts to avoid misinterpretation of information from the data. This is implemented by creating dummy variables for each class present in every categorical variable. To avoid the occurrence of Dummy Variable Trap, n-1 dummy variables were created for a categorical variable with n different values.

3.3 Algorithms and Evaluation Metrics

In this paper, several classification algorithms were used for comparison, in order to find the best one for each problem. The linear algorithms Logistic Regression, Support Vectors Machines and Naïve Bayes along with non-linear classifiers Decision Tree and Random Forest were implemented. Additionally, AdaBoost was used for boosting. The primary parameter for logistic regression was L2 regularization with the inverse of regularization strength(C) value being 1. On the other hand for SVM, the RBF kernel was used with the penalty parameter C of the error term being 1. Gaussian Naïve Bayes is the specific algorithm that was applied in the case of Naïve Bayes. CART was the Decision Tree and was utilized with the split criterion based on Gini impurity as one of the non-linear classifiers. Furthermore, AdaBoost was employed using the SAMME.R real boosting algorithm with maximum number of estimators being 50. Finally, the Random Forest Classifier had 80 trees as the primary number of estimators. The data is imbalanced for both cases, with a prevalence (π) of disease being 0.249 for nephropathy and 0.096 for cardiovascular disease. Hence, accuracy is a poor evaluation metric in this case [26]. For the performance evaluation of classifiers, AUC score, Average Precision (AP) and F1 Score are usually better metrics. Since it can sort models by overall performance, the AUC is considered more in model assessments. However, AUC score masks poor performance if the dataset is imbalanced [27]. This is especially the case when the value of π is less than 0.1 [28]. Yuan et al. (2015) have shown this by demonstrating a drastic difference between the AUC and AP score when π changes from 0.5 to 0.1 and less [28]. Therefore, while AUC is a good metric for Nephropathy, it does not work well in case of Cardiovascular Disease. F1 score takes both precision and recall into account by taking their harmonic mean with a high score indicating that the model performs better on the positive class [29]. F1 Score was prioritized in this paper to evaluate the performance of the algorithms with individual considerations to the precision and recall also. In some medical papers, Average Precision (AP) is considered to evaluate results.

3.4 Training

Dataset

In this paper, cross-validation was done by splitting the overall dataset into training and test set with a ratio of 70:30. For Nephropathy, the test set consisted of 216 instances, 162 belonging to the false class (does not have Kidney complications) and 54 belonging to the true class (has Kidney complications). In the case of Cardiovascular disease, the test set consisted of 234 instances with 211 belonging to the false class (does not have Cardiovascular complications) and 23 belonging to the true class (has Cardiovascular complications).

4 Results

The linear classifiers used were Logistic Regression (LR), Support Vector Machines (SVM) and Naïve Bayes (NB) and the non-linear classifiers used were Decision Tree and Random Forest were implemented. AdaBoost was used since it is capable of handling the class imbalance problem. Oversampling using SMOTE was applied on Cardiovascular Disease. This paper discusses all the models experimented in three subsections: i. Nephropathy, ii. Cardiovascular without Oversampling and iii. Cardiovascular with Oversampling. Each results table contain Accuracy, Precision, Recall, Average Precision (AP), F1 Score and AUC Score in regard of all algorithms that are being applied.

4.1 Nephropathy

Table 1 shows scores of different performance metrics for all the algorithms for Nephropathy.

Table 1: Nephropathy Scores

	Logistic Regression (LR)	Support Vector Machines(SVM)	Naïve Bayes (NB)	Decision Tree (DT)	Decision Tree with AdaBoost(Ada)	Random Forest (RF)
Accuracy	0.79	0.83	0.81	0.84	0.82	0.88
Precision	0.56	0.91	0.61	0.66	0.61	0.87
Recall	0.67	0.35	0.61	0.74	0.80	0.63
AP	0.63	0.75	0.61	0.55	0.54	0.86
F1 Score	0.61	0.51	0.61	0.69	0.69	0.73
AUC	0.77	0.87	0.83	0.81	0.82	0.92

Figure 2(a) is the graphical representation of the values of different algorithms' recall, precision, AP and F1 score. The difference of a specific metric for each algorithm can be signified. Additionally, Figure 2(b) represents the AUC scores for each algorithm, showing the best and the worst algorithms based on AUC score.

Linear Classifiers: Even though LR and SVM are both linear classifiers, a significant difference can be observed in the performance metrics. The precision is higher in SVM, while the recall and F1 Score is better for LR. So while SVM

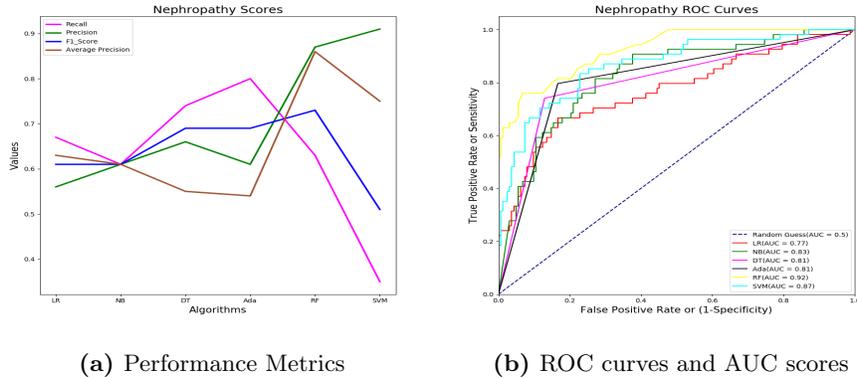


Fig. 2: Prediction of Nephropathy (Kidney Disease)

has a higher AUC score of 0.87, it is disregarded. Hence, between these two algorithms, for the prediction of Nephropathy, Logistic Regression is a better option with recall of 0.67 and F1 Score 0.61 since it classifies the positive class better. Naïve Bayes has a result with the same precision and recall, 0.61. Hence, the AP and F1 Score are also 0.61. Further, it can be noticed that the AUC score of NB are 0.83, which is also quite high. However, comparing to LR and SVM, the performance of NB cannot be said to be significantly better as none of the results are better than the former algorithms.

Non-linear Classifiers: Decision Tree was used in two ways; with and without the boosting. AdaBoost was used as the booster. The scores in both these cases are similar, with the accuracy, precision and AP falling slightly for AdaBoost. However, the recall increases to 0.80 for boosted DT while F1 score remains the same at 0.69. Comparing to the linear classifiers, DT has a better overall performance. On the other hand, among all the algorithms, Random Forest has the best AP (0.86), F1 score (0.73) and AUC score (0.92). For disease prediction, recall is an important metric. In this case, the recall is not so high, which is 0.63. However, the accuracy and precision scores of RF is also above 0.85.

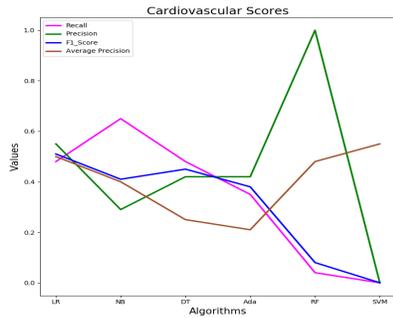
For Nephropathy the best result was given by Random Forest(RF) when taking into account all the performance metrics. Oversampling was not considered for Nephropathy since the data was only slightly imbalanced with 25% belonging to the positive class. This imbalance is countered by RF by aggregating several decision trees together. Hence, satisfactory results were obtained without the introduction of oversampling. Finally it can be observed that for Nephropathy, non-linear classifiers work better than linear classifiers as DT, DT AdaBoost and RF gives better F1 Score than LR, SVM and NB. Since the classes are divided by a non-linear boundary, linear classifiers fail to perform as well as their non-linear counterparts.

4.2 Cardiovascular without Oversampling

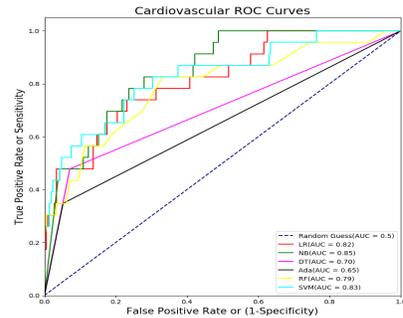
Table 2 shows scores of different performance metrics for all the algorithms for Cardiovascular Disease. Figure 3(a) represents the values of different algorithms' performance metrics and Figure 3(b) represents the AUC scores for each algorithm.

Table 2: Cardiovascular Disease Scores (Without Oversampling)

	Logistic Regression (LR)	Support Vector Machines(SVM)	Naïve Bayes (NB)	Decision Tree (DT)	Decision Tree with AdaBoost(Ada)	Random Forest (RF)
Accuracy	0.91	0.90	0.81	0.89	0.89	0.91
Precision	0.55	NaN	0.29	0.42	0.42	1.00
Recall	0.48	0.00	0.65	0.48	0.35	0.04
AP	0.50	0.55	0.40	0.25	0.21	0.48
F1 Score	0.51	0.00	0.41	0.45	0.38	0.08
AUC	0.82	0.83	0.84	0.70	0.65	0.79



(a) Performance Metrics



(b) ROC curves and AUC scores

Fig. 3: Prediction of Cardiovascular(Heart) Disease

Linear Classifiers: In the prediction of Cardiovascular disease, LR has a high accuracy of 0.91 and AUC score of 0.82, while having average values for precision (0.55), AP (0.5), recall (0.48) and F1 score (0.51). The results of SVM though are much unexpected as the accuracy and AUC score are high. While on the other hand, the recall is zero and the precision is undefined. Usually, SVM works well for moderate imbalance and performance decreases when it goes towards high imbalance but this concept can vary with the nature of the dataset [30, 1]. However, for this particular dataset, even though it is moderately imbalanced, F1 score is zero which means SVM completely fails to predict the positive class. The aforementioned values of recall and precision corresponds to only one particular threshold. For other thresholds values, however, the precision and recall are defined giving SVM an average precision of 0.55. Comparing, in this case, LR is clearly the better algorithm. Compared to LR, Naïve Bayes has a higher recall of 0.65 but a lower F1 score of 0.41. Moreover, all the other linear evaluators have

lower value of F1 Score than that of LR. Hence, among all the linear classifiers, Logistic Regression gives the best prediction model.

Non-linear Classifiers: Similar to Nephropathy, DT was implemented in two ways; with and without AdaBoost. The scores in both these cases are similar, though a fall of recall, AP, F1 score and AUC score can be noticed. There is no increase in any of the metrics and the accuracy and precision is same for both cases being 0.89 and 0.42 respectively. Comparing to LR, unlike in Nephropathy, all the evaluators have a poorer score, with a steep decrease in AP and F1 score. In case of RF, it can be noticed that the accuracy is very high (0.91). However, the recall and F1 score is very low with only 0.04 and 0.08 respectively. Therefore, even though the precision is 1, it is a very bad predictor for this case.

However, even though LR is the best predictor for Cardiovascular disease, the precision, recall, AP and F1 score are all very low being close to 0.5 in all cases. The recall is of particular interest in this paper since all the patients who have a chance of developing complications needs to be predicted correctly. The recall in most cases are poor since the prevalence of heart patients is less in the dataset. Further, it is seen that the AUC score is not a suitable metric when the prevalence of positive class is extremely low [28]. So in the comparison of the classifiers for this case, more importance was given to the AP score than AUC score. To overcome these issues, oversampling has been used, and the results are discussed in the next subsection.

4.3 Cardiovascular with Oversampling

Table 3 shows scores of different performance metrics for all the algorithms for Cardiovascular Disease. Figure 4(a) represents the values of different algorithms' performance metrics and Figure 4(b) represents the AUC scores for each algorithm.

Table 3: Cardiovascular Disease Scores (With Oversampling)

	Logistic Regression (LR)	Support Vector Machines(SVM)	Naïve Bayes (NB)	Decision Tree (DT)	Decision Tree with AdaBoost(Ada)	Random Forest (RF)
Accuracy	0.58	0.91	0.78	0.87	0.87	0.91
Precision	0.31	0.66	0.28	0.36	0.37	0.56
Recall	0.47	0.17	0.78	0.44	0.44	0.22
AP	0.48	0.52	0.49	0.21	0.22	0.40
F1 Score	0.38	0.28	0.41	0.39	0.40	0.31
AUC	0.80	0.85	0.86	0.68	0.68	0.85

Linear Classifiers: Oversampling the data leads to a decrease in performance for LR shown by a decrease in value of all the metrics. However in case of SVM, there is an increase in the value of precision and recall which increases from zero and hence the F1 score also increases. Since oversampling leads to the prevalence being close to 0.5, the AUC can be considered to measure performance. However,

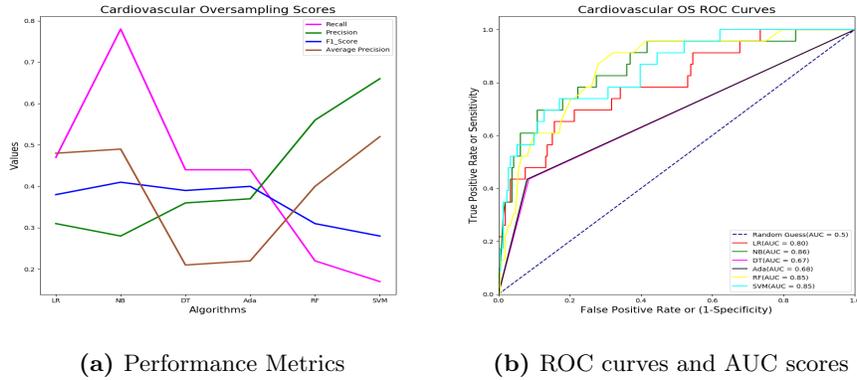


Fig. 4: Prediction of Cardiovascular (Heart) Disease with Oversampled Dataset even though SVM has a better accuracy and AUC score than LR, its recall is still lower. The precision is higher for SVM but a low recall leads to a F1 score of 0.28 which is lower than LR's F1 score of 0.38. Hence, even in this case LR is a better algorithm than SVM. As mentioned before, the performance of SVM degrades as class imbalance increases. Hence when oversampling is applied SVM performs better as class imbalance problem is resolved. For NB oversampling gives a better performance since AP score increases while keeping the F1 score same. The main observation is that the recall increases significantly to 0.78 which is highest across all combinations for Cardiovascular disease. Since NB has a better AUC and F1 score than LR, it is considered to be better of the two and hence the best linear classifier after oversampling.

Non-linear Classifiers: Just like without oversampling, in oversampling the scores for both DT and Boosted DT are similar. Though Boosted DT has a F1 score which is 0.01 greater than DT, making it slightly better. Overall, oversampling improves the performance of Boosted DT with F1 score increasing from 0.38 to 0.40. For DT, the opposite happens with the performance deteriorating when oversampled. The F1 score decreases from 0.45 to 0.39. Hence it can be deduced that oversampling works for AdaBoosted DT but not DT. The AP score is still poor for both and since AP is a better measure for the Cardiovascular case, it can be deduced that both classifiers fail. This is due to the linear characteristic of the data for Cardiovascular Disease. Again for RF with oversampling, the accuracy and AUC are very high with both being 0.85 and 0.91 respectively. The recall (0.22) and F1 score (0.31) though are very low. Even though oversampling improves the values from 0.04 and 0.08, the overall performance is still same. Therefore, after taking all the performance scores into account, Naive Bayes is the best classifier for the prediction of Heart diseases after oversampling.

To sum up, after comparing both with and without oversampling, Logistic Regression can be considered the best algorithm for classification of Cardiovascular Disease. Despite Naive Bayes with oversampling having better recall scores, NB's

precision is lower, leading to a F1 score of 0.41 which is much lower than LR's F1 score of 0.51. For Cardiovascular Disease, the decision boundary between classes is much more linear with the classes being less well-separated. This leads to poor results of non-linear classifiers due overfitting of the data.

4.4 Improving Prediction Model

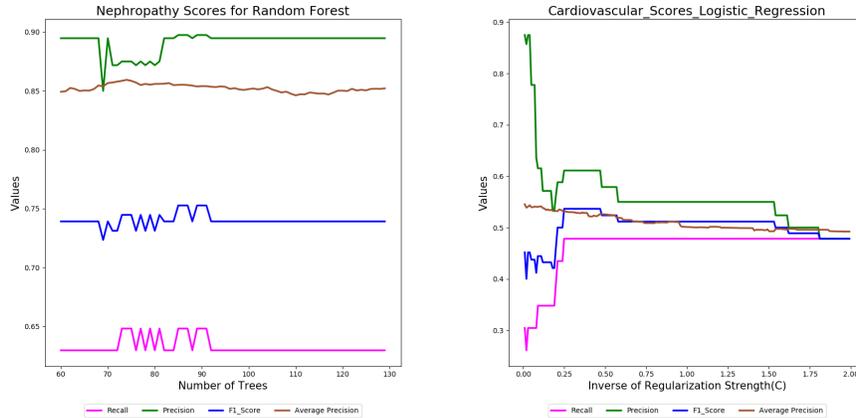


Fig. 5: Improved Nephropathy scores for **Fig. 6:** Improved Cardiovascular Disease scores for Logistic Regression Random Forest

Nephropathy For Nephropathy the best performance in terms of F1 score and recall was given by Random Forest with 80 trees. It is possible to obtain a good balance between performance, processing time, and memory usage when the number of trees in a forest is between 64 and 128 [31]. The Random Forest algorithm was implemented on a variable number of trees within the range of 60 to 130 to find which one maximizes the performance. It can be seen from Figure 5 that best performance is given by 85 trees with the F1 score, recall, precision and AP being 0.75, 0.65, 0.89 and 0.86 respectively. In general it can be observed from Figure 5 that the value of the metrics' fluctuates up and down until it reaches 100 trees and then for all metrics, the values stabilizes.

Cardiovascular Disease In the case of Cardiovascular Disease the best performance in terms of both F1 score and recall was given by Logistic Regression without oversampling. The inverse of regularization strength(C) in this case was 1. As regularization strength increases, the model generalizes better by taking slightly useful features into account. Since C is the inverse of the regularization strength, as C decreases regularization increases. Hence by changing the value of C, performance of the Logistic Regression model can be improved. To test this, Logistic Regression algorithm was applied with C values ranging from 0.01 to 2. From Figure 6 it can be seen that the best performance when considering all metrics comes when the value of C is 0.3. At this point, the F1 score, recall,

precision and AP are 0.54, 0.48, 0.61 and 0.53 respectively. Even though recall remains same, the value of the other metrics' increases. The greatest change is in precision which increases by 0.06. Overall, it can be seen from Fig. 8 that with a C value close to 0, the precision is very high and the recall is very low, leading to a poor F1 score. However as the C value is increased the F1 score increases, maximizing at around 0.3, which is where both recall and precision Scores are acceptable. Increasing C even further leads to the value of all the metrics either stabilizing or decreasing even more.

5 Discussion

In this paper, the best classifier for Nephropathy in terms of F1 Score and AUC is Random Forest with 85 trees. Furthermore, in the case of Cardiovascular Disease, Logistic Regression is the best classifier without oversampling in terms of F1 Score and AUC. Among all the classifiers applied, SVM showed high precision and Decision tree had the best recall score for predicting the risk of Nephropathy. However, Random Forest provided the highest AUC and F1 Score so it is more logical to select RF as the best classifier for prediction of Nephropathy onset. For Nephropathy, the dataset had an imbalance of ratio 1:4. From the results, it is evident that this data imbalance did not degrade the performance of the classifiers. Moreover, the best result was given by Random Forest, an ensemble method which is better at handling imbalanced data since it runs several Decision Tree classifiers and aggregates the result. On the other hand, in the case of Cardiovascular Disease prediction the imbalance ratio was 1:10, hence oversampling was used. The best result was obtained without applying Oversampling: Logistic Regression with the highest F1 score. Again the Recall score was not very high owing to less prevalence of patients in the dataset. Cardiovascular Disease onset prediction was further explored with the addition of oversampling. Nevertheless, oversampling failed to provide much improvement to the result. This was due to overfitting the data when the minority class was oversampled. Also the optimal class distribution is not known, in this paper it was assumed to be 50:50 which is not the case. SVM showed extremely low F1 Score of only 0.28 but the precision was higher than that of Logistic Regression without oversampling. Even though SVM had higher AUC score after oversampling, it was disregarded since AUC score can be misleading in cases of low prevalence. So, although AUC is a good metric for Nephropathy, it does not work well in case of Cardiovascular Disease. It is noticed that nonlinear classifiers perform better while predicting Nephropathy onset in comparison with Cardiovascular Disease prediction. Furthermore, in the later portion of the paper, Random Forest (for Nephropathy) and Logistic Regression without Oversampling (for Cardiovascular Disease) were further tuned to improve the results more. In the case of Random Forest, different number of estimators within the range of 60 to 130 trees was applied to find which one maximizes the performance. It has been observed that 85 trees maximizes Recall, Precision and F1 Score. On the other hand, for Cardiovascular disease prediction, the initial value of the inverse of regularization strength(C) was 1. C values ranging from 0.01 to 2 was exercised and the best results are

obtained when the value is 0.3. Value of both F1 Score and Recall increases and highest increment is observed in Precision which increases by 0.06. It can be seen from Figure 4 and Figure 5 that the performance metrics become horizontal lines after a certain value of C (for Logistic Regression) and specific number of trees (for Random Forest). Couple of interesting observations can be made from this research. Firstly, the model worked better for Nephropathy than it did for Cardiovascular Disease, even though the same dataset is used for both. This implies, the variables in the dataset is more inclined towards Nephropathy than Cardiovascular Disease. Further, the prevalence of the disease also played a significant part in the results. Since Nephropathy had 25% positive cases while Cardiovascular Disease had a mere 10%, the results were better for Nephropathy. The class imbalance led to the introduction of oversampling to in order to enhance the performance. Secondly, although the dataset had 164 features, it had only 779 instances to begin with, the model gave encouraging results.

6 Conclusion

This paper explains how Machine Learning based linear and non-linear classifiers can be adopted in clinical diagnostics to create systems that uses patient-specific information to predict the probability of Diabetes induced complications. A total of five classifiers have been applied here on an imbalance dataset and the results have been measured using various performance metrics. Random Forest and Logistic Regression provided best results for the prediction of Nephropathy and Cardiovascular Diseases respectively. Furthermore, Random Forest has been applied multiple times, each time with different number of estimators and the classifier works best with 85 trees. Similarly, Logistic Regression delivers best result when inverse of regularization strength, C , is 0.3 amongst all separate values that were tried. This research work was quite challenging owing to various facts. A major obstruction was the missing values present in the current dataset. Although the missing data were imputed using one of the best data imputation algorithms available, nevertheless, accuracy of all the imputed missing values may not be completely correct, according to medical aspect. Regardless of that, the model can satisfactorily predict onset of diabetes induced Nephropathy and Cardiovascular Disease. This paper also reflects how the clinical data, which are usually imbalanced, affects the prediction system. It is certain that in the foreseeable future a predictive model like this can be successfully used for prognosis, diagnosis and treatment planning of patients within clinical information systems. Since prevention or cure of Diabetes is yet to be found, it is vital to come up with systems that can aid in providing a better lifestyle to the ever increasing Diabetic population.

References

1. Sun, Y., Wong, A. K., and Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04), 687-719.

2. Mani, S., Chen, Y., Elasy, T., Clayton, W., and Denny, J. (2012). Type 2 diabetes risk forecasting from EMR data using machine learning. In AMIA annual symposium proceedings (Vol. 2012, p. 606). American Medical Informatics Association.
3. Cho, N. H., Shaw, J. E., Karuranga, S., Huang, Y., da Rocha Fernandes, J. D., Ohlogge, A. W., and Malanda, B. (2018). IDF Diabetes Atlas: global estimates of diabetes prevalence for 2017 and projections for 2045. *Diabetes research and clinical practice*, 138, 271-281.
4. Fowler, M. J. (2008). Microvascular and macrovascular complications of diabetes. *Clinical diabetes*, 26(2), 77-82.
5. Gregg, E. W., Li, Y., Wang, J., Rios Burrows, N., Ali, M. K., Rolka, D., et al. (2014). Changes in Diabetes-related complications in the United States, 1990–2010. *New England Journal of Medicine*, 370(16), 1514-1523.
6. De Fauw, J., Ledsam, J. R., Romera-Paredes, B., Nikolov, S., Tomasev, N., Blackwell, S., et al. (2018). Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine*, 24(9), 1342.
7. AbuKhousa, E., and Campbell, P. (2012, March). Predictive data mining to support clinical decisions: An overview of heart disease prediction systems. In *Innovations in information technology (iit), 2012 international conference on* (pp. 267-272). IEEE.
8. Yun, 15 July 2018, Chinese AI Beats Doctors in Diagnosing Brain Tumors.
9. Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., and Fotiadis, D. I. (2015). Machine Learning applications in cancer prognosis and prediction. *Computational and structural biotechnology journal*, 13, 8-17.
10. Barakat, N., Bradley, A. P., and Barakat, M. N. H. (2010). Intelligible Support Vector Machines for diagnosis of Diabetes mellitus. *IEEE transactions on information technology in biomedicine*, 14(4), 1114-1120.
11. Dagliati, A., Marini, S., Sacchi, L., Cogni, G., Teliti, M., Tibollo, V., et al. (2018). Machine Learning methods to predict Diabetes complications. *Journal of Diabetes science and technology*, 12(2), 295-302.
12. Liu, B., Li, Y., Sun, Z., Ghosh, S., and Ng, K. (2018). Early Prediction of Diabetes Complications from Electronic Health Records: A Multi-task Survival Analysis Approach.
13. Cho, B. H., Yu, H., Kim, K. W., Kim, T. H., Kim, I. Y., and Kim, S. I. (2008). Application of irregular and unbalanced data to predict diabetic Nephropathy using visualization and feature selection methods. *Artificial intelligence in medicine*, 42(1), 37-53.
14. Tanaka, S., Tanaka, S., Imuro, S., Yamashita, H., Katayama, S., Akanuma, Y., et al. (2013). Predicting macro-and microvascular complications in type 2 diabetes: the Japan Diabetes Complications Study/the Japanese Elderly Diabetes Intervention Trial risk engine. *Diabetes Care*, DC_120958.
15. Wilson, P. W., D'Agostino, R. B., Levy, D., Belanger, A. M., Silbershatz, H., and Kannel, W. B. (1998). Prediction of coronary heart disease using risk factor categories. *Circulation*, 97(18), 1837-1847.
16. Rahman, M. M., and Davis, D. N. (2013). Addressing the class imbalance problem in medical datasets. *International Journal of Machine Learning and Computing*, 3(2), 224.
17. Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A., and Tourassi, G. D. (2008). Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural networks*, 21(2-3), 427-436.
18. Ali, A., Shamsuddin, S. M., and Ralescu, A. L. (2015). Classification with class imbalance problem: a review. *Int J Adv Soft Comput Appl*, 7(3), 176-204.

19. Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4), 221-232.
20. Tomonaga, O. (2017, April 27). JAMP_DATA0722figshaer.xlsx (Version 1). [Retrieved from: doi.org/10.6084/m9.figshare.4924037.v1].
21. Gross, J. L., De Azevedo, M. J., Silveiro, S. P., Canani, L. H., Caramori, M. L., and Zelmanovitz, T. (2005). Diabetic Nephropathy: diagnosis, prevention, and treatment. *Diabetes care*, 28(1), 164-176.
22. Mogensen, C. E. (1987). Microalbuminuria as a predictor of clinical diabetic Nephropathy. *Kidney international*, 31(2), 673-689.
23. Han, S. H., Nicholls, S. J., Sakuma, I., Zhao, D., Koh, K. K. (2016). Hypertriglyceridemia and Cardiovascular Diseases: Revisited. *Korean Circ J*, 46(2), 135-144. Doi: 10.4070/kcj.2016.46.2.135.
24. Stekhoven, D. J., and Bühlmann, P. (2011). MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1), 112-118.
25. Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357.
26. He, H., and Garcia, E. A. (2008). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, (9), 1263-1284.
27. Jeni, L. A., Cohn, J. F., and De La Torre, F. (2013, September). Facing Imbalanced Data—Recommendations for the Use of Performance Metrics. In *Affective Computing and Intelligent Interaction (ACII)*, 2013 Humaine Association Conference on (pp. 245-251). IEEE.
28. Yuan, Y., Su, W., and Zhu, M. (2015). Threshold-free measures for assessing the performance of medical screening tests. *Frontiers in public health*, 3, 57.
29. Bekkar, M., Djemaa, H. K., and Alitouche, T. A. (2013). Evaluation measures for models assessment over imbalanced datasets. *Journal Of Information Engineering and Applications*, 3(10).
30. Wu, G., and Chang, E. Y. (2003, August). Class-boundary alignment for imbalanced dataset learning. In *ICML 2003 workshop on learning from imbalanced data sets II*, Washington, DC (pp. 49-56).
31. Oshiro, T. M., Perez, P. S., and Baranauskas, J. A. (2012, July). How many trees in a random forest?. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition* (pp. 154-168). Springer, Berlin, Heidelberg.

FF-SVM: New FireFly-based Gene Selection Algorithm for Microarray Cancer Classification

Nada Almugren and Hala Alshamlan

King Saud University, College of Computer and Information Sciences,
Riyadh, Kingdom of Saudi Arabia
`nada.almugren@hotmail, halshamlan@ksu.edu.sa`

Abstract. Several bio-inspired evolutionary based feature selection algorithms for microarray data classification have been proposed in the literature and show a good performance. In this research we proposed a wrapper feature selection algorithm for classifying cancer microarray gene expression profile that uses FireFly algorithm along with SVM classifier named FF-SVM. Support vector machine SVM classifier with leave-one-out cross validation LOOCV are used to measure the classification accuracy for the selected gene subset. Five benchmark microarray datasets of binary and multi class are used to evaluate FF-SVM algorithm. To validate the result of the proposed algorithm we compare it with other related state-of-the-art algorithms. The experiment proves that the FF-SVM show high classification accuracy using small number of selected genes.

Keywords: Gene Selection, Cancer Classification, Microarray, Gene Expression Profile, FireFly, SVM.

1 Introduction

DNA Microarray technology is a powerful tool that helps researchers monitor the gene expression level in an organism. Microarray data analysis is used to determine which genes are being differentially expressed. Differently expressed genes can be used in cancer diagnosing to differentiate between infected and uninfected tissues. Microarray data analysis provides valuable results which contribute towards solving gene expression profile problems. One the most important applications of Microarray data analysis is cancer classification. Classifying microarray data is challenging and considered as (NP)-Hard problem due to the high dimensionality found in a small sample size of gene expression data [1]. Also, gene expression data has a high complexity; genes are directly or indirectly correlated to each other [2].

Hence, most practical method to overcome these challenges is therefore a feature selection technique. The main idea behind the feature selection method is selecting the most informative and significant genes for the prediction (classification) problem. Several gene selection algorithms have been reported in the

literature, fall in three categories filter, wrapper and hybrid. Many filter approach statistical algorithms have been used for dimension reduction to remove redundant and irrelevant genes without using any learning algorithms, e.g. Mutual Information [3, 4]. In addition to these filter approaches, several wrapper algorithms and machine learning algorithms have been applied [5, 6]. Wrapper methods have achieved better performance than filter methods [7]. The hybrid approach is also adopted in order to utilize the advantages of both the filter and wrapper approaches [6, 8–10].

The aim of this study is to identify the most informative genes that contribute to cancer diagnosis. Therefore, we developed a new wrapper feature selection method for Microarray gene expression profiles to select the most informative genes that cause cancer. The proposed method consists of two phases: gene selections phase and classification phase. In the gene selection phase, the Firefly wrapper method was employed to find the optimal gene subset. In the classification phase, this optimal gene subset is tested based on a Support Vector Machine (SVM) classifier and the classification accuracy is obtained using leave-one-out cross validation (LOOCV). Five Microarray benchmark datasets of different cancer types was used to evaluate the proposed model. To validate the effectiveness of the proposed algorithm we compare it with other state-of-the-art algorithms. The experiment result shows the improvement in term of classification accuracy and the number of selected genes.

The rest of this paper is organized as follows: In Section 2, we briefly present background about FireFly algorithm and SVM classifier. This is followed by an explanation of our proposed *FF-SVM* algorithm in Section 3. Subsequently, Section 4 outlines the experimental setup and provides results. Finally, Section 5 draws conclusions of this paper.

2 Background

In this section, we briefly present a general background about FireFly algorithm (FFA) and Support Vector Machine (SVM) classification method.

2.1 FireFly Algorithm (FFA)

Firefly algorithm is bio-inspired global optimization method inspired by the flashing light patterns and behaviour of firefly insects thus it simulates the attraction behaviour of fireflies [11]. Fireflies use their flashing pattern to attract other fireflies (from the opposite sex). Though, firefly algorithms development was based on three idealised rules: first, being the assumption that all fireflies are attracted to each other, regardless of their sex. Second, their attractiveness is based on their brightness ability, thus their attractiveness will decrease as the distance between them increases. As a result, it will always be the less bright fireflies that move towards the brighter ones. Third, a firefly's brightness is determined or affected by the objective functions form. Firefly algorithm is metaheuristic population based where each firefly represents a possible solution

in the search space. This section presents the main behavior of artificial firefly algorithm. Two issues can be considered with the Firefly algorithm and they are; the variation of the brightness intensity, as well as how attractiveness is formulated. In the standard Firefly algorithm attractiveness is basically determined by brightness, which is associated with the objective function. Therefore, the brightness of firefly I at specific location x can be presented as $I(x) = f(x)$. Attractiveness β is also relative, in that it varies with the distance r_{ij} between firefly i and j . Thus, it differs from one firefly to the next, purely based on distance. Thus, the light intensity can be formulated in Equation 1.

$$I = I_0 e^{-\gamma r_{ij}^2} \quad (1)$$

Where the I_0 is the light intensity at the beginning. The Attractiveness β of the firefly is determined by the brightness. The attractiveness can be measured as shown in Equation 2 :

$$\beta = \beta_0 e^{-\gamma r_{ij}^2} \quad (2)$$

where β_0 is the attractiveness of the firefly at $r = 0$. γ is light absorption coefficient, which is fixed as 1.0 in FA. The distance between any two fireflies i and j at x_i and x_j is calculated in Equation 3 as Cartesian distance:

$$r_{ij} = \|x_i - x_j\| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3)$$

The movement of the firefly i towards more attractive firefly j calculated using Equation 4 , as bellow:.

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha(\text{rand} - 0.5) \quad (4)$$

In the Equation 4 the second term refer to the attraction, and the third term is the randomize term with α as randomization parameter, $\alpha \in [0, 1]$. β_0 is always set to 1 and rand is a random number between $[0, 1]$. In firefly the α is set to allow the variation in the solution. The value of γ characterizes the variation of the attractiveness, in many applications the value of γ varies from 0.01 to 100.

2.2 Support Vector Machine (SVM) Classifier

SVM is a supervised machine learning algorithm, which is typically used for classification purposes, based on statistical learning theory proposed by Vapnik [12]. The SVM has been used widely in many applications related to bioinformatic

and show good performance. Moreover, SVM has been extensively used to classify microarray data and it improved the classification accuracy. Main advantage for using SVM to classify microarray data, it works well when classifying high dimensional data [13]. In addition, it works well when the number of features is greater than the number of samples. SVM is centred around searching for a hyperplane that optimally divides the tuples, from one class to another. The hyperplane is found using the support vector and the margin. The support vector is calculated from the vectors (data points) that define the hyperplane. The margin is the shortest distance between the hyperplane and the nearest point (on two sides).

In order to evaluate the performance of the SVM classifier we apply leave-one-out cross validation (LOOCV). LOOCV is a model evaluation method, it is equal to K-fold cross validation based on its logic where K equal to N , the number of sample in the dataset. LOOCV works as follows, takes one sample as validation data (testing data) and the remaining as training. The process is repeated such that each sample in the dataset is used once as testing data. The aim of using LOOCV in our proposed algorithm is that it can prevent the problem of overfitting [14]. On the other hand, from the computational point of view LOOCV is considered as very expensive since the many times the training process is repeated.

3 Proposed Algorithm (FF-SVM)

In this section, the proposed FF-SVM algorithm will be described. The aim of FF-SVM algorithm is to find the most informative genes that maximize the SVM classifier performance. The basic concept of the proposed method is to compare each firefly on the swarm to every other firefly and based on the brightness (fitness value) one best firefly will be chosen. The steps of FF-SVM algorithm can be described as follows:

Step 1: Swarm Initialization The firefly algorithm first initiates a population of n fireflies $x_i, i = 1, 2, 3, \dots, n$ where n is the swarm size. The fireflies are positioned randomly in the search space. Every firefly x_i in the population represents a set of predefined number of features (genes) i.e. a possible solution to gene selection problem.

Step 2: Calculate the Fitness Function After the initialization step, the light intensity of the initial swarm that associated with the fitness function $f(x_i)$. Figure 1 represents a sample of the initial population. Where the x_i represents a firefly i.e. one possible solution with its fitness $f(x_i)$. Each firefly contains D number of genes.

Step 3: Find the Best Firefly In this step the algorithm finds the best firefly that maximizes the classification accuracy while keeping the minimum number

of selected genes. The idea behind this step is to compare each firefly in the population with every other firefly.

Given a fixed number of generation (iteration), the proposed algorithm starts the comparison. If the fitness of the firefly i is less than the fitness of the firefly j , then firefly i will move toward firefly j . The movement of the firefly is done using Equation 4. Because of the position of the firefly i is updated its fitness must also updated. Then the algorithm follows the same procedure for subsequent iteration.

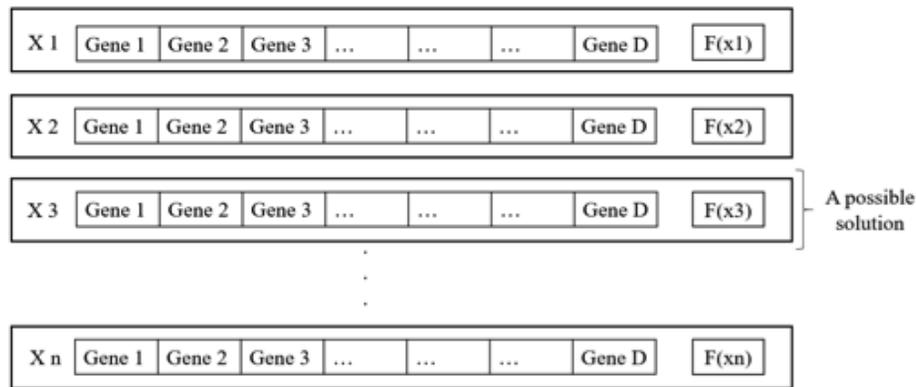


Fig. 1. The FireFly population sample. Where x_1 to x_n in the population represent a candidate solution that contain set of predefined number of genes D . The $f(x_1)$ to $f(x_n)$ is the fitness function for each firefly x_i , $i = 1, 2, \dots, n$

Step 4: Ranking the Result Ranking the fireflies that resulting from the comparison step and return the best firefly. Figure 2 show the proposed algorithm flow chart.

The fitness function for the FF-SVM is to maximize classification performance i.e. classification accuracy while keeping minimum number of selected features. The classification accuracy is obtained using Leave One Out Cross Validation LOOCV. In Algorithm 1, we present pseudo code for FF-SVM algorithm.

4 Experimental Result

In this section the datasets used to test the proposed algorithm as well as the experiment setup and experiment result are explained.

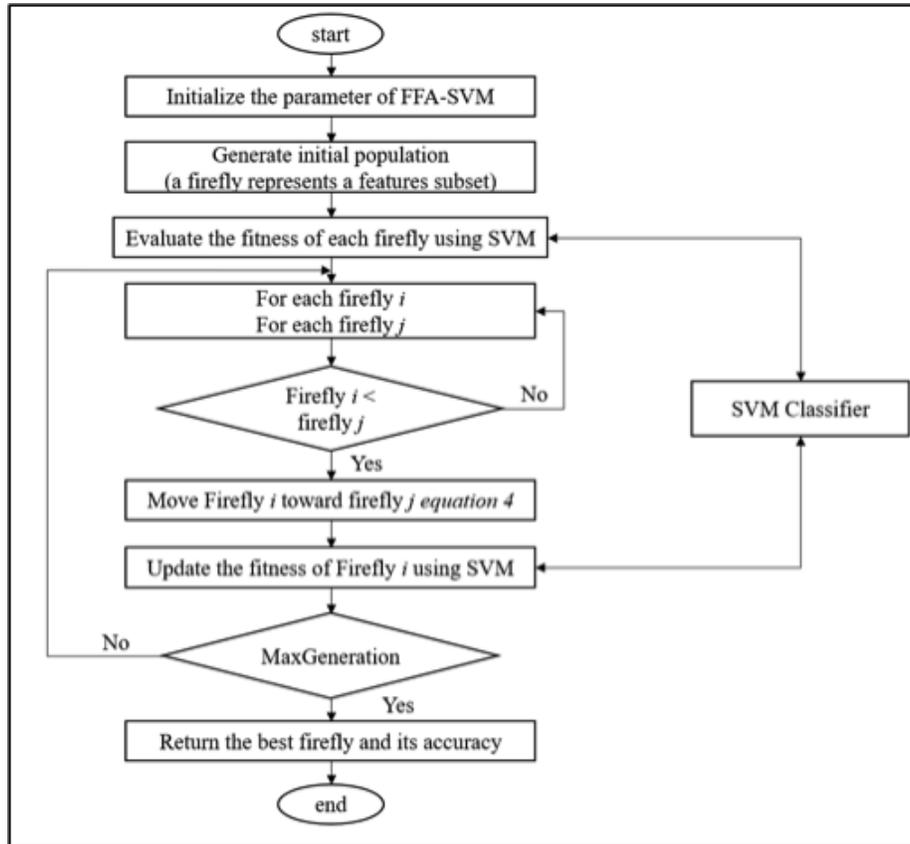


Fig. 2. Proposed (FF-SVM) algorithm flow chart.

4.1 Microarray Dataset

In order to evaluate the proposed algorithm Five bunch mark microarray dataset have been used. The used datasets are of binary and multi class namely, SRBCT, Lung, Colon, Leukemia1 and Leukemia2. Table 1 present a detailed description of these datasets.

4.2 Experiment Setup

The proposed algorithm is written in Python development environment. All the experiments are performed in iMac desktop computer 4 GHs processor and 8 GB of RAM memory. In the experiment different parameter need to be set, the number of iteration, number of population and the number of run. The number of population set to be 70 fireflies i.e. 70 possible solution. One iteration gives

Algorithm 1 FF-SVM Algorithm

Input:

Microarray dataset
 Size of firefly i.e. number of features or genes, D .
 Define light absorption coefficient, $\gamma \in [0.01, 100]$.
 Define randomization parameter, $\alpha = 1$.
 Define attractiveness parameter, $\beta_0 = 1$.
 Set maximum number of iteration, MaxGeneration=25.

Output:

The best firefly and its fitness

Algorithm:

Generate initial population of n fireflies randomly $x_i, i = 1, 2, 3, \dots, n$;
 Evaluate the fitness each firefly using objective function $f(x)$;
while ($t < \text{MaxGeneration}$) **do**
 for $i = 1$ to n **do**
 for $j = 1$ to n **do**
 if ($f(x_i) < f(x_j)$) **then**
 Move firefly i towards j using
 Calculate the distance r
 Calculate the new position x_i of the firefly i
 Update the fitness of firefly i
 end if
 Evaluate new solutions and update light intensity
 end for
 end for
end while
 Rank the fireflies and find best firefly

possibilities to get the best firefly in one generation. Therefore, multiple iteration (generation) is give the possibilities to get the optimal feature set (firefly). Thus, the number of generation is set to 25. In order to make the result more accurate and statistically valid, the experiment is repeated 20 times for each dataset. The best, worst and average of the classification accuracy is calculated. The accuracy is obtained using SVM classifier with Leave One Out Cross Validation LOOCV.

4.3 Results and Analysis

In this section we present and analyze the result obtained from proposed algorithm. In addition, we will compare our result with other relevant algorithms

Experiment Result The objective of feature selection is to maximize the classification performance while minimizing the number of selected features. For each dataset we implement the algorithm on different number of features. For example, we implement the FF-SVM in the Lung dataset using 2,5 and 10 number of genes (feature). The result of different experiments is presented in this section.

Table 1. Description of Microarray Datasets.

DATA SET	NO. CLASSES	NO. SAMPLE	NO. GENES	NO. IN CLASS	SAMPLE EACH
LEUKEMIA2 [15]	3	72	7129	28	AML, 24 ALL, AND 20 MLL
SRBCT [16]	4	83	2308	29	EWS, 18 NB, 11 BL, AND 25 RMS
LUNG [17]	2	96	7129	86	CAN-CER AND 10 NORMAL
LEUKEMIA1 [2]	2	72	7129	25	AML AND 47 ALL
COLON [18]	2	62	2000	40	CAN-CER AND 22 NORMAL

Table 2. Colon dataset result

NO. GENES	AVERAGE	BEST	WORST
10	92.9%	95.2%	90.3%
15	92.7%	95.3%	90.3%
17	92.6%	98.4%	90.3%
19	93.5%	98.4%	90.3%

Table 2 show the best, worst and average classification accuracy of applying FF-SVM algorithm in Colon dataset. We can notice that the best average accuracy is obtained when the number of genes is 19. The other number of selected gene does not improve the classification accuracy.

Table 3 show the best, worst and average classification accuracy of applying FF-SVM algorithm in Leukemia1 dataset. The result show that in all targeted number of genes the best accuracy achieves 100%. The highest average accuracy is obtained when the number of selected genes is 11 with 99.5%.

Table 4 show the best, worst and average classification accuracy of applying FF-SVM algorithm in Lung dataset. The result shows that in all different number of selected genes the average accuracy is 100%.

Table 5 show the best, worst and average classification accuracy of applying FF-SVM algorithm in SRBCT dataset. The highest accuracy obtained when the number of selected gene is 12 and 14. For the other number of selected genes the average accuracy is higher than 92.5%.

Table 6 show the best, worst and average classification accuracy of applying FF-SVM algorithm in Leukemia2 dataset. the highest accuracy obtained when

Table 3. Leukemia1 dataset result

NO. GENES	AVERAGE	BEST	WORST
3	98.7%	100%	97.2%
5	97.8%	100%	94%
8	99.1%	100%	97.2%
10	98.9%	100%	97.1%
11	99.5%	100%	98.6%

Table 4. Lung dataset result

NO. GENES	AVERAGE	BEST	WORST
2	100%	100%	100%
5	100%	100%	100%
10	100%	100%	100%

the number of selected genes is 19. For the other numbers of selected gene, the accuracy is less than 91%.

Comparison Result To validate effectiveness of the FF-SVM algorithm we compare it with other wrapper-based and hybrid-based state-of-the-art algorithms. The algorithms used in the comparison employed evolutionary based wrapper algorithm for the feature selection. The comparison results are presented in Table 7 in term of classification accuracy and number of selected genes.

According to Table 7 the proposed method improved the classification accuracy and reduced the search space complexity. When comparing the result to wrapper approach the FF-SVM outperforms the other reported wrapper methods in four out of five datasets with small number of selected genes. However, the proposed method achieves more than 92.5% in all dataset and selects less than 22 genes. In leukemia2 dataset the PSO-SVM [5] gets 95.83% which is higher than the proposed algorithm but selects 61 genes which is relatively high number of selected genes compare to 19 gene in our proposed algorithm. For the Colon dataset the FF-SVM achieve the second-best performance. In the Leukemia1, lung and SRBCT datasets the FF-SVM obtained the highest accuracy with smallest number of selected genes.

However, when compared the result to hybrid approach the FF-SVM achieve better performance in Lung dataset. for the rest of the datasets the proposed algorithm obtained lower performance. Comparing to the other wrapper-based algorithm the FF-SVM achieve good performance in terms of the accuracy and the number of selected genes.

Table 5. SRBCT dataset result

NO. GENES	AVERAGE	BEST	WORST
5	92.6%	95.2%	90.3%
7	94.3%	95.2%	92.3%
10	96.7%	98.8%	95.1%
12	97.5%	98.8%	96.4%
14	97.5%	98.8%	96.4%

Table 6. Leukemia2 dataset result

NO. GENES	AVERAGE	BEST	WORST
5	81.33%	83.3%	79.2%
10	88.38%	97.2%	84.5%
13	90.55%	95.8%	87.5%
15	90.94%	94%	88.9%
19	92.58%	95.8%	90.3%

Table 7. Comparison of the classification accuracy of our proposed algorithm (FF-SVM) with more recent state-of-the-art gene selection algorithms with regard to five of the microarray datasets (The number between parentheses represents the number of selected genes).

GENE SELECTION ALGORITHMS	BINARY CLASS DATASE			MULTI CLASS DATASE	
	COLON	LUNG	LEUKEMIA1	SRBCT	LEUKEMIA2
FF-SVM	92.7(22)	100(2)	99.5(11)	97.5(14)	92.6(19)
PSO-SVM [5]	93.55(78)	94.79(65)	95.83(53)	93.97(68)	95.83(61)
GA-SVM [5]	93.55(83)	95.83(62)	91.99(51)	92.77(74)	94.44(57)
ABC-SVM [5]	92.44(20)	93.7(8)	92.5 (14)	91.5(10)	93.1(20)
ACO-SVM [19]	91.5(8)	-	-	-	-
GA-SVM [19]	84.6(8)	-	91.5(5)	-	-
MOBBA-LS [9]	-	-	-	85(6)	-
HS-GA [20]	95.9(20)	-	97.5(20)	-	-
BPSO-CGA [21]	99.964(214)	-	-	-	100(196)
HPSO-LS [22]	84.38 (60)	-	89.28(100)	-	-
IDGA [23]	-	-	100(15)	100(18)	-
IG/SGA [24]	85.48 (60)	100(9)	97.06 (3)	-	-
CLA-ACO [8]	-	-	95.95(3)	-	-
RFR-BBHA-BAGGING [25]	91.93(3)	-	-	-	-
ICA+ABC [10]	98.14(16)	-	98.68(12)	97.33(15)	-
SU-HSA [26]	87.53(9)	-	100(26)	99.89(37)	100(24)
mRMR-ABC [27]	96.77 (15)	100 (8)	100 (14)	100 (10)	100 (20)
MIMAGA [6]	83.41(202)	-	-	88 .64 (207)	-

5 Conclusion

In this research we proposed FF-SVM feature selection algorithm for microarray gene expression profile. The proposed algorithm was combined with SVM classifier. Experiment result show that the proposed method achieved higher accuracy with smaller number of selected genes than other wrapper-based algorithm reported in the literature. On the other hand, when compare the algorithm to hybrid algorithms, the hybrid algorithms outperform the proposed algorithm in terms of the classification accuracy and the number of selected genes. Thus, as future work we will add filter phase to the proposed algorithm so that it will the microarray dataset will be filtered. Then, the filtered data will input to the FF-SVM algorithm in order to obtain higher performance.

References

1. Narendra, P.M., Fukunaga, K.: A branch and bound algorithm for feature subset selection. *IEEE Transactions on computers* (9) (1977) 917–922
2. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science* **286**(5439) (1999) 531–537
3. Lazar, C., Taminau, J., Meganck, S., Steenhoff, D., Coletta, A., Molter, C., de Schaetzen, V., Duque, R., Bersini, H., Nowe, A.: A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* **9**(4) (2012) 1106–1119
4. Cai, R., Hao, Z., Yang, X., Wen, W.: An efficient gene selection algorithm based on mutual information. *Neurocomputing* **72**(4-6) (2009) 991–999
5. Alshamlan, H.M., Badr, G.H., Alohal, Y.A.: Abc-svm: artificial bee colony and svm method for microarray gene selection and multi class cancer classification. *International Journal of Machine Learning and Computing* **6**(3) (2016) 184–190
6. Lu, H., Chen, J., Yan, K., Jin, Q., Xue, Y., Gao, Z.: A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing* **256** (2017) 56–62
7. Alshamlan, H., Badr, G., Alohal, Y.: A comparative study of cancer classification methods using microarray gene expression profile. In: *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, Springer (2014) 389–398
8. Sharbaf, F.V., Mosafer, S., Moattar, M.H.: A hybrid gene selection approach for microarray data classification using cellular learning automata and ant colony optimization. *Genomics* **107**(6) (2016) 231–238
9. Dashtban, M., Balafar, M., Suravajhala, P.: Gene selection for tumor classification using a novel bio-inspired multi-objective approach. *Genomics* **110**(1) (2018) 10–17
10. Aziz, R., Verma, C., Srivastava, N.: A novel approach for dimension reduction of microarray. *Computational biology and chemistry* **71** (2017) 161–169
11. Yang, X.S.: Firefly algorithms for multimodal optimization. In: *International symposium on stochastic algorithms*, Springer (2009) 169–178
12. Vapnik, V.N.: Adaptive and learning systems for signal processing communications, and control. *Statistical learning theory* (1998)

13. Han, J., Pei, J., Kamber, M.: Data mining: concepts and techniques. Elsevier (2011)
14. Ng, A.Y., et al.: Preventing” overfitting” of cross-validation data. In: ICML. Volume 97. (1997) 245–253
15. Armstrong, S.A., Staunton, J.E., Silverman, L.B., Pieters, R., den Boer, M.L., Minden, M.D., Sallan, S.E., Lander, E.S., Golub, T.R., Korsmeyer, S.J.: Mll translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature genetics* **30**(1) (2001) 41
16. Khan, J., Wei, J.S., Ringner, M., Saal, L.H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C.R., Peterson, C., et al.: Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature medicine* **7**(6) (2001) 673
17. Beer, D.G., Kardia, S.L., Huang, C.C., Giordano, T.J., Levin, A.M., Misek, D.E., Lin, L., Chen, G., Gharib, T.G., Thomas, D.G., et al.: Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nature medicine* **8**(8) (2002) 816
18. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences* **96**(12) (1999) 6745–6750
19. Hernandez, J.C.H., Duval, B., Hao, J.K.: A genetic embedded approach for gene selection and classification of microarray data. In: European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, Springer (2007) 90–101
20. Vijay, S.A.A., GaneshKumar, P.: Fuzzy expert system based on a novel hybrid stem cell (hsc) algorithm for classification of micro array data. *Journal of medical systems* **42**(4) (2018) 61
21. Chuang, L.Y., Yang, C.H., Li, J.C., Yang, C.H.: A hybrid bps-cga approach for gene selection and classification of microarray data. *Journal of Computational Biology* **19**(1) (2012) 68–82
22. Moradi, P., Gholampour, M.: A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Applied Soft Computing* **43** (2016) 117–130
23. Dashtban, M., Balafar, M.: Gene selection for microarray cancer classification using a new evolutionary method employing artificial intelligence concepts. *Genomics* **109**(2) (2017) 91–107
24. Salem, H., Attiya, G., El-Fishawy, N.: Classification of human cancer diseases by gene expression profiles. *Applied Soft Computing* **50** (2017) 124–134
25. Pashaei, E., Ozen, M., Aydin, N.: Gene selection and classification approach for microarray data based on random forest ranking and bbha. In: Biomedical and Health Informatics (BHI), 2016 IEEE-EMBS International Conference on, IEEE (2016) 308–311
26. Shreem, S.S., Abdullah, S., Nazri, M.Z.A.: Hybrid feature selection algorithm using symmetrical uncertainty and a harmony search algorithm. *International Journal of Systems Science* **47**(6) (2016) 1312–1329
27. Alshamlan, H., Badr, G., Alohal, Y.: mrmr-abc: a hybrid gene selection algorithm for cancer classification using microarray gene expression profiling. *BioMed research international* **2015** (2015)

Fast K-medoids Clustering Algorithm Using Triangle Inequality

Xiaochun Wang¹, Xia Li Wang², Xuan Xiong Lin³

¹School of Software Engineering, Xi'an Jiaotong University, Xi'an, 710049, CHINA
xiaocchunwang@mail.xjtu.edu.cn

²School of Information Engineering, Changan University, Xi'an, 710061, CHINA
xlwang@chd.edu.cn

³Xi'an Jiaoda JointSky Corporation, Xi'an, 710065, CHINA
linxuanxiong88@163.com

Abstract Clustering is an interesting and important problem which is relevant for a wide variety of applications, including multimedia information retrieval, data mining, and pattern recognition. As a result, many techniques have been developed for it. Being one of the most popular clustering methods, K-means algorithm is very efficient for massive datasets but is very prone to the effects of outliers. To improve, K-medoids algorithm has been proposed as a better alternative. Unfortunately, the scalability to large datasets tends to be a major obstacle in the development of efficient K-medoids algorithms. To partially circumvent this drawback, in this paper, we propose a fast version of INCK, a state-of-the-art K-medoids clustering algorithm, which eliminates unnecessary distance computations in the medoid initialization and updates by way of triangle inequality. Experimental results indicate that our scheme scales well for large sizes of data.

Keywords: clustering; Partitioning based clustering; K-means algorithm; K-medoids algorithm; Triangle inequality.

1 Introduction

The goal of modern clustering algorithms is to group data points into clusters with identical characteristics and to generate patterns and knowledge that can be exploited further [1]. Over the past two decades, clustering has drawn considerable attention within the data mining research community and, as a result, much progress has been realized. Being one of the most popular clustering techniques, *K*-means clustering algorithm was proposed by James MacQueen in 1967 [2] and works as a simple centroid-based method which, for a given data set, iteratively finds *k* centroids and assigns every object to the nearest centroid until no change happens in the cluster membership [3]. In spite of its being very efficient in terms of the computational time for large datasets, *K*-means algorithm is known to be sensitive to outliers. As a better alternate in this respect, *K*-medoids clustering algorithm was proposed where representative objects called medoids (i.e., the most centrally located object in a cluster)

are considered instead of centroids [4]. Because it is less sensitive to outliers and the dimensionality of a dataset in comparison with the K -means clustering, K -medoids algorithms have been applied to various fields [5-8]. Among many algorithms proposed for K -medoids clustering, a simple and fast algorithm (referred to as FastK in the following) proposed by Park and Jun in 2009 is known to be the most popular improved one [9]. However, the initial medoids optimized by the FastK algorithm usually appear in the same cluster, resulting in degraded final clustering performance. To overcome this drawback, recently in 2018, Yu et al. proposed a more effective algorithm called INCK [10]. Unfortunately, both methods require all pairwise distances be computed beforehand and maintained in the main memory, and therefore, work inefficiently for large data sets due to their high time and space complexities.

In this study, we propose a fast K -medoids clustering algorithm by modification of the basic INCK method. The proposed algorithm aims at faster processing by skipping unnecessary distance computations in the initialization and updates of medoids. More specifically, a favorable property of INCK algorithm is that the sum of distances of a data point to all others in a cluster is a monotonic nondecreasing function of the portion of the dataset already explored. To this end, we also propose the use of triangle inequality that allows early detection of such unnecessary operations in the medoid selection and updates. The most important contribution of the proposed method is its significant tempo and spatial efficiency for large data sets in the sense that our method does not require to hold in main memory all the pairwise distances among data points, which is otherwise seldom feasible when memory is limited. To be as general as possible, our algorithm has no specific requirements on the size of data sets and explores the applications to clustering in large multi-dimensional data sets. Experiments conducted demonstrate the efficiency of the proposed approach in comparison with the INCK clustering algorithm.

The rest of the paper is organized as follows. We briefly introduce the INCK algorithm in Section 2. In Section 3, the proposed fast INCK (FINCK) method is presented. In Section 4, experiments are conducted to demonstrate the efficiency of FINCK. Finally, conclusions are made in Section 5.

2 Related work

In the INCK algorithm, a dataset consisting of n d -dimensional objects, $D = \{x_1, x_2, \dots, x_n\}$, and the number of clusters, k , are given, and the output is a set of clusters, $C = \{C_1, C_2, \dots, C_k\}$, formed based on the minimization of the total cost, E , as,

$$E = \sum_{i=1}^k \sum_{x \in C_i} \text{dist}(o_k, x)^2 \quad (1)$$

where $\{o_1, o_2, \dots, o_k\}$ denotes the set of k medoids, and the Euclidean distance between object i and object j is used,

$$\text{dist}(x_i, x_j) = \sqrt{\sum_{m=1}^d (x_{im} - x_{jm})^2} \quad i, j = 1, \dots, n \quad (2)$$

To exclude some objects that are not suitable to act as medoids, INCK first defines a candidate medoids subset S_m based on two definitions, the centroid (i.e., the object mean) and the variance of D as,

$$\begin{aligned} \bar{x} &= \frac{1}{n} \sum_{i=1}^n x_i \\ \sigma &= \sqrt{\frac{1}{n-1} \sum_{i=1}^n \text{dist}(x_i, \bar{x})^2} \end{aligned} \quad (3)$$

The variance for all the data objects is defined to be,

$$\sigma_i = \sqrt{\frac{1}{n-1} \sum_{j=1}^n \text{dist}(x_i, x_j)^2}, \quad i = 1, \dots, n \quad (4)$$

Based on Eq.(3) and Eq.(4), the candidate medoids subset S_m can be defined to be,

$$S_m = \{x_i \mid \sigma_i \leq \lambda \sigma, i = 1, \dots, n\} \quad (5)$$

where λ is a stretch factor. The distance, d_i , of object x_i is defined to be,

$$d_i = \sum_{j=1}^n \text{dist}(x_i - x_j) \quad i, j = 1, \dots, n \quad (6)$$

The INCK algorithm starts with two initial medoids, then increases the number of medoids in a step-wise fashion, ending at k medoids. The first medoid o_1 is determined according to the following condition,

$$o_1 = \arg \min_{x_i \in S_m} \{d_i \mid i = 1, \dots, n\} \quad (7)$$

The two initial medoids should be as far away from each other as possible. The second medoid o_2 is determined as,

$$o_2 = \arg \max_{x_i \in S_m} \{\text{dist}(x_i, o_1) \mid i = 1, \dots, n\} \quad (8)$$

To select the rest medoids, a candidate medoid set $O' = \{o'_1, o'_2, \dots, o'_i\}$ is determined as,

$$o'_i = \arg \max_{x_l \in C_i \cap S_m} \{dist(x_l, o_i) \mid l = 1, \dots, n\} \quad (9)$$

A new medoid, o_{i+1} , $i+1 \leq k$, can be selected as,

$$o_{i+1} = \arg \max_{x_j \in O'} \{dist(o_j, o'_i) \mid j = 1, \dots, i\} \quad (10)$$

This process continues until all k initial medoids are selected. The whole clustering process of INCK algorithm flow is presented in Fig. 1.

Algorithm 1 INCK.

Input: dataset D , clusters number K , stretch factor λ
Output: clusters C_1, \dots, C_K

1. Calculate the distance between each pair of objects using Eq. (2).
2. Calculate the data set D variance σ using Eq. (3), and the object variance σ_i using Eq. (4), then determine the candidate medoids subset S_m using Eq. (5).
3. Select two initial medoids $O = \{o_1, o_2\}$ using Eq. (7) and Eq. (8).
4. Assign each object to the nearest medoid and calculate the total cluster cost E using Eq. (1).
5. for k from 2 to $K-1$
6. Calculate the new increased medoids o_{k+1} using Eq. (10) and generate a new medoids set $O \leftarrow O \cup \{o_{k+1}\}$.
7. Repeat
8. Assign each object to the nearest medoid based on the nearest distance principle.
9. Update the medoids set O similar to FastK.
10. Calculate the total cluster cost E using Eq. (1).
11. until the total cluster cost E no longer changes.
12. end for

Fig. 1. The INCK algorithm

3 The proposed approach

From the previous section, it can be seen that standard INCK algorithm relies on the precomputation of all pairwise distances. In d -dimensional space, for a dataset of size n , the tempo and spatial complexities of a dissimilarity (i.e., distance) matrix are both $O(n^2)$, which limits its use to modern large datasets. Fortunately, this may not be necessary. In this part, an efficient K -medoids algorithm is developed.

3.1 A simple idea

The fast INCK algorithm proposed in this study attempts to achieve greater speed by skipping unnecessary distance calculations. In terms of reducing computational complexity, our basic approach is based on the following property.

Property 1. The sum of distances of a data point to all other data points in a cluster is a monotonic nondecreasing function of the portion of the dataset already explored.

To explain the main idea of Property 1, a 2-dimensional sample dataset of size 6 is shown in Fig. 2. Suppose that data point 4 is the medoid candidate whose total sum of distances to other 5 data points is 8, which can be used as a threshold, and that data point 1 is furthest to the rest data points whose partial sum of distances to data points 5 and 6 is already 9. Since the total sum of distances of data point 1 to other 5 data points is larger than 9, there is no need to further calculate its distances with the rest data points, i.e., data points 2, 3 and 4.

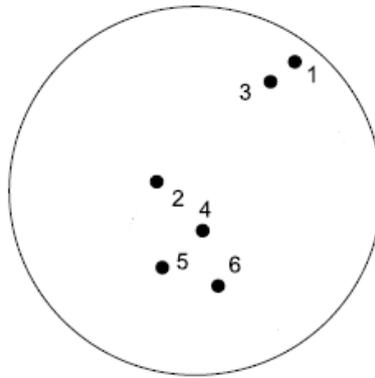


Fig. 2. An illustration of Property 1

Motivated by this property, to modify INCK for our purpose, the proposed algorithm is based on the following observations.

First of all, for early detection of unnecessary distance calculations, a threshold value should be provided as tight as possible. In the medoid initialization, according to Eq. (3), (4) and (5), a threshold can be determined to start the search for the candidate medoids subset. For the first medoid selection and in the subsequent medoid updating, a centroid is first calculated from all the data points in the database or in a cluster. Next the data point closest to the centroid can be found to be the medoid candidate and its sum of distances to all other data points in the database or in the cluster can be used as a threshold. For the latter case, the threshold is not fixed but can be updated with a smaller value if such value happens during the search process thereafter.

Secondly, with the threshold being set, it is true that the search for medoid candidates can be conducted in any order starting with any data points in the database (or a cluster). According to Eq. (6), in a linear fashion, a data point is compared sequentially with all the data points in the database (or in a cluster) and the number of distance calculations increases monotonically with the number of data points to be searched through. In other words, the computational complexity increases directly with the database size. This is neither efficient nor necessary, because, if a data point is associated with the largest distance span over the whole database (or a cluster), say data

point 1 and data point 6 in Fig.2, a small number of distance computations to its furthest data point at the other end of the whole database (or the cluster) can quickly make its partial sum of distances larger than the threshold and prune it from further consideration. Therefore, for this effect to be utilized, the search for medoid update should proceed in some order so that the data involved in the search can be diminished. In other words, data points with the largest distance spans over the whole dataset (or the cluster) should be located and given highest priority to begin the search process. To do so, with no a priori assumptions on the distribution model underlying the data, for simplicity, we randomly select a data point, say the first data point encountered, and calculate its distances to the rest of the database (or the cluster). The data point with the largest distance is selected and its distances to all other data points are computed. By doing so, a larger (if not the largest) distance span of the data distribution can be resulted and the partial distance sums can be used to pruning the distance computations efficiently. When such a data point is located, say data point 1 in Fig. 2, to generate the search order, its distances to all other data points in the database (or the cluster) are ordered in a non-decreasing order, and the data points in its local nearest neighborhood (i.e., starting from the nearest end), that is, data point 3, can be pruned by calculating its distances with the furthest points of data point 1 (i.e., starting from the furthest end), that is, data points 5 and 6, if the partial sum is larger than the threshold, 8.

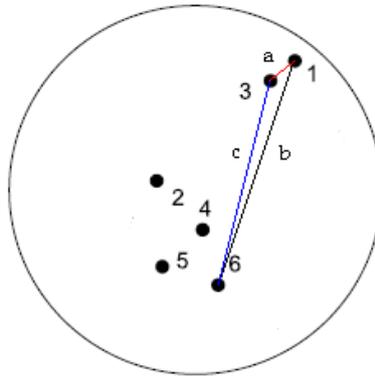


Fig. 3. An illustration of triangle inequality.

Thirdly, to further reduce the number of distance computations and enhance the development of early detection of unnecessary operations, the use of triangle inequality can be of some help. With triangle inequality, reduction of computational complexity and further acceleration are possible if unnecessary operations in the calculation of the cumulative distance sum for every point can be detected early. An illustration for this mechanism using the sample data in Fig.2 is shown in Fig. 3. In the figure, data points 1, 3 and 6 form a triangle with edges being a , b and c . According to triangle inequality, $b - a \leq c \leq b + a$. Therefore, the distance between data point 3 and data point 6 (the furthest point of data point 1) is upper bounded by $b + a$ and lower bounded by $b - a$. With large data spans, the upper bound and the lower bound are

very close and the lower bound can be used as an approximation for c . In other words, one arithmetic operation will save one distance computation. And data point 3 can retire from further considerations by only one distance computation with data point 1 and several arithmetic operations instead of multi-dimensional distance computations with all those data points in the database (or the cluster).

Finally, as we process data points away from data point 1 in the order of sorted distances, the difference between the upper and lower bound gradually become larger and larger, say for data point 2, the approximation is not very accurate and multi-dimensional distance computations become necessary, which can make the search rather inefficient. In other words, when most data points residing in local neighborhood close to the data points corresponding to the largest distance span are excluded from the further consideration, the attempts to find data points with larger partial sum of distances over all other data points so as to be terminated at an early stage do not have much gains. For a large number of boundary points to be pruned from search as much as possible, when the lower bound and the upper bound have a certain large difference, the data points processed so far can then retire, data distribution may change and another largest span structure can be constructed to replace the current one so that the search process can continue and be more effective. An illustration of such strategy is given in Fig.4, when data points 1 and 3 retire, data points 2 and 6 will form the next largest span for early detection of unnecessary distance computations. By this way, the average number of multi-dimensional distance computations can be further reduced.

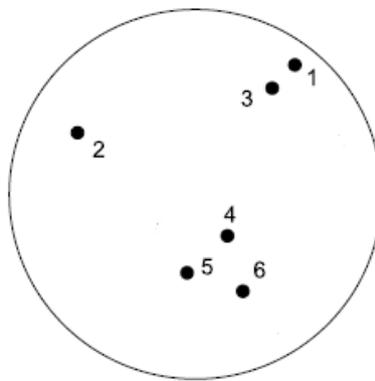


Fig. 4. An illustration of reconstructed search.

Therefore, in our search process, first thresholds are determined, next, a search order should be established, then data points can be processed and some can retire on the flying if satisfying the requirement, and, after certain number of data points retire, the search structure can be updated with a new one. Based on these ideas, a fast INCK (FINCK) algorithm is developed in the following.

3.2 Fast INCK algorithm

Suppose that n objects having d variables each should be grouped into K ($K < n$) clusters, where K is assumed to be given. The Euclidean distance is used as a dissimilarity measure in this study. The proposed algorithm is composed of the following steps.

1. Calculate the data set D 's variance σ using Eq.(3) and then determine the candidate medoids subset S_m using Eq.(5) which determines its threshold, threshold_1.
2. To find the first medoid candidate, calculate the mean (i.e., the centroid) of all objects in the whole database. Then search for data point with the smallest distance value to the mean, and calculate its sum of distances according to Eq.(7), which determines its threshold, threshold_2.
3. Calculate the distances of a randomly chosen data point to all other data points and remember the data point with the largest distance, say data point x_i . Calculate the distances of data point x_i to all other data points, v_j , $1 \leq j \leq n$. Sort v_j 's in ascending order and construct the first search structure. Next, starting from the first object having the smallest v_j value, do an arithmetic calculation using triangle inequality to find its accumulated partial sum. If the accumulated partial sum is larger than threshold_1, it is removed from further consideration. If the accumulated partial sum is larger than threshold_2 but smaller than threshold_1, it is removed from further consideration of being the first candidate medoid but is retained for the consideration of being a member of the candidate medoids subset. If the accumulated partial sum is smaller than threshold_2, it is retained for further consideration of being the first candidate medoids and threshold_2 could be updated. After a predefined number, m , objects are checked such that the upper bound and lower bound have a large difference, a new search structure is constructed. This process continues until all the data points are checked, at which time, the first medoid and the medoids candidate subset are discovered.
4. Select two initial medoids $O = \{o_1, o_2\}$ using Eq.(7) and Eq.(8) .
5. Assign each object to the nearest medoid and calculate the total cluster cost E using Eq. (1).
6. for k from 2 to K
7. Calculate the new increased medoids o_{k+1} using Eq.(10) and generate a new medoids set $O \leftarrow O \cup \{o_{k+1}\}$.
8. Repeat
9. Assign each object to the nearest medoid based on the nearest distance principle.
10. Update the medoid of each cluster by our pruning rules. Calculate the distances of a data point to all other data points in a cluster based on the chosen dissimilarity measure and remember the data point with the largest distance, say data point x_i . Calculate the distances of data point x_i to all other data points in the cluster. Sort the distances in ascending order and construct the first search structure. Next, starting from the first object having the smallest distance value, do an arithmetic calculation using triangle inequality to find its accumulated partial sum. If the accumulated partial sum is larger than threshold_2 (determined by step 2 with the whole database being changed to the cluster), it is

removed from further consideration. If the sum of the distances of the data point to the rest of points in the cluster is smaller than threshold_2 , threshold_2 could be updated. After a predefined number, m , objects are checked such that the upper bound and lower bound have a large difference, a new search structure is constructed. This process continues until all the data points in the cluster are checked, at which time, the medoids candidate for the cluster is updated.

11. Calculate the total cluster cost E using Eq. (1).
12. until the total cluster cost E no longer changes.
13. end for

Table 1 and Table 2 present the medoid candidate subset initialization using the pruning rule and the search structure construction in terms of pseudo-code, respectively.

Table 1. Initialization using pruning rules to find Medoids Subsets and the first medoid

Input:	D, a set of samples; n, the number of samples in D; k, the number of medoids; m-block, the number of samples in process using Triangle Inequality
Output:	O, an initialized medoids candidate subset, and Medoid, the first medoid.
	Let $\text{maxdist}(Y, \text{DONE})$ return the two data points with the maximum distance span of examples in Y.
	Let $\text{Closest}(x, Y, \text{m-block}, \text{DONE})$ return the m-block closest examples in Y to x obtained by $\text{maxdist}(Y, \text{Done})$.
	Let $\text{Centroid}(Y)$ return the centroid of examples in Y.
	Let $\text{Distance}(x, y)$ return the distance between samples x, y in Y.
Begin:	
1:	DONE \leftarrow false; $\text{thred_1} \leftarrow \lambda\delta$; // set the cutoff for pruning to $\lambda\delta$ according to Eq.(5)
2:	CENTROID \leftarrow Centroid(D); $x \leftarrow$ Closest(CENTROID , D, 1); DONE[x] = true;
3:	$\text{thred_2} = 0$; for (j=0; j<n; j+ +) { $\text{thred_2} = \text{thred_2} + \text{Distance}(x, x_j)$; }
4:	$O \leftarrow \emptyset$; // initialize to the empty medoids set
5:	for (i=0; i<n; i=i+ m-block) {
6:	(a,b) \leftarrow $\text{maxdist}(D, \text{done})$; Neighbors(a) \leftarrow Closest(a,D,m-block, DONE); Neighbors(b) \leftarrow Closest(b,D,m-block, DONE) upper_bound = 0; lower_bound = 0;
7:	for each c in Neighbors(a) and the same for each d in Neighbors(b) {
8:	upper_bound = upper_bound + fabs(Distance(c,a) + Distance(a,b)); lower_bound = lower_bound + fabs(Distance(c,a) - Distance(a,b));
9:	if (lower_bound>thred_1) {break; DONE[c] = true;}
10:	if ((upper_bound<thred_1) ((upper_bound>thred_1) & (lower_bound<thred_1))) {
11:	sum_dist = 0; for (j=0; j<n; j+ +) { sum_dist = sum_dist + Distance(c, x_j); }
12:	if (sum_dist < thred_1) { O = [O, c]; DONE[c] = true; }
13:	if (sum_dist < thred_2) {thred_2 = sum_dist; medoid = c; DONE[c] = true; } }
14:	return O and Medoid
End	

Lower case variables represent scalar values and upper case variables represents sets.

Table 2. Search structure construction by maxdist(Y, DONE).

Input:	Y, a set of samples; n, the number of samples in Y; m-block, the number of samples in process using Triangle Inequality; Done, the Boolean array remembering the data points not processed yet.
Output:	a; b; //two furthest points whose Boolean labels in DONE are false, that is, not processed yet. dist_array_a; index_array_a; dist_array_b; index_array_b; //their search structures of sorted distance arrays
Let Distance(x,y) return the distance between samples x, y in Y.	
Begin:	
1:	maxdist \leftarrow 0.0;
2:	if (DONE[x ₀] == false) { f \leftarrow x ₀ ; }
3:	for (i=1; i<n; i++) { if (done[x _i] == true) { continue; } }
4:	temp_dist = Distance (f,x _i);
5:	if (temp_dist > maxdist) { maxdist = temp_dist; f = i; }
6:	maxdist \leftarrow 0.0; dist_array_a = []; index_array_a = [];
7:	for (i=0; i<n; i++) { if (DONE[x _i] == true) { continue; } }
8:	temp_dist = Distance (f,x _i);
9:	dist_array_a = [dist_array_a; temp_dist]; index_array_b = [index_array_a; i];
10:	if (temp_dist > maxdist) { maxdist = temp_dist; a = i; }
11:	maxdist \leftarrow 0.0; dist_array_b = []; index_array_b = [];
12:	for (i=0; i<n; i++) { if (DONEe[x _i] == true) { continue; } }
13:	temp_dist = Distance (a,x _i);
14:	dist_array_b = [dist_array_b; temp_dist]; index_array_b = [index_array_b; i];
15:	if (temp_dist > maxdist) { maxdist = temp_dist; b = i; }
16:	sort(dist_array_a; index_array_a); sort(dist_array_b; index_array_b);
17:	return a; b; dist_array_a; index_array_a; dist_array_b; index_array_b;
End	

Lower case variables represent scalar values and upper case variables represents sets.

4 A performance study

In this section, we present the results of an experimental study performed to evaluate the proposed fast INCK (FINCK) algorithm. We evaluate the running time performance of the proposed FINCK algorithm on several large real data sets from UCI machine learning repository to check the technical soundness of this study. All the data sets used in the experiments are briefly summarized in Table 3.

We implemented all the algorithms in C++. All the experiments were performed on a computer with Intel Core i7 2.3GHz CPU and 8GB RAM. The operating system running on this computer is Ubuntu Linux. We use the timer utilities defined in the C standard library to report the CPU time. In our evaluation, the total execution time in seconds accounts for all the phases of the proposed FINCK algorithm and the original

INCK algorithm. The results show the superiority of the proposed FINCK algorithm over INCK in running time performance.

Table 3. Descriptions of all datasets

Data Name	Data Size	Dimension	# of Classes
Localization	164,860	3	11
MiniBooNE	129,595	50	2
Skin	245,057	3	2
Pokerhand	1,000,000	10	10

4.1 Running time performance on large datasets

INCK is one of popular state-of-the-art K-medoids clustering algorithms. The main challenges of INCK reside in its high time and space complexity for large datasets. In this subsection, we compare its actual computational time with the proposed approach on four UCI real datasets as summarized in Table 3. These data sets span a range of problems and have very different types of features. Particularly, in this experiment, we would like to show how the running time of FINCK scales with the number of data points for large data sets so as to show the impact of the proposed pruning rules on the run time performance of INCK algorithm. The results in seconds are shown in Table 4 with $m = 1000$ for the first three UCI datasets, $m = 500$ for the pokerhand dataset, respectively. The stretch factor is set to 1.5 for all the datasets. From the table we can see that the proposed FINCK algorithm runs significantly faster than the INCK algorithm on these datasets.

Table 4. The run time performance for four different real datasets

Data Name	INCK	FINCK	m
Localization	3619	451	1000
MiniBooNE	26620	862	1000
Skin	11192	924	1000
Pokerhand	281772	14899	500

In addition, we are also interested in understanding how the running time scales with m , the number of data points to be processed at each construction of the search structure. As a result, we examine the total running time scalability of the proposed FINCK by varying m from 50 to 5000, and the results for the four datasets are shown in Fig. 5.

Presented in upper plot of Fig. 5 are the running times of the FINCK algorithm on the first three datasets when m is changed from 100 to 5000. Presented in lower plot of Fig. 5 are the running times of the FINCK algorithm on the pokerhand datasets when

m is changed from 500 to 5000. From the plots, it can be seen that the running time first decreases and then increases with m , that is, the number of search structures constructed. This is because the main term that contributes to complexity are $O((N/m)N)$, and, generally, when m is a small integer, the time complexity is near $O(N^2)$. When m becomes larger, searching structure works and the pruning mechanism using triangle inequality takes effects, and therefore, the computational time of FINCK is significant less than INCK. This is one reason that its computational efficiency is lower. However, for larger m , the running time increases eventually because distance computations dominate again.

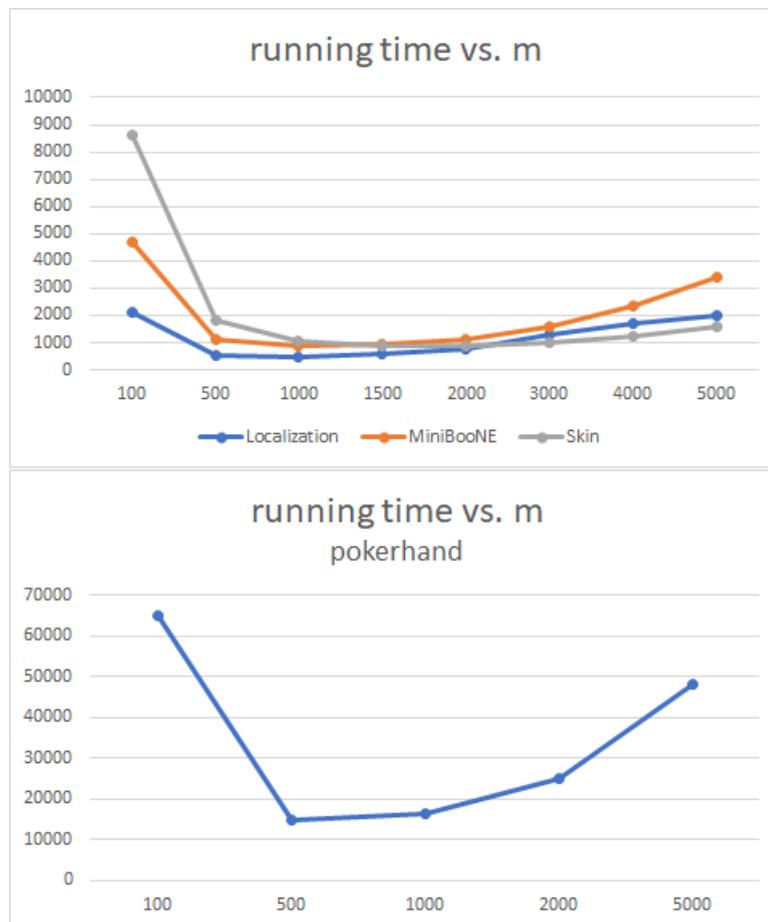


Fig. 5. Run time performance with varying m 's (upper) for the first three datasets (middle) for the pokerhand dataset (lower) for the two image datasets.

The two plots in Fig.5 demonstrate a similar change trend in seconds for different data sets. Compared with the INCK algorithms, the FINCK algorithm showed better running time performance in all data sets.

5 Conclusions

Applying K-medoids algorithms to clustering large scale datasets, it can be difficult for reasons such as high time and space complexity. In this paper, we present an efficient alternative to perform INCK clustering on large scale datasets based on triangle inequality. The proposed approach provides a better run time performance which is extremely useful for modern large datasets. The key innovations of the proposed approach are the use of triangle inequality in the search of the initial medoids candidate subsets and medoids' updating. Therefore, in contrast to INCK, the proposed approach is less computationally intensive, and particularly suitable for large datasets. Future work aims at how to further improve the classification accuracy of the INCK algorithm.

Acknowledgment

The authors would like to thank the National Science Foundation of China for its valuable support of this work under award 61473220.

References

1. Han, J., Kamber, M. and Tung, A. K. H., "Spatial Clustering Methods in Data Mining: A Survey," In H. J. Miller & J. Han (Eds.), *Geographic data mining and knowledge discovery*. Taylor & Francis, 2001.
2. MacQueen, J. B. "Some Methods for Classification and Analysis of Multivariate Observations," *Proc. the 5th Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press. pp. 281–297, 1967.
3. Jain, A.K. "Data Clustering: 50 Years Beyond Kmeans," In: Daelemans W., Goethals B., Morik K. (eds) *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2008*. Lecture Notes in Computer Science, vol. 5211, pp. 3-4, 2008, Springer, Berlin, Heidelberg.
4. Kaufman, L. and Rousseeuw, P.J. "Clustering by Means of Medoids," in *Statistical Data Analysis Based on the Norm and Related Methods*, edited by Y.Dodge, North-Holland, pp. 405-416, 1987.
5. Arumugam, M., Raes J. and Pelletier, E. "Enterotypes of the Human Gut Microbiome," *Nature*, vol. 506, pp. 174–180, 2011.
6. Ohnishi, Y., Huber, W. and Tsumura, A. "Cell-to-cell Expression Variability Followed by Signal Reinforcement Progressively Segregates Early Mouse Lineages," *Nature Cell Biology*, vol. 16, pp. 27–37, 2014.
7. Amor ̂se, D., Bossu, R. and Mazet-Roux, G. "Automatic Clustering of Macroseismic Intensity Data Points From Internet Questionnaires: Efficiency of the Partitioning Around Medoids (PAM)," *Seismological Research Letters*, vol. 86, pp. 1171–1177, 2015.

8. Khatami, A., Mirghasemi, S., Khosravi, A., Lim, C. P. and Nahavandi, S. "A new PSO-based Approach to Fire Flame Detection Using K-medoids Clustering," *Expert Systems with Applications*, vol. 68, pp. 69–80, 2017.
9. Park, H.-S. and Jun, C.-H. "A Simple and Fast Algorithm for K-medoids Clustering," *Expert Systems with Applications*, vol. 36, pp. 3336–3341, 2009.
10. Yu, D., Liu, G., Guo, M. and Liu, X. "An Improved K-medoids Algorithm Based on Step Increasing and Optimizing Medoids," *Expert Systems With Applications*, vol. 92, pp. 464–473, 2018.

Design and Implementation of an Intrusion Detection System using Deep Neural Network

Hyun-chul Chang¹ and Sungbum Park²

¹ Graduate School of Technology Management, Hoseo University, Asan, Korea
suu8@naver.com

² Graduate School of Technology Management, Hoseo University, Asan, Korea
parksb@hoseo.edu

Abstract. This research introduces deep neural network based new intrusion detection method in order to improve existing security systems' performance and the web service vulnerability. We described web application attack techniques and related detection methods proposed in the previous studies. We also discuss the recent trends in the area of machine learning and the detection of security threats based on deep neural networks. In order to conduct research, this paper collected real-time network traffic from Korea's nation-wide web server farm, and introduced an intrusion detection method by various deep neural network techniques for a web applications to identify security threats that bypass the existing intrusion detection method such as the signature-based security model.

Keywords: web application threat detection, CNN, LSTM, C-LSTM

Introduction

As web services are frequently used as the places for new malware or ransomware to be spread, the resulting damage is increasing worldwide. Recently, due to the impact of cloud and mobile environments, the rate of web service attacks to the total number of cybersecurity threats more than doubled [1]. Signature-based intrusion detection (General intrusion detection) methods for responding to various cyber threats are divided into misuse detection models and anomaly detection models according to the detection strategy for intrusion [2]. The Signature-based intrusion detection (misuse detection) model is currently used in many security systems. By analyzing known attack types it identifies and formalizes rules about signatures or patterns specific to a packet to detect them.

However, this approach has two disadvantages; it has a low detection capability for new types of attacks and only detects attacks exceeding certain thresholds in order to reduce false-positive errors. Meanwhile, the anomaly detection model is a method of detecting an attack by judging the data deviating greatly from a rule for a predetermined normal behavior as an anomaly behavior. Although this is an effective method for detecting new types of attacks, it cannot manage to identify detailed types of attacks [3].

In recent years, although research on intrusion detection using the deep neural network in this field is still in an early stage, studies to apply deep neural network or deep learning have also been conducted [4-6]. However, a review of studies reveals that it is difficult to obtain high-quality big sample data for learning, and that the data attacks are concentrated on the specific types of attacks, such as Distributed Denial of Service (DDoS) or information system scanning, which occurs mainly in a network in the lower layer of the TCP/IP model or a transportation layer.

Therefore, the purpose of this research is to study intrusion detection algorithms based on deep neural networks to deal with the cyber threats that are currently difficult to detect in the form of the complex syntax of Hypertext Transfer Protocol (HTTP), a web service protocol which is an upper layer or application layer of the TCP/IP model.

In this idea, we collected network traffic in real-time from one of the biggest nationwide web service server farm switches in Korea. We propose an intrusion detection method for a web application to identify security threats that can bypass or slip through the detection methods of signature-based security systems with this dataset by utilizing various deep neural network techniques.



Fig. 1. The number of world-wide-web application attacks (2016~2017)

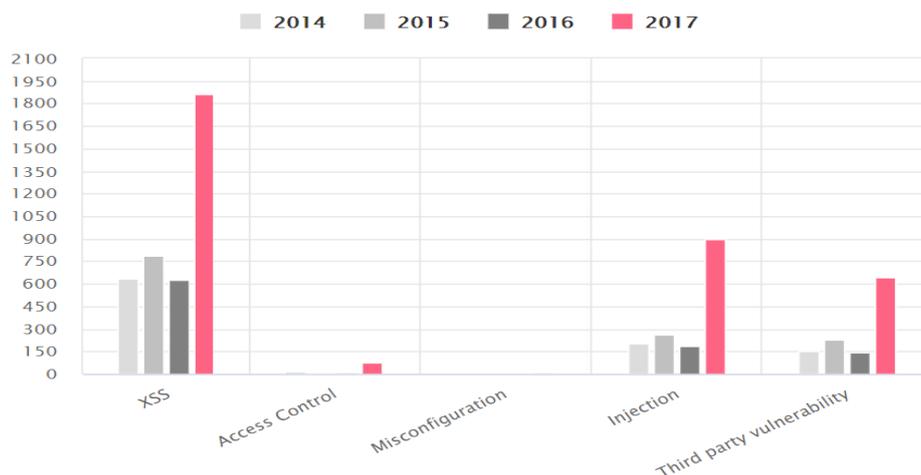


Fig. 2. The number of world-wide-web application attacks by categorization (2014~2017)

Literature Review

The deep neural network was inspired how the human brain is configured, and it performs classification by training the algorithm with a data set in multiple layers in a hierarchical network [7]. Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) are the representative methods of the Deep Neural Network.

In traditional neural networks, it is assumed that the input and output are independent of each other. However, in the case of RNN, the same activation function is applied to every element of a sequence, and thus an output result is influenced by the previous calculation result. However, when RNN is actually implemented, it has a limitation that it handles only relatively short sequences effectively. This drawback is referred to as the problem of Long-Term Dependencies. In order to overcome this problem, Long Short-Term Memory (LSTM), a modified algorithm of RNN, has been proposed. It differs from the basic RNN structure in that it has input, forget, and output gates on each path to calculate the data. Each gate controls the flow of information in the LSTM. The input gate determines the input rate, and the forget gate decides whether to discard or remember the previous information. The output gate determines whether to pass the output of the memory cell. The problem of Long-Term Dependencies has been partially resolved by storing each state value in the memory and reducing unnecessary operations and errors by adjusting the gate part that contacts the data [8]. Kim et al. applied the LSTM architecture to the RNN and trained the intrusion detection system using the Knowledge Discovery and Data Mining (KDD) Cup'99 datasets (Kim et al., 2016). Compared to other intrusion detection classifiers, LSTM-RNN achieved an accuracy of 96.93% and a detection rate of 98.99%.

CNN is a neural network developed to imitate how visual information is processed in humans and animals. The middle and lower layers of CNN consist of a convolution layer and a max-pooling layer. They first extract and abstract features. Then, they gradually extract higher-level features. The final result is calculated from the extracted high-level features. The convolution layer and the max pooling layer are composed of a plurality of feature maps, and each feature map has a structure in which multiple nodes are two-dimensionally arranged. Each node of a max-pooling layer is connected to the input nodes in the window located at a specific coordinate among the connected input feature maps, and selects the maximum value among the values of the nodes to fetch the value as its own value. The top layers of the CNN are fully-connected layers, which determine the final recognition result from higher-level features extracted from lower layers. Each node of fully-connected layers is connected to all nodes of the lower layers. Since CNN extracts high-level features through convolution layers and max-pooling layers, which are very effective for feature extraction, it has recently been reported to show high performance not only in image recognition but also in sentence classification.[9].

As mentioned above, AI-based intrusion detection systems have attracted considerable interest both commercially and in the research community. Especially, RNN (LSTM) and CNN, which are classified as deep neural network models, exhibit excellent classification performance for the classification of unstructured data, such as the classification of images and sentences. As the data size increases, the intrusion detection system

must possess high accuracy and processing capability for noisy data, and a deep neural network has been found to have much better performance in the detection of cyber-attacks.

A recent study by Arnaldo et al. compared the network intrusion detection performances of Feedforward neural network (FFNN), RNN (LSTM), and CNN techniques by training them with the log data collected from a corporate security system. However, their study was still focused on network intrusion detection through the attributes of lower layers (IP address, etc.) of the TCP/IP model [10]. Therefore, with respect to research on the intrusion detection for web applications of syntax structures composed of unstructured letters and numbers of application (or presentation layer) in the higher layers of the TCP/IP model, it is meaningful to compare and analyze the intrusion detection performances of deep neural network models, such as RNN (LSTM) and CNN, in terms of unstructured data classification.

Research Model

Dataset

The data used for the analysis were gathered by a random sampling of the raw data that flowed into the homepage server farm on April 5, 2017. The data sets were divided into the model training set and the test set. Of 14,215 data records in total, 13,942 records were classified as normal behaviors and 273 records as attacks. After 90% of the dataset was extracted as the learning set along with the label data, and the remaining 10% was extracted as the validation set with the label data. An evaluation was conducted every 100 steps.

Research Model Structure

In the dataset in the presentation (or application) layer in the Transmission Control Protocol/Internet Protocol (TCP/IP) model constructed through data preprocessing, the syntax of the text format (HTTP header message information) was converted into vector values. After the vector values were used as the inputs passed through the three models of ①LSTM-RNN, ②CNN, and ③C-LSTM which is a model developed by combining CNN and LSTM, the results were compared with each other. The comparison of the models with other machine learning models such as Decision Tree, Support Vector Machine (SVM), and K-Nearest Neighbor (KNN) was excluded because the input variables to learn are variable. For example, each time input data has a different length of the phrase or the number of words. However, it is a reasonable approach to compare deep neural network models such as CNN or RNN because the value of an input variable is not meaningful but the whole meaning of the phrase consisting of words must be understood. Fig 3 is a simplified representation of each structure of the research model

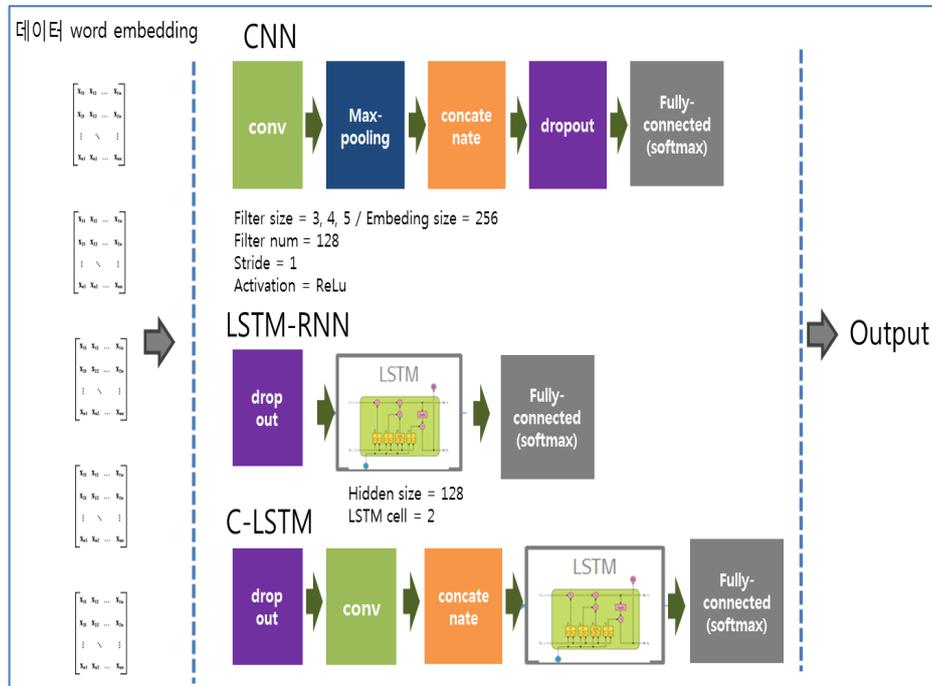


Fig. 3. Research Model Structure

First, words are extracted from the content attribute of the data set by word embedding and they are indexed to construct a lookup table. Embedding refers to mapping words to vector values of a specific dimension and the converted vector values are updated through weighting during learning for the association and distinction of meanings among words.

In the convolution layer used in CNN and C-LSTM, three convolution filters with the sizes of 3, 4, and 5 are used, and there are 128 filters. The feature map is generated by extracting the local information while sliding the filter by 1 pixel at a time (stride=1) and by extracting as many features like the number of filters. While the max-pooling process is performed after feature map creation in the CNN model, the max-pooling process is omitted in the C-LSTM model. If max pooling is performed, it is possible to conduct a sampling of the input value by taking the maximum value in each feature map to map it as the output of a fixed dimension and reduce the dimensions. However, in C-LSTM, the information extracted from the feature map is concatenated without dimension fixation or reduction in that the output value is used as the next input to LSTM.

The LSTM model is comprised of two cells, with 128 hidden units per LSTM cell. Finally, the analysis is as follows: The values obtained from the LSTM are passed through the fully-connected layer, and the score corresponding to each class (0, 1) is calculated. Then, intrusion detection classification is performed, followed by error

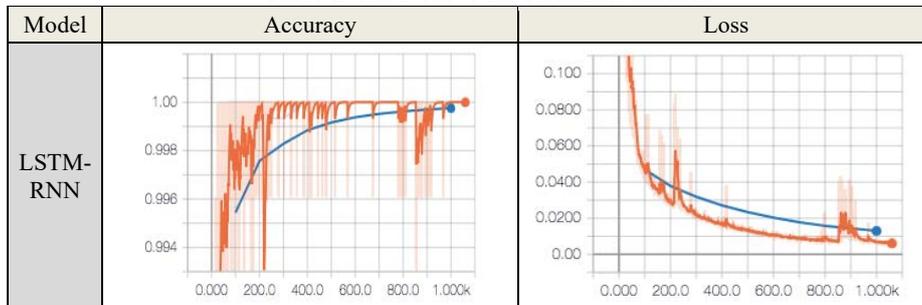
backpropagation for the results, and weights and other parameters are updated. Table 1 shows a summary of the hyper-parameters of the proposed model.

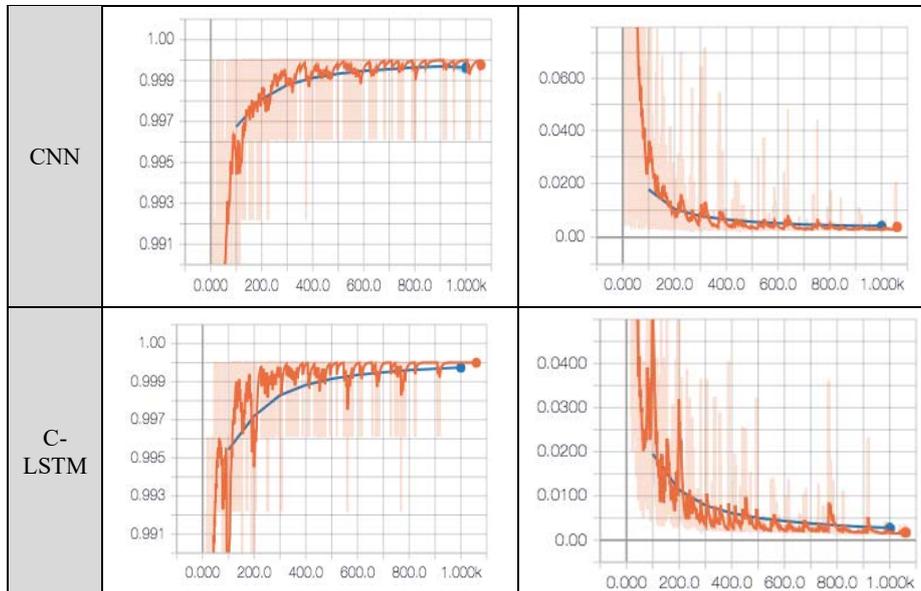
Table 1. Performance Test Result for Research Models

Model	Precision	Recall	Accuracy	F1 score
LSTM-RNN	0.838	0.966	0.997	0.898
CNN	0.899	0.888	0.995	0.893
C-LSTM	0.787	0.793	0.988	0.790

The analysis results were obtained by proceeding with training by dividing a total of 14,215 data records into the sets of 256 input records (batch size = 256) to be learned at a time. This process was repeated 20 times (epoch = 20) and the results of each step (the total number of steps = 989) were averaged. As shown by the results, the LSTM-RNN model showed better overall performance than the other models, with a recall of 0.966, an accuracy of 0.997, and an F1 score of 0.898. In terms of precision, the CNN model showed excellent performance with 0.899 precision. On the other hand, the overall performance of C-LSTM in recall, accuracy, and precision was lower compared to other models. Table 2 shows the ROC curve, which represents the accuracy and loss of each step for each model.

Table 2. ROC curve for Research Models





Conclusion

The construction of an AI-based intrusion detection system is needed to detect anomalous symptoms of large-volume traffic in real time. It is difficult for humans to analyze and to detect better than the intrusion detection systems currently being used.

In this study, it was confirmed that it is possible to conduct the simultaneous collection and analysis of a large amount of traffic data in real-time without delays even in the case of classification of protocols. In addition, it was also shown that the deep neural network technique, which has a good performance in the classification of images or sentences, also shows an excellent performance for the detection of web application intrusions which are not detected by signature-based intrusion detection systems. In addition, due to the nature of the HTTP protocol, which is a representative web service protocol, the length and pattern of HTTP header messages used in attacks are limited and thus there is no need to stack multiple layers, so the burden of processing performance makes it possible to implement an intrusion detection system with excellent performance.

In order to protect key assets safely from evolving cyber threats and to secure competitiveness in the global information security market, changes in the perspective of special institutions for information security and security companies are urgently needed. Further efforts are required to form public and social consensus along with the revision of related laws and regulations are urgently needed. In future research, we plan to conduct a study on AI-based intrusion detection model suitable for the domestic situation

through a case study of overseas cases of constructing an intrusion detection system in the cloud environment as well as technologies and related laws and systems of foreign countries in comparison with the domestic situation.

Acknowledgment

"This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2019-2018-08-01417) supervised by the IITP(Institute for Information & communications Technology Promotion)

References

1. Nadav Avital, N.E. The State of Web Application Vulnerabilities (2017).
2. Noel, S., D. Wijesekera, and C. Youman, Modern intrusion detection, data mining, and degrees of attack guilt, in Applications of data mining in computer security. p. 1-31. Springer (2002)
3. Lee, H.-S., et al., Adaptive Intrusion Detection System Based on SVM and Clustering. Journal of Korean Institute of Intelligent Systems, 13(2): p. 237-242. (2003).
4. Fiore, U., et al., Network anomaly detection with the restricted Boltzmann machine. Neurocomputing, 122: p. 13-23. (2013).
5. Gao, N., et al. An intrusion detection model based on deep belief networks. in Advanced Cloud and Big Data (CBD), 2014 Second International Conference on. IEEE. (2014).
6. Kim, J., et al. Long short term memory recurrent neural network classifier for intrusion detection. in Platform Technology and Service (PlatCon), 2016 International Conference on. IEEE. (2016).
7. Hodo, E., et al., Shallow and deep networks intrusion detection system: A taxonomy and survey. arXiv preprint arXiv:1701.02145, 2017.
8. Hochreiter, S. and J. Schmidhuber, Long short-term memory. Neural computation, 9(8): p. 1735-1780. (1997).
9. Kim In-jung, Deep Learning: New trend of machine learning. Information and Communications Magazine, 31(11): p. 52-57. (2014).
10. Arnaldo, I., et al. Learning Representations for Log Data in Cybersecurity. in International Conference on Cyber Security Cryptography and Machine Learning. Springer. (2017.)

Community detection in networks with fuzzy boundaries

Pradumn Kumar Pandey,¹ T. Ramalingeswara Rao² and Satyajeeet Maharana³

¹ Department of Computer Science and Engineering,
Indian Institute of Technology Roorke, India
pradumn.fcs@iitr.ac.in; pandeypradumn16@gmail.com

² Theoretical Computer Science Group, Department of Mathematics,
Indian Institute of Technology Kharagpur-721302, India
trrao@iitkgp.ac.in

³ Computer Science, Courant Institute of Mathematical Sciences,
New York University, New York, NY 10012
sm8235@nyu.edu

Abstract. Community detection is an emerging and significant research area that plays a key role in detecting patterns in various real-world networks such as social networks, user-item networks, web-log networks and terrorist networks. Many algorithms have been developed for community detection in the recent years. All the existing algorithms perform well when the community structure is well defined. However, when the boundaries between the communities are fuzzy, the existing algorithms show variations in their performances. In this paper, we consider the problem of community detection in networks which have fuzzy boundaries between the communities. We define an objective function motivated from the concept of social harmony of the society and the structure of signed networks. In our work, we propose a method that measures the density of edges against the non-edges using the matrix $(A - \lambda A^c)$ which performs better than other considered state-of-the-art algorithms to identify fuzzy boundaries of community structures. Moreover, we conduct various experiments on bench marked datasets and compare the performance with the existing state-of-the-art algorithms.

Keywords: Fuzzy community structure, Social harmony, Signed representation.

1 Introduction

Due to the rapid growth of social networks, citation networks, web log networks, the research topics relevant to network science have got much significance [2], [5], [27] in the area of complex networks. The study of networks is associated with many disciplines including biology, physics, computer science, sociology, etc, [6], [25], [27]. Networked systems including Internet [6], transportation networks [3], communication networks [23], social networks [9], power transmission networks [1], and biological networks [25] have been examined under the paradigm of network science using the network measures such as degree distribution, average path length, centrality measures, modularity and clustering to find out informative structural patterns [2], [5], [13], [19]. Most of the real-world networks have modular structure [10] and the distribution of edges in the networks are not uniform. Real-world networks have sub-graphs with high

edge density than it-selves. In general, more connections will occur among the nodes of the same sub-graph and less in between the nodes of different sub-graphs. Hence, the densely connected sub-graph of nodes is known as a community [10]. Philosophically, a community is a group of similar nodes in a network. The measure of similarity between nodes could be contextual such as functional similarity, structural similarity or attribute based predefined similarities. A community structure is obtained by the division of nodes into groups that consists of dense connections inside and sparse connections outside the groups in a network [17]. Most of the real-world networks exhibit community structure which affect the behaviour of underlying networked systems. Community detection is a problem of paramount interest to understand and control the diffusion dynamics in real-world networks. Propagating information through online social networks, disease spreading due to the migration of humans and birds, failure cascading prediction in power-grid-networks, scheduling of jobs in parallel computing are some of the examples in real life relevant to information diffusion. To partition a network into communities various algorithms have been proposed in the literature that follows one of the two strategies, namely divide or clustering. The algorithms based on divide approach works on bisecting a network recursively [17], whereas the algorithms adopt clustering strategy do amalgamation of smaller group of nodes into larger groups [4] to improve or optimize a given quality function. Some of the widely used quality measures are as k-cut [24], Normalized Mutual Information (NMI) [7] and modularity index Q [17]. Modularity index measures the aggregate gap between the density of existing edges against the expected edges in the groups of nodes in a given network. Most of the existing algorithms of community detection are not able to detect clear boundary line between communities when the ratio of the internal degree to external degree of the nodes is less than a threshold [21], [12]. Such community structure is called as fuzzy community structure.

In this work, we adopt two different approaches to partition a given network. The first one is inspired by minimization of social conflict in a society (Social Harmony based fuzzy community detection (SHCD)), and the second is based on separation transformation of data points in n -dimensional space (Space Reduction of cliques and Separation Transformation based community detection (SRSTCD)). In Social Harmony based community detection (SHCD), we consider the spectral decomposition of the matrix $(A - \lambda A^c)$, where A is the adjacency matrix corresponding to the network G and A^c is the adjacency matrix corresponding to complement of the network. The benefit of considering the matrix $(A - \lambda A^c)$ for spectral decomposition is that it bisects the network from where the gap between edges to non-edges is more. Considering the matrix A^c in place of P (where $P_{i,j} = \frac{k_i k_j}{2m}$, k_i (k_j) is the degree of the node i (j), and m is the number of edges in the network) in $(A - \lambda A^c)$ sharpens fuzzy boundaries between communities in the network. In Space Reduction of cliques and Separation Transformation based community detection (SRSTCD), we consider the connection vector of the node i including self loop (i^{th} column or i^{th} row vector of the adjacency matrix A including 1's in diagonal) as a data point in n dimensional vector space, where n is the size of the matrix A . Next, apply separation transformation on the data points so that distance between the points from different communities will increase to improve the visibility of fuzzy boundaries between the communities in the network. Again, com-

pute the similarity between each pair of vectors corresponding to data points and obtain a similarity matrix. Now apply the spectral decomposition over the similarity matrix to identify communities in the network. Performances of the proposed community detection algorithms are compared by using community reconstruction capability of the algorithms. We compute the percentage of the nodes which are detected in the wrong communities. It is observed that our proposed algorithms are able to identify fuzzy community structures more accurately when compared to the existing algorithm such as Louvain algorithm [4].

2 Related work

The existing literature on community detection can be divided into two main streams [10]. The first approach is top-down approach in which a network is divided into smaller parts and find out relatively dense graphs known as communities. The second method is bottom-up approach, in which small parts of a network are merged for the same. Modularity index Q is the quality function proposed by Newman [17] that represents the strength of edge connections which causes to separate modules or groups of nodes. Mathematically, the modularity measure is given as

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j),$$

where $A_{i,j}$ is the ij th entry of the adjacency matrix A , c_i is the community in which the node i lie and m is the number of edges in the network. In recent time, most of the community detection methods are developed to maximize the modularity metric. Newman and Girvan proposed a community detection algorithm (GNA) using betweenness centrality of the edges in a network [17]. GNA has gained much attention because of good accuracy but its computation complexity is expensive. Jordi Duch and Alex Arenas [8] proposed a novel algorithm (DA) that optimizes the modularity index Q using a heuristic search based on extremal optimization which provides better results compared to Newman's "fast community detection algorithm" (Fast-Newman) [14]. Scott White and Padhraic Smyth proposed a community detection algorithm (WS) based on spectral clustering [31]. They compared the algorithm with Fast-Newman [14] algorithm. Further, Jianhua Ruan and Weixiong Zhang proposed another modified spectral clustering algorithm (K -cut) [24] and compared with Newman's method of community detection and WS algorithms. They have obtained good reasonable results. Moreover, Newman proposed a new method based on spectral partitioning of modularity matrix B called as Q_{Newman} [15], [16] and achieved better results than Newman's previous methods and DA algorithm [8], [16]. Su, J., Havens, T.C.: Further, Su J et al. [28] developed fuzzy community detection methods by defining a single node in a network can have membership of multiple communities. An extension of modularity which is the combination of Hamiltonian and modularity is proposed to measure the quality of fuzzy community detection [18], [26], [30]. Several extensions of modularity have been proposed for fuzzy community identification in [28], [29], [32]. In this paper, we consider each node in a network has a single and unique membership in communities but the boundaries between the communities are fuzzy.

3 Community Detection Algorithms

In this section, we propose two algorithms for community detection based on the notion of spectral clustering concepts. We now explain the proposed work based on the concepts of conflict minimization and social harmony that play a significant role in the decomposition of a network.

3.1 Social Harmony based community detection Approach (SHCD)

Let A be the adjacency matrix of a social network N in which each entry a_{ij} represents the relation between node i and node j . If x_i and x_j are the actions taken by nodes i and j then the *social harmony* in the network is defined as

$$\chi = \sum_{i,j} x_i a_{ij} x_j = \mathbf{X}^T \mathbf{A} \mathbf{X}. \quad (1)$$

where $\mathbf{X}^T = [x_1, x_2, \dots, x_n]$ is an action vector. To maximize χ , vector \mathbf{X} should be parallel to leading eigenvector v corresponding to the matrix A , which means that according to sign of the entries of the vector v , the network can be divided into two groups of actions of opposite nature that minimize social conflict or maximize social harmony. The process of bisecting a network can be applied to each sub-network recursively until the algorithm meets the termination condition. Now, consider

$$\chi = \mathbf{X}^T \mathbf{A} \mathbf{X} = \mathbf{X}^T \mathbf{A}_+ \mathbf{X} - \mathbf{X}^T \mathbf{A}_- \mathbf{X}. \quad (2)$$

where A_+ and A_- are the adjacency matrices of the social network which have positive relations and negative relations respectively. If A is a positive matrix then there will be no network decomposition on the basis of social conflict theory. Hence, to make this algorithm applicable to the unsigned networks, we use complement network of the given unsigned network that is defined as

$$A^c = O - A - I, \quad (3)$$

where A^c is the adjacency matrix of the complement network of the given unsigned network, O is a all-one matrix (matrix of ones) and I is an identity matrix of appropriate size. Now

$$\chi = \frac{1}{2m} [\mathbf{X}^T \mathbf{A} \mathbf{X} - \lambda \mathbf{X}^T \mathbf{A}^c \mathbf{X}] = \frac{1}{2m} \mathbf{X}^T \mathbf{B} \mathbf{X}, \quad \mathbf{B} = \mathbf{A} - \lambda \mathbf{A}^c, \quad (4)$$

where λ is a positive constant and m is the number of positive edges in the network. In the function χ , free parameter λ tunes the level of bisection of the network. If we require clique level decomposition of a network, then the value of λ should be large enough. As the value of λ increases from zero to n (size of the network), partition level of the network reaches up to the cliques of the network. During the decomposition, if a subgraph contains two cliques of different sizes, it preserves the structure of bigger clique. Let s and $(s-1)$ be the sizes of two cliques in a network of size $(s+1)$. There is a single negative link in the network. In that link, participating nodes are from different

cliques and all other nodes are common in both the cliques. If the value of λ is large enough, then the node which is connected with the negative link and is the part of the smaller clique separates from the network to maximize χ . In Fig. 1, we consider an example of 5 nodes in which nodes 2, 3, 4, 5 are the part of the bigger clique and 1, 4, 5 are the part of the smaller clique. There exists a negative edge between nodes 1 and 2. For the suitable value of the parameter λ , separation of node 1 happens to maximize χ .

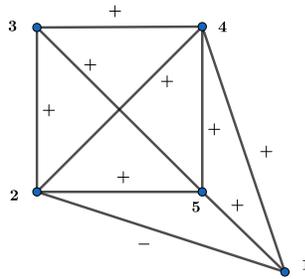


Fig. 1: An example network of 5 nodes.

Algorithm 1 Social Harmony based community detection (SHCD)

Input: Given a network $G(V, E)$.

Output: Set of communities \mathcal{C} .

Initialize $G \in \mathcal{C}$.

For (each network/sub-graph $G' \in \mathcal{C}$)

Find the adjacency matrix A corresponding to G' .

Compute $A^c = O - A - I$, $\mathcal{B} = A - \lambda A^c$, where $\lambda = \frac{\mathbf{1}^T A \mathbf{1}}{\mathbf{1}^T A^c \mathbf{1}}$.

If ($\max_eig(\mathcal{B}) > 0$)

Let v be the eigenvector corresponding to the maximum positive eigenvalue of the matrix \mathcal{B} .

Divide the network/sub-graph G' into two sub-graphs G'_1 and G'_2 such that node $i \in G'_1$ **if** $v(i) \geq 0$ **else** $i \in G'_2$.

$\mathcal{C} \leftarrow \mathcal{C} \cup \{G'_1, G'_2\}$.

$\mathcal{C} \leftarrow \mathcal{C} \setminus \{G'\}$.

End(If)

End(For)

Let a pair $G'_1, G'_2 \in \mathcal{C}$, merge them in a single community **if** modularity index Q gets increased (repeat until no more merging is possible).

Now, we discuss another method to define community detection algorithm known as SRSTCD algorithm.

3.2 Space Reduction and Separation Transformation approach (SRST)

In this approach, we consider the nodes of a network as data points in a vector space represented by their connection vectors in \mathbb{R}^n . A row or column of an adjacency matrix is the connection vector of the corresponding node. Consider an example network which has two isolated triangles and the adjacency matrix of the network is as follows

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \quad \mathcal{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}, \quad \Theta = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

The matrix A has 6 different columns which are in different directions but the example network has only two communities (cliques). If we put 1's in the diagonal of the matrix, we get only two vectors that are perpendicular.

Two representative vectors corresponding to cliques can be visualized. In the network, the diagonal elements of A are unit elements which represents generating self loops. This process is named as *space reduction* of cliques or *rank reduction* of vector space. Now, we compute the cosine of the angles between nodes i and j (vectors) that is given by

$$\Theta_{i,j} = \frac{\mathcal{A}_{i,:} \cdot \mathcal{A}_{:,j}}{\|\mathcal{A}_{i,:}\| \|\mathcal{A}_{:,j}\|}.$$

We have a matrix Θ which represents the cosine of angles between any two nodes in n dimensional vector space.

The matrix Θ has all positive entries, because all the nodes (vectors) are in the space such that the cosine of the angles between a pair of vectors is positive (less than or equal to perpendicular). The Θ matrix has the same problem as adjacency matrix, because both of them have positive leading eigenvectors. To handle this problem, we move one step back in the process. We modify the adjacency matrix of the network to obtain space matrix \mathcal{A} in which the two vectors corresponding to two cliques are perpendicular. Now in \mathcal{A} , we replace zeros by -1 and get new matrix M which still has two vectors that are anti-parallel to each other. This time, separation between points is maximum and the process is called as a *separation transformation*.

$$M = \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 \end{bmatrix}, \quad \Theta = \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 \end{bmatrix}$$

Again, we compute cosine of angles between nodes (vectors) in space represented by the column vectors of matrix M that is given by

$$\Theta_{i,j} = \frac{M_{i,:} \cdot M_{:,j}}{\|M_{i,:}\| \|M_{:,j}\|}.$$

Now in this case the matrix Θ is more informative and spectral decomposition of the matrix Θ provides the clear separation of two cliques of the network.

We now generalize this process for any network. Let A be the adjacency matrix of a network without self loop. Now, Matrix M is defined by $M = A + I - \alpha A^c$, where α is a positive constant which is used to tune the separation between communities of the network in the space. In our work, we consider $\alpha = \frac{\mathbf{1}^T(A+I)\mathbf{1}}{\mathbf{1}^T A^c \mathbf{1}}$. Also, we compute the matrix Θ and apply spectral decomposition over the matrix Θ for detecting communities.

One of the issue faced by community detection algorithms based on modularity maximization is resolution limit that is discussed in the next section.

3.3 Resolution limit problem

Consider a network which has m edges and two isolated cliques that are the parts of the network. Let $\delta = \frac{4}{2m}$. If m is large enough then $\delta \rightarrow 0$ and the structure of the modularity matrix for the isolated cliques is given as

$$Q_6 = \begin{bmatrix} -\delta & 1-\delta & 1-\delta & -\delta & -\delta & -\delta \\ 1-\delta & -\delta & 1-\delta & -\delta & -\delta & -\delta \\ 1-\delta & 1-\delta & -\delta & -\delta & -\delta & -\delta \\ -\delta & -\delta & -\delta & -\delta & 1-\delta & 1-\delta \\ -\delta & -\delta & -\delta & 1-\delta & -\delta & 1-\delta \\ -\delta & -\delta & -\delta & 1-\delta & 1-\delta & -\delta \end{bmatrix} \approx \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} = A_6.$$

This does not show any significant improvement in modularity measure Q after the division of network into small groups. The *resolution limit* [11] depends on the number edges of the network that has $\mathcal{O}(n)$ as the lower bound. As the size of the network increases, resolution of the community detection decreases. In the next section, we discuss our defined objective function, social harmony χ , and do comparison with modularity index.

4 Social harmony (χ) for network decomposition

In this section, we compare the well-known quality function Q to the function χ defined in Section 3. One of the drawback of the modularity measure is resolution limit and we have already discussed about the resolution limit of the modularity index in Section 3.3. Initially, we introduce the concept of perfect community/module.

Definition 1. *An isolated clique or complete network is considered as a perfect community/module.*

Modularity measure does not consider a complete graph as a perfect graph. The value of quality function Q for a complete network is zero. The modularity of Collection of \mathcal{C} cliques $(1 - \frac{1}{\mathcal{C}})$ depends on the number of cliques. According to the definition of perfect module, collection of two cliques is as modular as collection of multiple cliques.

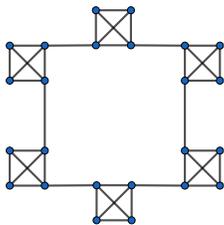


Fig. 2: Example of a caveman network.

If we have two networks in which, one is the combination of two independent cliques, and other has two independent modules of almost perfect cliques, then both have same value of modularity function Q which is not able to detect the structural differences of these two networks. In an example of community detection in caveman networks, in Fig. 2, decomposition of the network based on modularity maximization considers two cliques in the same community. Modularity maximization detects three communities in the considered example of caveman network as shown in Fig. 2, but it is a collection of six communities. These drawbacks restrict the applicability of the modularity index.

To improve the resolution limit of the algorithms, based on Q index [22], researchers modified the modularity function by introducing a parameter λ such that

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \lambda_1 \frac{k_i k_j}{2m} \right] \delta(c_i, c_j).$$

An important observation is that in modularity matrix $B = A - P$, where $P_{i,j} = \frac{k_i k_j}{2m}$, the dispersion of the negative values is more (over the positive links also) when compared to proposed defined matrix χ , which considers the dispersion of negative values over non-links. Also, which improves the visibility and sharpens the boundaries between communities.

Community detection based on χ has more advantage over modularity matrix for the above mentioned drawbacks. However, it can detect cliques at the lowest level of decomposition. In caveman network, each clique represents as a community. A complete network or the collection of cliques has maximum χ value 1. In case of unsigned networks, for the function χ , the penalty part comes from the non-existing links that are to be replaced by negative links. Unlike in modularity matrix B , the function χ represents the weighted difference in positive and negative edges inside the community. An example of two isolated triangles has same decomposition irrespective of the structure of the network that contains it. In Eq. (4), the parameter λ increase the utility of the function χ . It can be tuned accordingly to increase the level of decomposition up to cliques. λ can be used to identify dense sub-graphs inside a network along with community detection application. We now conduct various experiments based on our proposed algorithms and compare with some of the state-of-the-art community detection algorithms.

5 Simulation and Results

In the previous sections, we have discussed the importance and applicability of χ in real-world problems and the limitations of modularity function in detecting the communities of a network. We now conduct various experiments particularly for the networks, where the accuracy of community detection is more important than computational complexity. Two important measures are used to compare the quality of community detection algorithms. These are modularity index Q and percentage of wrong identification of nodes inside the communities. We use computer-generated benchmark networks and real-world networks that are already used in literature to measure the accuracy of the existing algorithms for community detection.

We compare the performance of the proposed methods with the well-known state-of-the-art algorithms using modularity index Q . In this analysis, we consider Girvan-Newman algorithm (GNA) which detects communities using dynamic betweenness of edges of the network. Also, we use the fast algorithm of Clauset *et al.* (CNM) which is a greedy approach to find out the community structure, algorithm based on external optimization given by Duch and Arenas (DA), Newman's Modularity (Q) maximization algorithm (Q_{Newman}) and fast algorithm for community detection by Newman (Fast-Newman). We also compare some cases with the algorithms proposed by White *et al.* (WS) and Jianhua Ruan *et al.* (K-cut). In each case, we compare Social Harmony based community detection (SHCD) and SRSTCD against the best performing community detection algorithm from the above mentioned (GNA, Fast-Newman, WS) algorithms.

5.1 GN benchmark networks

GN benchmark networks are used to evaluate the performance of community detection algorithms. GN networks are fixed size networks with 128 nodes and 4 communities with same size. Let p_{in} be the probability of link formation between the nodes of same community and p_{out} be the probability of connection formation between the nodes of different communities. The total degree of a GN benchmark network is a fixed value 16. Now, p_{in} and p_{out} can be changed and the ratio of external degree of a node to its total degree is known as mixing parameter μ . For lower values of μ almost all the algorithms are accurate. As the value of μ increases beyond $\mu > 6/16$, the accuracy of these algorithms changes, significantly. They are not able to detect clear boundary of communities. We compare our methods SHCD and SRSTCD with the algorithms that are Louvain, Fast-Newman, Girvan-Newman in the following subsections.

We observed that the methods SHCD and SRSTCD defined in this paper can compute community structure better than Louvain algorithm (see 1) and the error in reconstruction of the community structure is negatively correlated with the modularity index in the considered networks. Each value is averaged over 100 networks for each mixing parameter μ . When $\mu = 8/16$, the error of community reconstruction is almost half in case of our methods when compared to Louvain algorithm. There is a non-linear relation between the error in community reconstruction and the modularity index, but the modularity index has negative correlation with the error. In other examples we use that observation and compute modularity index only to show the performance of the methods considered in this paper.

μ	SHCD, Er (Q)	SRSTCD, Er (Q)	Louvain, Er (Q)
6/16	0 (0.3934)	0 (0.3934)	0.42 (0.3914)
7/16	1.25 (0.3311)	0.77 (0.3317)	5.85 (0.3176)
8/16	23.76 (0.2568)	22.35 (0.2574)	40.35 (0.2361)
9/16	52.17 (0.2241)	53.20 (0.2240)	60.45 (0.2127)

Table 1: Error (Er) in community reconstruction [20] and modularity (Q) in GN benchmark networks.

5.2 Zachary’s karate club network:

This is a real-world network of 34 nodes. Zachary observed the interaction between the member of a karate club around 1970 and constructed a network. Unfortunately, later on the club is divided into two small clubs due to disputes between administrator and principal instructor. One group centred around administrator and the other around instructor that are node 1 and node 33, respectively as shown in Fig.3. Here a natural division is present. Now using our algorithm (SHCD), we got four groups for the network. There is no wrong selections. Fig. 3 shows red and green coloured nodes are in small club of instructor, while purple and blue coloured nodes belong to second small club. Therefore, SHCD got the maximum modularity index among the other considered algorithms.

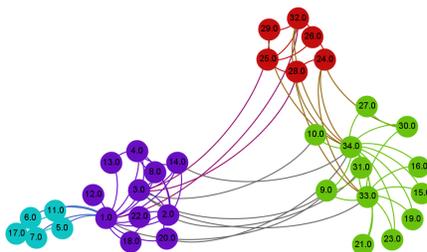


Fig. 3: Division of karate network using SHCD. It divides the whole network in 4 groups. There is no conflict between the natural division of karate-club and network partition using SHCD.

Apart from karate club network there are other examples that are used as experimental data for community detection algorithms. Largest component of a network of collaborations between physicists who conduct research on networks (CbN), social network of dolphins, network of web pages (WebN), football network, the network of interactions between major characters in the novel *Les Misérables* (Lesmis) by Victor Hugo.

5.3 Collaboration Network (CbN):

Collaboration network is the network between physicists who are working on networks. Nodes represent the physicists and two nodes are connected if they are the co-authors of atleast one paper in networks [17]. GNA optimizes 0.72 modularity value of the network partitions with 13 communities. SHCD achieved 0.7427 modularity with 10 community (in Fig. 4). In this case the performance of Fast-Newman is poor ($Q = 0.7011$).

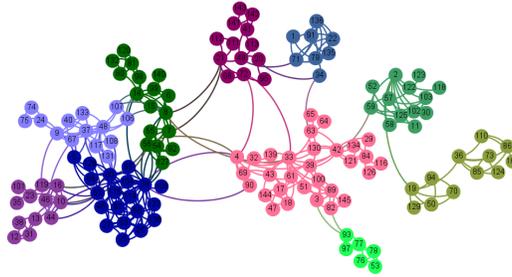


Fig. 4: Community structure of collaboration network of physicists who conduct research on networks. The large component of the network has 145 nodes. SHCD achieved 0.7427 modularity index.

5.4 Network of web pages (WebN):

Web network is an example of a non-social network. It is a network of 180 web pages from the web site of a large corporation [17] and hyper links represent the edges between them. GNA achieves 0.65 modularity of for the network with 8 communities [17]. Also, SHCD achieves 0.6550 modularity with 7 communities. However, SRSTCD is better than SHCD with 8 communities with 0.6592 modularity index. The performance of Fast-Newman is better than GNA. Fast-Newman does partition of the network with 0.6548 modularity which is closer to SHCD.

5.5 Lesmis Network (lesmis):

Lesmis is the network between the characters in the novel *Les Misérables* (lesmis) by Victor Hugo. Using the list of character appearances by scene compiled by Knuth [17], the network is constructed in which the vertices represent characters and an edge between two vertices represents co-appearance of the corresponding characters in one or more scenes. Optimum modularity value is 0.54 while community detection is performed by GNA. It detects 11 communities. Here also, Fast-Newman is not performing well. SHCD achieves 0.5596 modularity and detected 7 communities (in Fig. 5), while SRSTCD find out 6 communities and 0.56 modularity value.

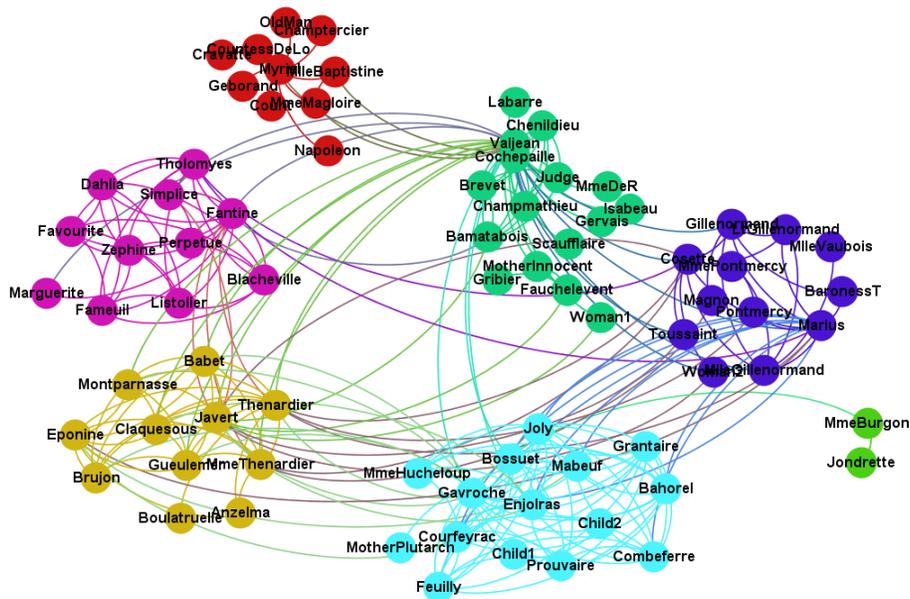


Fig. 5: Community structure of the network of the characters in the novel *Les Misérables* (Lesmis) by Victor Hugo. SHCD achieved 0.55960 modularity index.

5.6 Social network of bottlenose dolphins of Doubtful Sound (dolphins):

The Social network of dolphins is a non-human social network with 62 nodes. A node represents a dolphin and two nodes are linked if dolphins associated with nodes have significant frequent association. The network splits naturally into two groups. GNA splits the network into five groups with 0.52 modularity index. Fast-Newman divides the network into four parts with 0.4955 modularity index. SHCD achieves 0.5213 modularity index with five groups (see Fig. 6) and SRSTCD partition the network in to four parts and achieves 0.5224 modularity index.

The Modularity of partition of dolphins' network becomes 0.5269 if node 8 and 20 get shifted to the module in which nodes 2 and 28 lie naturally. This observation encourage us for further optimization. But shifting of multiple nodes in a single step increases the computational complexity. The shift done in case of dolphins' network has a noticeable point. Shifted nodes 8 and 20 are directly connected (In Fig. 6 nodes inside the red circle) and belong to the same community. To make the optimization practically feasible, we adopt the shifting of links (two directly connected nodes) from one community to another.

In Fig. 6, the black vertical line shows the natural separation of the network. It is observed that the modularity maximization is still supported the natural division. For sparse graph, further optimization can be obtained using the edge shift.

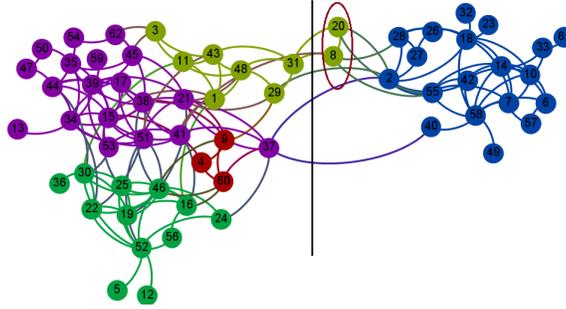


Fig. 6: Community structure in the bottlenose dolphins of Doubtful Sound using SHCD. It divides the whole network in 5 groups.

5.7 American college football network (football):

This network consists of 115 football players [17] and 12 ground truth communities. Many community detection algorithms have been tested over this network. WS algorithm performs well among the considered algorithms in this paper. WS detected 11 communities with 0.602 modularity index. k-cut achieved 0.6 modularity value. Our proposed SHCD method is better than others by detecting 10 communities with highest modularity value 0.6046.

5.8 Krebs' network of books on American politics (polbooks):

This network consists of 105 books on American politics bought by on-line buyers from Amazon.com. In this graph, nodes represent the books and two nodes are connected if these books are frequently purchased by same buyer. Books are divided into three groups, liberal (l) conservative (c) and neutral (n). We have applied SHCD in this network and found the community structure given in Fig. 7. We have applied Q_{Newman} [15] over the same network that places the circled node as shown in Fig. 7 in liberal community. SHCD is able to identify the circled node in Fig. 7 in right community.

6 Conclusion

Community detection is a significant research area due to the rapid growth of online social networks, biological networks, transportation networks and various kinds of graph based networks. In this paper, we presented simple yet effective ideas to sharpen the fuzzy boundaries between the communities in networks and proposed two algorithms for community detection known as SHCD and SRSTCD. The algorithms SHCD and SRSTCD performs better than the fundamental community detection algorithms. Also, maximization of the proposed objective function leads to maximization of modularity index. We have used the concept of social harmony, space reduction and space transformation, and complement network for a given network in identify the fuzzy boundaries

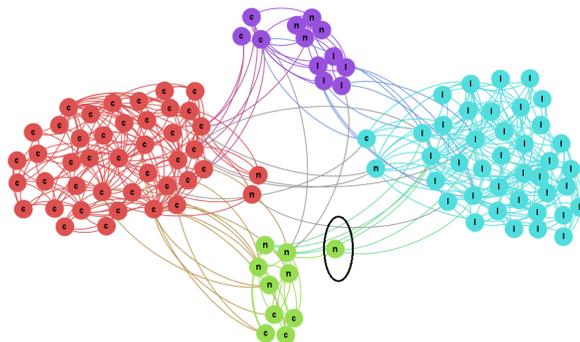


Fig. 7: Community structure of the network of 105 books (polbooks). SHCD divided the network into 4 groups.

between the communities more accurately. Implementing distributed approach of the proposed algorithms using efficient data structures such as edge list, and multi-hop neighborhoods over large networks is a future problem relevant to our work.

References

1. Albert, R., Albert, I., Nakarado, G.L.: Structural vulnerability of the north american power grid. *Physical review E* **69**(2), 025103 (2004)
2. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. *Reviews of modern physics* **74**(1), 47 (2002)
3. Bell, M.G., Iida, Y.: *Transportation network analysis* (1997)
4. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* **2008**(10), P10008 (2008)
5. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U.: Complex networks: Structure and dynamics. *Physics reports* **424**(4), 175–308 (2006)
6. Caldarelli, G.: *Scale-free networks: complex webs in nature and technology*. OUP Catalogue (2007)
7. Danon, L., Díaz-Guilera, A., Arenas, A.: The effect of size heterogeneity on community identification in complex networks. *Journal of Statistical Mechanics: Theory and Experiment* **2006**(11), P11010 (2006)
8. Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. *Physical review E* **72**(2), 027104 (2005)
9. Ellison, N.B., et al.: Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication* **13**(1), 210–230 (2007)
10. Fortunato, S.: Community detection in graphs. *Physics Reports* **486**(3), 75–174 (2010)
11. Fortunato, S., Barthelemy, M.: Resolution limit in community detection. *Proceedings of the National Academy of Sciences* **104**(1), 36–41 (2007)
12. Mahmood, A., Small, M.: Subspace based network community detection using sparse linear coding. In: *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. pp. 1502–1503. IEEE (2016)

13. Newman, M.E.: The structure and function of complex networks. *SIAM review* **45**(2), 167–256 (2003)
14. Newman, M.E.: Fast algorithm for detecting community structure in networks. *Physical review E* **69**(6), 066133 (2004)
15. Newman, M.E.: Finding community structure in networks using the eigenvectors of matrices. *Physical review E* **74**(3), 036104 (2006)
16. Newman, M.E.: Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* **103**(23), 8577–8582 (2006)
17. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. *Physical review E* **69**(2), 026113 (2004)
18. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**(7043), 814 (2005)
19. Pandey, P.K., Badarla, V.: Reconstruction of network topology using status-time-series data. *Physica A: Statistical Mechanics and its Applications* **490**, 573–583 (2018)
20. Pandey, P.K., Bhattacharya, S., Ganguly, N.: Non-link preserving network embedding using subspace learning for network reconstruction. In: *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*. pp. 10–17. ACM (2019)
21. Radicchi, F.: A paradox in community detection. *EPL (Europhysics Letters)* **106**(3), 38001 (2014)
22. Reichardt, J., Bornholdt, S.: Statistical mechanics of community detection. *Physical Review E* **74**(1), 016110 (2006)
23. Rogers, E.M., Kincaid, D.L.: *Communication networks: Toward a new paradigm for research*. (1981)
24. Ruan, J., Zhang, W.: An efficient spectral algorithm for network community discovery and its applications to biological and social networks. In: *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. pp. 643–648. IEEE (2007)
25. Rubinov, M., Sporns, O.: Complex network measures of brain connectivity: uses and interpretations. *Neuroimage* **52**(3), 1059–1069 (2010)
26. Simon, H.A.: The architecture of complexity. In: *Facets of systems science*, pp. 457–476. Springer (1991)
27. Sporns, O.: The human connectome: a complex network. *Annals of the New York Academy of Sciences* **1224**(1), 109–125 (2011)
28. Su, J., Havens, T.C.: Fuzzy community detection in social networks using a genetic algorithm. In: *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*. pp. 2039–2046. IEEE (2014)
29. Sun, P.G.: Community detection by fuzzy clustering. *Physica A: Statistical Mechanics and its Applications* **419**, 408–416 (2015)
30. Wang, Q., Fleury, E.: Community detection with fuzzy community structure. In: *2011 International Conference on Advances in Social Networks Analysis and Mining*. pp. 575–580. IEEE (2011)
31. White, S., Smyth, P.: A spectral clustering approach to finding communities in graph. In: *SDM*. vol. 5, pp. 76–84. SIAM (2005)
32. Zhang, H., Chen, X., Li, J., Zhou, B.: Fuzzy community detection via modularity guided membership-degree propagation. *Pattern Recognition Letters* **70**, 66–72 (2016)

Precise Feature Selection and Case Study of Intrusion Detection in an Industrial Control System (ICS) Environment

Terry Guo, Animesh Dahal, and Ambareen Siraj

Tennessee Tech University, Cookeville, TN 38505, USA
tguo@tntech.edu; adahal42@students.tntech.edu; ASiraj@tntech.edu

Abstract. This paper presents analytical techniques to improve redundancy and relevance assessment for precise selection of features in practical multi-class raw datasets. We propose a matrix-rank based k -medoids algorithm that guarantees to output all independent medoids. The new algorithm uses matrix rank as a robust indicator, while a traditional k -medoids algorithm depends on specific datasets and how the distance between any of two features is defined. Another advantage is that the total number of operations in the nested loops is bounded, different from some k -medoids algorithms that involve random search. Sparse regression is an efficient tool for feature relevance analysis, but its outcome can depend on what labeled datasets are employed. A compensation method is introduced in this paper to handle the inequality of class-occurrence in a practical raw dataset. To assess the proposed techniques quantitatively, an existing Industrial Control System (ICS) dataset is used to perform intrusion detection. The numerical results generated from this case study validate the effectiveness and necessity of the proposed analytical framework.

Keywords: Feature selection · k -medoids clustering · $l_{2,1}$ -norm minimization · Industrial Control Systems (ICSs) · intrusion detection.

1 Introduction

Accurate selection of the features in an experimental dataset is the key to successful classification. To use the features wisely, it is necessary to identify the “right” features that can lead to reduction in run time and/or improvement of classification performance. The process of selecting a subset of relevant features from a large set of features is called feature selection which can often times yield an efficient learning model [1]. As mentioned in [2], feature selection can be used in data from various fields to create a fast and efficient learning model, for example to quickly discover key genes from a large number of candidate genes in biomedical problems [3], to investigate representative features that describe the dynamic business environment [1], to identify key terms like words or phrases in text mining [4], and to choose and construct important visual compositions like shape, texture, pixel and color in image analysis [5]. Similarly, feature selection can be used to build efficient intrusion detection system by selecting most important features [6].

* Supported by Cybersecurity Education, Research and Outreach Center(CEROC), as well as Center for Manufacturing Research (CMR), both at Tennessee Tech University.

Features can be categorized into three groups: relevant features, irrelevant features and redundant features, note that a relevant feature can be redundant as well. It is desirable to identify and eliminate redundant and irrelevant features in a dataset of interest. In general, these issues are related to “feature selection” [1, 7–27]. Feature selection enables development of simpler and faster learning algorithms by saving memory and eliminating irrelevant features. The removal or selection of such relevant yet redundant features may lead to sub-optimal or optimal feature subset, making feature selection a tricky task [2]. There are many existing feature selection methods, and they can be categorized into filters, wrappers, embedded and others [20, 24]. However, filter and wrapper based techniques are the two representative approaches to feature selection [2]. The wrapper approach includes a classification/learning algorithm in the feature subset evaluation step which is used to evaluate the goodness of the selected features. Whereas, the filter approach is not dependent on any classification algorithm. Generally, filter approaches tend to be computationally less expensive compared to wrapper approaches [8, 28, 29]. Our technique is a filter based feature selection approach which is suitable for effective and efficient dimensionality reduction in a high dimensional dataset. It needs to be pointed out that in literature the two issues related to feature selection, redundancy and relevance, may not be handled at the same time. In [19, 30] both relevance and redundancy are taken into account in spectral feature selection at relatively high computation. In this paper we consider supervised feature selection and deal with the problem by conducting two separated tasks: redundancy analysis and relevance analysis.

The fundamental idea for redundancy analysis is distance (or similarity) based clustering. In general, k -medoids clustering with predefined distance measure can partition features into clusters based on the distances between them [12–14, 31–33]. However, the performance of k -medoids clustering depends on what specific dataset is used and how a distance measure is defined [14, 34]. In addition, the number (k) of clusters is a critical predetermined parameter to most clustering algorithms, but it is not straightforward to determine its value. Simplified Silhouette Filter (SSF) [9, 12, 14] is a clustering method that does not need to know the number of clusters in prior. However, it is found that this method is computationally expensive and not quite robust. In this paper we propose an alternative clustering technique that relies on measuring matrix rank thus is more robust. The proposed feature matrix rank based k -medoids clustering algorithm does not need an exhaustive search to determine parameter k . Moreover, the algorithm has a bounded complexity.

A feature, even if it is not redundant, could be irrelevant to a classification task. Evaluating feature relevance is as important as assessing feature redundancy in feature selection. Recently, sparse regression based feature relevance analysis has drawn attention [15–17, 22, 25, 26]. Algorithms in this subset belong to embedded feature selection category and typically exhibit both efficiency and tractability. For a given dataset with labels, one hidden parameter is the class occurrence, i.e., the number of instants that are associated with a particular class. As verified by experiment, class occurrences do affect analysis result. We introduce a compensation method that can be integrated with existing sparse regression framework for relevance analysis.

Industrial Control Systems (ICSs) of the past have been shielded from network intrusions by means of an “air gap” separating the system from the open internet. However, this protection is no longer universally present in modern networked ICSs. There has been a growing demand for designing protection mechanisms against various attacks on the ICSs, and intrusion detection is one of such mechanisms. The proposed feature selection techniques are examined by using a case study of ICS intrusion detection.

Major Contributions in this work include:

1. Proposal of a matrix-rank-preserving k -medoids algorithm which is more robust and has a bounded complexity;
2. Proposal of a class-occurrence compensation technique integrated with the $l_{2,1}$ -norm minimization framework to ensure fairness of feature relevance analysis.
3. Experimental validation of the proposed techniques.

The rest of the paper is organized as follows. The feature redundancy analysis including a matrix-rank-preserving k -medoids algorithm is provided in the next section. Section III introduces the compensation for fair assessment with the sparse regression based feature relevance analysis. A case study of ICS intrusion detection is given in Section IV to generate numerical results and validate the proposed techniques. Section V summarizes our work and presents some remarks.

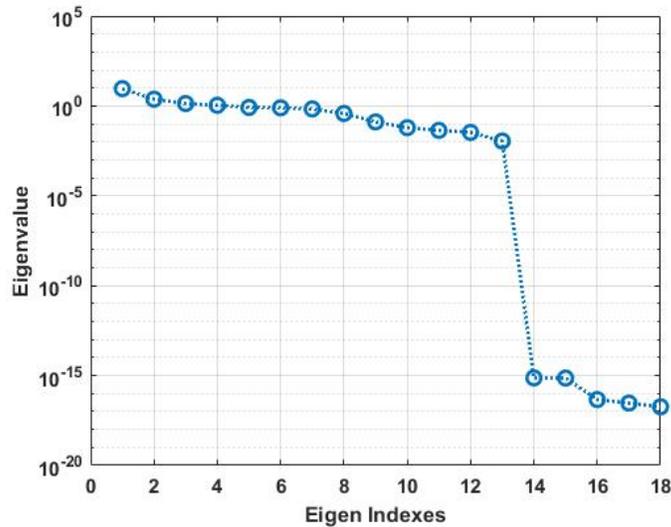


Fig. 1. Eigen spectrum of the water storage tank dataset [35].

2 Matrix-Rank Based Redundant Feature Identification

In this paper we propose an alternative clustering technique that relies on measuring matrix rank thus is more robust and accurate. In the analysis below a given dataset is

represented as either an $m \times n$ matrix $F = (\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m)^T \in \mathbb{R}^{m \times n}$ or an m -member set $\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m\}$, where each member represents a feature, m is the number of features and n is the number of instants. The rank of matrix FF^T/n (the sample covariance matrix of the feature dataset) tells how many significant eigen modes F contains. For instance, from the eigenvalue spectrum (shown in Fig.1) of the water tank data matrix we can say that all the information embedded in the feature matrix can possibly be represented by as less as 13 independent features.

Algorithm 1 Matrix-rank-preserving k -medoids algorithm

Inputs: data matrix F .

Initialization: $C_1 = C_2 = \dots = C_m = \Phi$; $F_0 = \mathbf{0}$; $S = F$; $k = 0$.

Result: k clusters and k medoids.

Phase-1: Find all k clusters \mathcal{C}_j , $j = 1, 2, 3, \dots, k$.

while S is not empty **do**

$k \leftarrow k + 1$;

 remove one row from S and denote it by \mathbf{s}_0 ;

 add \mathbf{s}_0 to cluster \mathcal{C}_k ;

$F_0 \leftarrow F_0 \boxplus \mathbf{s}_0$;

$i \leftarrow 1$;

$len \leftarrow$ number of row in S ;

for $r = 1$ to len **do**

 take one row from S and denote it by \mathbf{s}_i ;

if $rank(F_0) = rank(F_0 \boxplus \mathbf{s}_i)$ **then**

 add \mathbf{s}_i to cluster \mathcal{C}_k ;

 remove one row from S ;

else

$i \leftarrow i + 1$;

end if

end for

end while

Phase-2: Determine k medoids.

for $j = 1$ to k **do**

if $|\mathcal{C}_j| \geq 3$ **then**

 choose a member from \mathcal{C}_j as the cluster medoid such that the sum of its distances to its neighbors is minimal;

else

if $|\mathcal{C}_j| = 2$ **then**

 randomly choose one of the two members in \mathcal{C}_j as the cluster medoid;

else

 the sole member of \mathcal{C}_j is the cluster medoid;

end if

end if

end for

* Symbol “ \boxplus ” represents attaching a row to a matrix.

The proposed algorithm is shown in **Algorithm 1** and it relies on the following facts. Let \tilde{F} be a $p \times n$ matrix that contains $p (< m, n)$ rows, and $\tilde{F}_{(i)}$ be a $(p + 1) \times n$ matrix that contains all rows of \tilde{F} and an additional row \mathbf{f}_i . Condition $rank(\tilde{F}) =$

$\text{rank}(\tilde{\mathbf{F}}_{(i)})$ is satisfied, if and only if \mathbf{f}_i depends on any of rows in $\tilde{\mathbf{F}}$. The algorithm does not require the parameter k to be set in advance. Another advantage of this algorithm is that the total number of operations in the nested loops is bounded, while many k -medoids algorithms do not have bounded complexities because of random search. The bound of loop operations in Phase-1 is $(m-1)+(m-1)+\dots+1 = m(m-1)/2 \sim O(m^2)$.

The medoid selection method (Phase-2) used in the algorithm is based on a distance metric defined as the total distance from a reference feature to all its neighbors, though there can be other criteria for medoid selection. Other than the medoids that have been recognized, all the rest of features are redundant.

3 Feature Relevance Analysis For Practical Datasets

Among many feature relevance analysis techniques are those based on sparse regression which are attractive in terms of computation and traceability [15–17, 22, 25, 26]. In particular, the techniques using joint $l_{2,1}$ -norms minimization [15] are especially interesting to us for its simplicity and efficiency.

3.1 Measuring Feature Relevance Based On $l_{2,1}$ -norm Minimization

The goal is to find a weighting matrix \mathbf{W} in a supervised learning manner. We adopt the framework used in [15] and the problem is formulated as follows.

Let c be the number of classes. Define the weighting matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m)^T \in \mathbb{R}^{m \times c}$ and its extended version $\hat{\mathbf{W}} \in \mathbb{R}^{d \times c}$,

$$\hat{\mathbf{W}} = \begin{pmatrix} \mathbf{W} \\ \dots \\ \hat{w}_{d,1}, \dots, \hat{w}_{d,c} \end{pmatrix}, \quad (1)$$

with $d = m + 1$. The value of $\hat{\mathbf{W}}$ will be determined later. Extend the data matrix \mathbf{F} into $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \in \mathbb{R}^{d \times n}$ by adding an all-one row at the bottom of \mathbf{F} ,

$$\mathbf{X} = \begin{pmatrix} \mathbf{F} \\ \dots \\ 1, \dots, 1 \end{pmatrix}, \quad (2)$$

Assume the dataset comes with n label samples denoted by $a_1, a_2, \dots, a_n \in \{1, 2, \dots, c\}$. Denote the class label matrix by $\mathbf{Y} = (\mathbf{y}_{a_1}, \mathbf{y}_{a_2}, \dots, \mathbf{y}_{a_n}) \in \mathbb{R}^{c \times n}$, where a column vector $\mathbf{y}_j = (0, \dots, 0, 1, 0, \dots, 0)^T$ contains $c - 1$ zero-valued entries and a sole one-valued entry at the j -th position associated with the class j . To find $\hat{\mathbf{W}}$, the regression (minimization) problem is

$$\min_{\hat{\mathbf{W}}} \sum_{i=1}^n \left\| \hat{\mathbf{W}}^T \mathbf{x}_i - \mathbf{y}_i \right\|_2 + \gamma \sum_{j=1}^d \|\hat{w}_j\|_2 \quad (3)$$

where \hat{w}_j is the j -th row of $\hat{\mathbf{W}}$, $\sum_{i=j}^d \|\hat{w}_j\|_2$ is the regularization term, and γ is a constant for tuning the regularization's influence. The problem (3) can be efficiently

solved using the algorithm described in [15] (refer to the reference for the analysis and proof). The first m rows of $\hat{\mathbf{W}}$, i.e., \mathbf{W} , is the outcome we expect. Each of $m \cdot c$ entries of \mathbf{W} reflects how relevant a feature is to a class.

With \mathbf{W} we can also evaluate how important an individual feature is to the overall classification. By adopting the way used in [16], the total relevance of the j -th feature can be calculated by

$$\bar{w}_j = \|\mathbf{w}_j\|_2, j = 1, 2, \dots, m \quad (4)$$

3.2 Class-Occurrence Compensation

The relevance analysis method presented in the last subsection will not work well if no proper compensation for class occurrence is made. Let n_l be the number of instants associated with class l , $l = 1, 2, 3, \dots, c$. Consider an ideal case that $n_l = n/c$, $l = 1, 2, 3, \dots, c$, i.e., equal occurrence for all c classes, we first apply Z-score normalization to the feature dataset and then calculate the weighting matrix \mathbf{W} . In this process all classes are represented equally, which is necessary for a fair analysis. However, equal occurrence does not hold in general, thus certain compensations are needed in order to obtain an unbiased analysis result.

In dataset normalization phase, we need to determine the mean μ_j and standard deviation σ_j for each feature in Z-score normalization: $f_{j,i} \leftarrow (f_{j,i} - \mu_j)/\sigma_j$, $j = 1, 2, \dots, m$, $i = 1, 2, \dots, n$. μ_j and σ_j are given by

$$\mu_j = \frac{1}{c} \sum_{l=1}^c \frac{1}{n_l} \sum_{i=1}^n \mathbf{1}_l(f_{j,i}) f_{j,i}, \quad (5)$$

$$\sigma_j = \frac{1}{c} \sum_{l=1}^c \frac{1}{n_l} \sum_{i=1}^n \mathbf{1}_l(f_{j,i}) (f_{j,i} - \mu_j)^2, \quad (6)$$

$j = 1, 2, \dots, m$

where $\mathbf{1}_l(f_{j,i})$ is an indicator function defined as

$$\mathbf{1}_l(f_{j,i}) = \begin{cases} 1, & \text{if } f_{j,i} \text{ belongs to class } l, \\ 0, & \text{if } f_{j,i} \text{ does not belong to class } l, \end{cases} \quad (7)$$

$j = 1, 2, \dots, m, l = 1, 2, \dots, c, i = 1, 2, \dots, n$

Certain compensation needs to be made in the phase of $l_{2,1}$ -norm minimization as well, and (3) can be extended into the following format:

$$\min_{\hat{\mathbf{W}}} \frac{n}{c} \sum_{l=1}^c \frac{1}{n_l} \sum_{i=1}^n \mathbf{1}_l(\mathbf{x}_i) \left\| \hat{\mathbf{W}}^T \mathbf{x}_i - \mathbf{y}_i \right\|_2 + \gamma \sum_{j=1}^d \|\hat{\mathbf{w}}_j\|_2 \quad (8)$$

To use the algorithm developed in [15], we can convert \mathbf{x}_i and \mathbf{y}_i into $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{y}}_i$, respectively, using the following formulas:

$$\begin{aligned} \tilde{\mathbf{x}}_i &= \frac{n}{cn_l} \mathbf{x}_i, \text{ if } \mathbf{1}_l(\mathbf{x}_i) = 1, \\ \tilde{\mathbf{y}}_i &= \frac{n}{cn_l} \mathbf{y}_i, \text{ if } \mathbf{1}_l(\mathbf{y}_i) = 1, \end{aligned} \quad (9)$$

$l = 1, 2, \dots, c, i = 1, 2, \dots, n$

Table 1. 18 effective features in two categories.

Feature	Description	Network	Payload	Physical
1. command address	Device ID in command packet	✓		
2. response address	Device ID in response packet	✓		
3. response memory	Memory start position in response packet	✓		
4. command memory count	Number of memory bytes for R/W command	✓		
5. response memory count	Number of memory bytes for R/W response	✓		
6. comm write fun	Value of command function code		✓	
7. response write fun	Value of response function code		✓	
8. sub function	Value of sub-function code in the command/response		✓	
9. response length	Total length of response packet	✓		
10. HH	Value of HH setpoint			✓
11. H	Value of H setpoint			✓
12. L	Value of L setpoint			✓
13. LL	Value of LL setpoint			✓
14. control mode	Automatic, manual or shutdown		✓	
15. pump state	Compressor/pump state		✓	✓
16. crc rate	CRC error rate	✓		
17. measurement	Water level		✓	✓
18. time	Time interval between two packets	✓		

By combining (8) and (9), we reach the following optimization which has the same format as (3):

$$\min_{\hat{\mathbf{W}}} \sum_{i=1}^n \left\| \hat{\mathbf{W}}^T \tilde{\mathbf{x}}_i - \tilde{\mathbf{y}}_i \right\|_2 + \gamma \sum_{j=1}^d \|\hat{\mathbf{w}}_j\|_2 \quad (10)$$

4 Case Study Of ICS Intrusion Detection

Precise feature selection can benefit design and evaluation of an Intrusion Detection System (IDS). In this section we use ICS intrusion detection as an example to examine the proposed techniques. Specifically, the water storage tank dataset provided by Morris's group [35] is employed to generate numerical results. The dataset includes class 0 for normal situation and classes 1 to 7 representing seven different types of attacks. Intrusion detection is actually multi-class classification and we use partial decision tree based PART classifier in Weka [36, 37] to perform the job. After removal of a few constant (zero-variance) features, the remaining 18 features are used for analysis. As shown in Table 1, these 18 features belong to three categories, and six of payload features are directly related to physical parameters.

After performing the proposed k -medoids algorithm, $k = 13$ medoids (primary features) are found. As mentioned above, the matrix rank analysis indicates that this

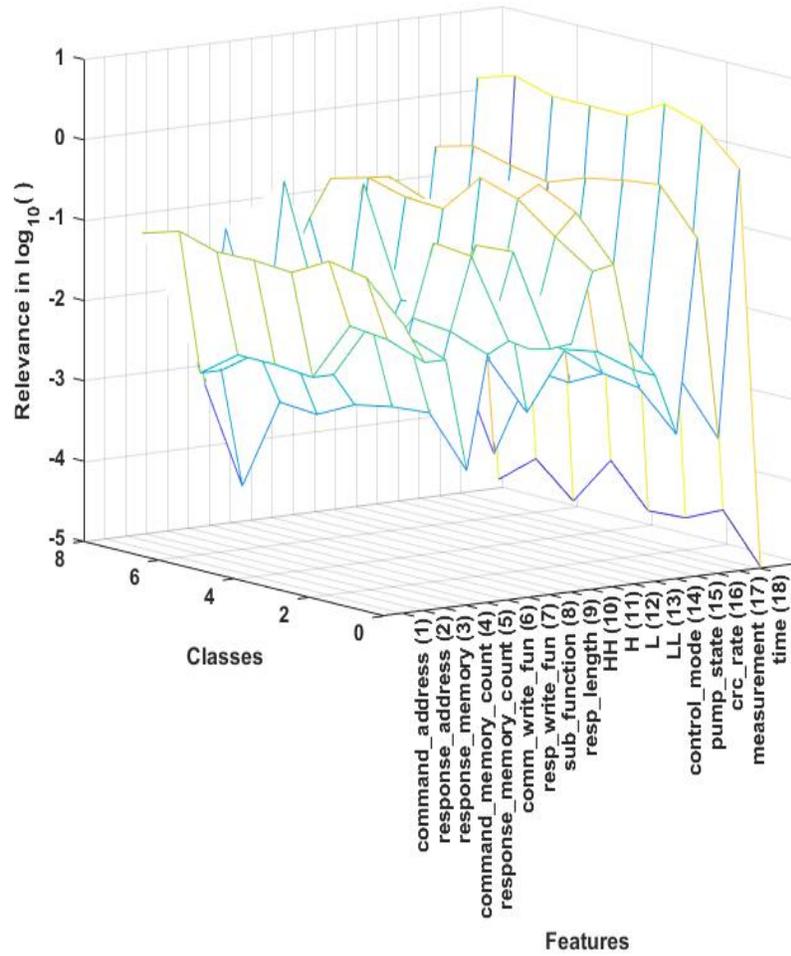


Fig. 2. Individual relevance.

dataset contains 13 effective eigen modes (refer to Fig.1), which implies that, for this particular dataset, each primary feature corresponds to an effective eigen mode, and all the 13 primary features are independent of each other. The algorithm generates 11 singleton clusters {1}, {6}, {8}, {10}, {11}, {12}, {13}, {14}, {15}, {17} and {18} along with two non-singleton clusters {16, 4} and {2, 3, 5, 7, 9} with medoids 4 and 3, respectively. In Fig.2 each data points represents a relevance level of an individual feature with respect to a class, and Fig.3 shows overall impact of each feature on all of the classes, where class-occurrence compensation has been performed prior to relevance calculati

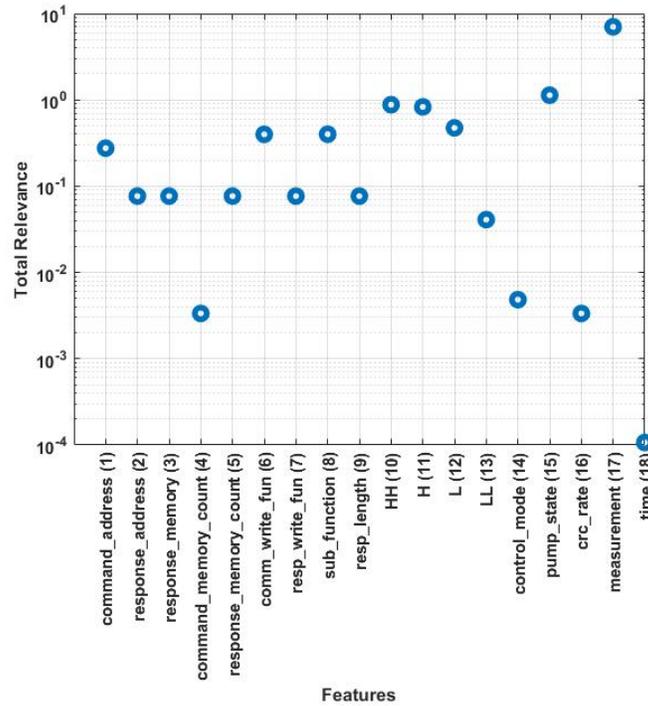


Fig. 3. Total relevance of each feature.

It can be seen in Fig.3 that the features belonging to the same cluster exhibit the same relevance level. In practice it is reasonable not to use redundant features, so we should only measure the relevance levels of the 13 independent features that are fed to the classifier. The total relevance of these selected features is shown in Fig.4.

Table. 2 shows classification performance for using different feature sets. As expected, it is found that removal of redundant features does not degrade classification performance, and it is even beneficial to eliminate some bad (low-relevance-score) features (e.g., features 4, 13, 14, 16 and 18). It can also be verified that removal of independent and important (high-relevance-score) features can degrade the performance. The 8-feature result shown in the Table 2 suggests that feature 17 is critical to the classification of the first 3 classes. These observations validate the correctness of the redundancy and relevance analysis.

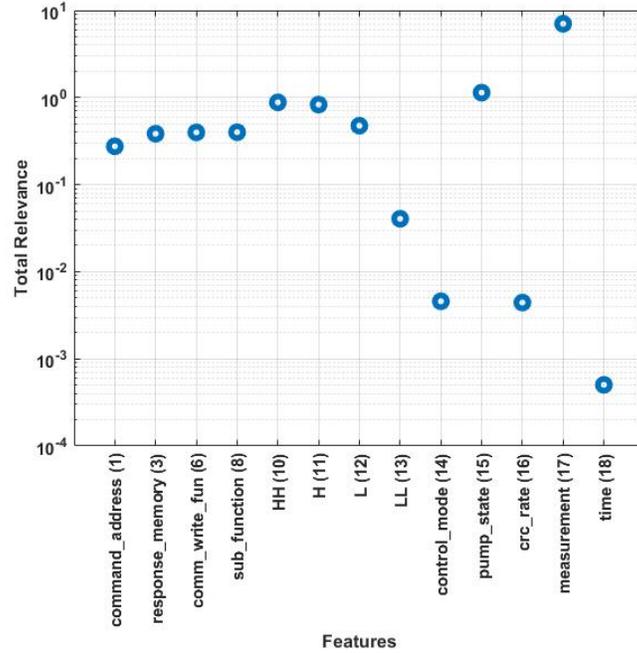


Fig. 4. Total relevance of each feature (only consider 13 independent features).

Table 2. Classification results for (a) all 18 features; (b) 13 features—eliminating 5 redundant features (2, 5, 7, 9, 16); (c) 9 features—eliminating redundant & bad features (2, 4, 5, 7, 9, 13, 14, 16, 18); (d) 8 features—eliminating feature 17 and 9 redundant & bad features.

Class	18 features		13 features		9 features		8 features	
	TP	FP	TP	FP	TP	FP	TP	FP
0	0.988	0.014	0.988	0.014	0.990	0.014	1.000	0.346
1	0.977	0.000	0.977	0.000	0.978	0.000	0.000	0.000
2	0.946	0.009	0.946	0.009	0.946	0.007	0.000	0.000
3	0.971	0.000	0.971	0.000	0.971	0.000	0.971	0.000
4	0.990	0.000	0.990	0.000	0.990	0.000	0.990	0.000
5	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000
6	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000
7	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000
Weighted Average	0.987	0.010	0.987	0.010	0.989	0.010	0.902	0.248

The necessity of class-occurrence compensation can be confirmed experimentally as well. Different from what is shown in Fig.3, a relevance distribution obtained based on the raw dataset without pre-compensation is shown in Fig.5. It can be verified that removal of the “bad” features (e.g., features 15 and 17 are, in fact, very important) suggested by this incomplete analysis can be harmful to the classification task.

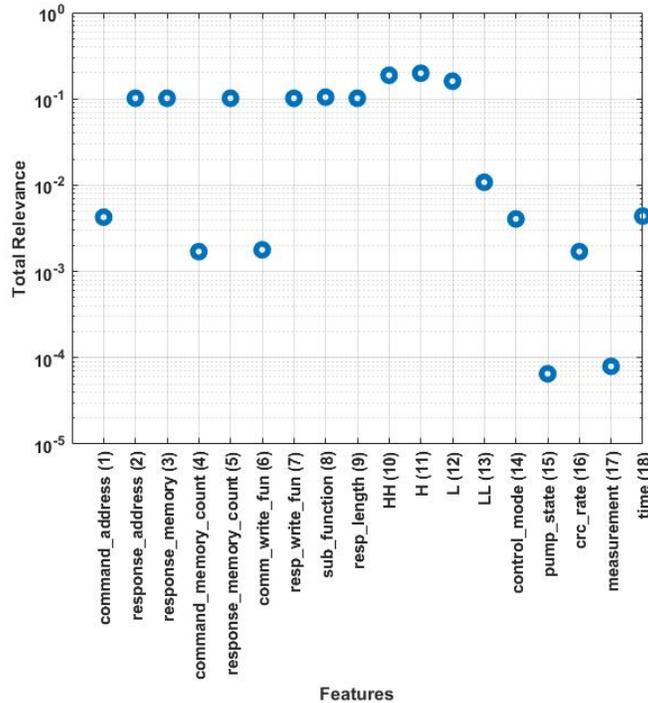


Fig. 5. Total relevance based on raw dataset without pre-compensation.

It has been seen that, without sacrificing detection accuracy, the intrusion detection complexity can be reduced by using only 9 independent and relevant features. In general, we can have a simpler classifier that uses fewer features with some performance penalties. However, the performance penalties on different classes are not equal. For example, classification result in Table 3 is obtained by using only 6 features, and the corresponding performance for detecting attacks 1, 2, 3, 4 and 7 is as good as that when more independent and relevant features are utilized. If the attacks corresponding to classes 5 and 6 were not of our interest, we could have designed a light-weight IDS that would have relied only on the 6 features.

Table 3. Classification results when using 6 features (1, 3, 10, 11 15, 17).

Class	TP	FP
0	0.992	0.039
1	0.978	0.000
2	0.946	0.006
3	0.967	0.000
4	0.990	0.000
5	0.000	0.000
6	0.719	0.000
7	1.000	0.000
Weighted Average	0.983	0.028

5 Conclusions

In this work we have proposed a set of analytical techniques for selecting features efficiently. The matrix rank of feature data is used as a robust indicator for feature clustering. To assess the feature relevance fairly, the inequality of class-occurrence in a practical raw dataset is compensated prior to applying relevance analysis. The compensation idea can be applied to different regression based methods. The effectiveness and necessity of the proposed methods are examined using an existing ICS dataset. One interesting observation from examining the water tank dataset is that some physical features (e.g., features 10, 11,12, 15, 17) can be more important than other types of features. This might be because they are directly related to the physical entities (say, the water level) of interest, suggesting that we could add more sensors to monitor an ICS in order to further improve intrusion detection. Our proposed framework for precise feature selection can help reduce computation of classifiers and guide the design of efficient classification systems, such as an IDS.

Acknowledgment

We express our gratitude towards Cybersecurity Education, Research and Outreach Center (CEROC), as well as Center for Manufacturing Research (CMR), both at Tennessee Tech University, for supporting this research. We would also like to acknowledge Dr. Thomas Morris and his colleagues for providing their datasets.

References

1. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering* **17**(4) (2005) 491–502
2. Xue, B., Zhang, M., Browne, W.N., Yao, X.: A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation* **20**(4) (2016) 606–626

3. Ahmed, S., Zhang, M., Peng, L.: Enhanced feature selection for biomarker discovery in lc-ms data using gp. In: Evolutionary Computation (CEC), 2013 IEEE Congress on, IEEE (2013) 584–591
4. Aghdam, M.H., Ghasem-Aghaei, N., Basiri, M.E.: Text feature selection using ant colony optimization. Expert systems with applications **36**(3) (2009) 6843–6853
5. Ghosh, A., Datta, A., Ghosh, S.: Self-adaptive differential evolution for feature selection in hyperspectral image data. Applied Soft Computing **13**(4) (2013) 1969–1977
6. Ambusaidi, M.A., He, X., Nanda, P., Tan, Z.: Building an intrusion detection system using a filter-based feature selection algorithm. IEEE transactions on computers **65**(10) (2016) 2986–2998
7. Narendra, P.M., Fukunaga, K.: A branch and bound algorithm for feature subset selection. IEEE Transactions on computers **9**(C-26) (1977) 917–922
8. Dash, M., Liu, H.: Feature selection for classification. Intelligent data analysis **1**(3) (1997) 131–156
9. Mitra, P., Murthy, C., Pal, S.K.: Unsupervised feature selection using feature similarity. IEEE transactions on pattern analysis and machine intelligence **24**(3) (2002) 301–312
10. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on pattern analysis and machine intelligence **27**(8) (2005) 1226–1238
11. Shen, K.Q., Ong, C.J., Li, X.P., Wilder-Smith, E.P.: Feature selection via sensitivity analysis of svm probabilistic outputs. Machine Learning **70**(1) (2008) 1–20
12. Covões, T.F., Hruschka, E.R., de Castro, L.N., Santos, Á.M.: A cluster-based feature selection approach. In: International Conference on Hybrid Artificial Intelligence Systems, Springer (2009) 169–176
13. Covões, T.F., Hruschka, E.R.: An experimental study on unsupervised clustering-based feature selection methods. In: Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on, IEEE (2009) 993–1000
14. Jaskowiak, P.A., Campello, R.J., Covoes, T.F., Hruschka, E.R.: A comparative study on the use of correlation coefficients for redundant feature elimination. In: Neural Networks (SBRN), 2010 Eleventh Brazilian Symposium on, IEEE (2010) 13–18
15. Nie, F., Huang, H., Cai, X., Ding, C.H.: Efficient and robust feature selection via joint l_2, l_1 -norms minimization. In: Advances in neural information processing systems. (2010) 1813–1821
16. Xiang, S., Nie, F., Meng, G., Pan, C., Zhang, C.: Discriminative least squares regression for multiclass classification and feature selection. IEEE transactions on neural networks and learning systems **23**(11) (2012) 1738–1754
17. Cai, X., Nie, F., Huang, H.: Exact top-k feature selection via $l_{2,0}$ -norm constraint. In: IJCAI. Volume 13. (2013) 1240–1246
18. Song, Q., Ni, J., Wang, G.: A fast clustering-based feature subset selection algorithm for high-dimensional data. IEEE transactions on knowledge and data engineering **25**(1) (2013) 1–14
19. Zhao, Z., Wang, L., Liu, H., Ye, J.: On similarity preserving feature selection. IEEE Transactions on Knowledge and Data Engineering **25**(3) (2013) 619–632
20. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Computers & Electrical Engineering **40**(1) (2014) 16–28
21. Hou, C., Nie, F., Li, X., Yi, D., Wu, Y.: Joint embedding learning and sparse regression: A framework for unsupervised feature selection. IEEE Transactions on Cybernetics **44**(6) (2014) 793–804
22. Peng, H., Fan, Y.: Direct $l_{2,p}$ -norm learning for feature selection. arXiv preprint arXiv:1504.00430 (2015)

23. Liu, H., Shao, M., Fu, Y.: Consensus guided unsupervised feature selection. In: AAAI. (2016) 1874–1880
24. Ang, J.C., Mirzal, A., Haron, H., Hamed, H.N.A.: Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection. *IEEE/ACM transactions on computational biology and bioinformatics* **13**(5) (2016) 971–989
25. Peng, H., Fan, Y.: A general framework for sparsity regularized feature selection via iteratively reweighted least square minimization. In: AAAI. (2017) 2471–2477
26. Gossmann, A., Cao, S., Brzyski, D., Zhao, L.J., Deng, H.W., Wang, Y.P.: A sparse regression method for group-wise feature selection with false discovery rate control. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2017)
27. Shang, R., Wang, W., Stolkin, R., Jiao, L.: Non-negative spectral learning and sparse regression-based dual-graph regularized feature selection. *IEEE transactions on cybernetics* **48**(2) (2018) 793–806
28. Liu, H., Zhao, Z.: Manipulating data and dimension reduction methods: Feature selection. In: *Encyclopedia of Complexity and Systems Science*. Springer (2009) 5348–5359
29. Liu, H., Motoda, H., Setiono, R., Zhao, Z.: Feature selection: An ever evolving frontier in data mining. In: *Feature Selection in Data Mining*. (2010) 4–13
30. Zhao, Z., Wang, L., Liu, H., et al.: Efficient spectral feature selection with minimum redundancy. In: AAAI. (2010) 673–678
31. Reynolds, A.P., Richards, G., Rayward-Smith, V.J.: The application of k-medoids and PAM to the clustering of rules. In: *International Conference on Intelligent Data Engineering and Automated Learning*, Springer (2004) 173–178
32. Park, H.S., Lee, J.S., Jun, C.H.: A k-means-like algorithm for k-medoids clustering and its performance. *Proceedings of ICCIE* (2006) 102–117
33. Park, H.S., Jun, C.H.: A simple and fast algorithm for k-medoids clustering. *Expert systems with applications* **36**(2) (2009) 3336–3341
34. Jain, A.K., Dubes, R.C.: *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc. (1988)
35. Morris, T., Gao, W.: Industrial control system network traffic data sets to facilitate intrusion detection system research. *Critical infrastructure protection VIII8th IFIP WG 11* (2014) 17–19
36. Holmes, G., Donkin, A., Witten, I.H.: Weka: A machine learning workbench. In: *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*, IEEE (1994) 357–361
37. GNU General Public License: Weka 3: Data mining software in Java

Entropy-based Approach for Parameter-free Attribute Clustering

Adison Khomprasert
Computer Engineering
Kasetsart University
Thailand
g571455011@ku.ac.th

Thanawin Rakthamanon
Computer Engineering
Kasetsart University
Thailand
thanawin.r@ku.ac.th

Kitsana Waiyamai
Computer Engineering
Kasetsart University
Thailand
fengknw@ku.ac.th

Abstract. Data clustering techniques have been well-studied for several decades. Traditional data clustering focuses on finding groups of transactions based on their attribute values, while attribute clustering focuses on grouping attribute values based on their transactions instead. Attribute clustering considers two attribute values similar if they appear in a similar set of transactions. Attribute clustering not only detects groups of similar attribute values, but it can also be used to reduce the number of attribute values needed to represent the whole dataset. When similar attributes have been grouped, the whole dataset will be easier to understand. Then, related attributes and outlier attributes will be discovered. In this paper, we proposed a new method to improve the best-known parameter free attribute clustering algorithm. The results on 23 real-world datasets demonstrate that our proposed method is not only faster but also achieves higher quality clusters compared to the best-known parameter-free attribute clustering method.

Keywords: attribute reduction; data summarization; minimum description length;

1 Introduction

Nowadays, big data are all around us and are important for our daily life. However, because of their complicated and large size, their processing and analysis become extremely difficult. One resolution to overcome these difficult problems is to summarize them into a more compact, but still informative version of the entire data. The main objective of data summarization techniques is to produce a good quality of summaries that require less storage and extremely shorten the time needed to process and analyze. For example, a whole book can be summarized into just title and abstract, which will provide a good first impression of what to expect from it.

Many techniques have been proposed for data summarization such as **Error! Reference source not found.**[4][5]. Attribute clustering [1] is an efficient data summarization technique for transactional or categorical data in which a summary is given in terms of sets of attributes, which are highly correlated to each other. By showing which attributes are highly correlated or interact most strongly, the summary is able to

provide data insight and good first impression of the original entire data. An example of summary forms can be a set of frequent or closed itemsets which are representative of the original datasets [5]. These descriptive frequent patterns can be used as a surrogate to understand and get the insight into the original data.

Many attribute clustering techniques have been proposed to find a good summary of transactional or categorical data [1][2]. The standard approach is to find the frequent patterns first [2]; the result of which quickly grows up too many times of the size of the original dataset. Further, these techniques do not consider interactions between attributes in the data. Several works have been proposed to find the interactions between attributes via the clustering mechanism [1][2]. The output is a set of clusters of attributes that strongly interreact or correlated each other Mampaey and Vreeken [1] proposed the best-known parameter-free algorithm to find the best attribute grouping using Minimum Description Length (MDL). During each merging step, the most two similar attributes will be selected. MDL will be also used to consider when the algorithm would be terminated. This algorithm processes all pairs of attributes, so it may take a long time to find the best grouping and it seems to be too fit to the training data.

The objective of this paper is to improve the performance of the best-known parameter-free attribute clustering proposed in [1]. The attribute selection mechanism of [1] is to select the best pair of attributes to combine by the smallest description length. entropy-based method improves the core selection briefly by finding the most robust candidate attribute, then merge this attribute to one of the best attributes from the rest. As a result, high-quality pairs of attributes have been selected by the entropy-based method and the clustering can achieve higher performance w.r.t classification measurements such as precision, recall, f-measure, area under the ROC curve. The entropy-based method can also lower the new compression data size. The experiments show that our entropy-based attribute clustering algorithm generates a smaller number of clusters, gives better compression ratio, and achieves higher accuracy compared to the best-known attribute clustering. Moreover, our entropy-based method is faster than the best-known algorithm.

The roadmap of this paper is as follows. In Section II, we introduce the best-known attribute clustering algorithm. Then, we introduce our entropy-based attribute clustering algorithm in Section III. In Section IV, we show the experimental results of our proposed algorithm compared to the best-known algorithm. The conclusion and the discussion are in the last section.

2 The Best-known Attribute Clustering

Several methods for attribute clustering have been proposed during the last decades. Although the minimum description length is intractable because of Kolmogorov Complexity **Error! Reference source not found.**, its idea has been widely used to remove parameters from the algorithm. The data description length of a given data can be defined as the length of code (i.e., numbers of bits) needed to describe the data.

In this section, we introduce the basic idea of attribute clustering, including its preliminary definitions. Attribute clustering methods [1][6][5] returns groups of attributes, which lead us to achieve more understanding of the data and can be used to summarize the data.

2.1 Basic Definitions

A dataset D is a set of transactions, $D = \{T_1, T_2, \dots, T_{|D|}\}$. \mathcal{I} be a set of all categorical items in the database, $\mathcal{I} = \{I_1, I_2, \dots, I_n\}$. In a transactional-dataset, a transaction T is a set of items, $T \subseteq \mathcal{I}$. However, in relational-dataset, a transaction is a set of values of attributes. Hence, in this paper, “attribute” will be used interchangeably as “item”. To do clustering, all items in \mathcal{I} can be partitioned into smaller groups, called attribute clusters. The attribute cluster A is a set of items (i.e., attributes), $A = \{I_1, I_2, \dots, I_{|A|}\}$.

Definition 1: An attribute clustering \mathcal{C} is a group of attribute clusters, $\mathcal{C} = \{A_1, A_2, \dots, A_{|\mathcal{C}|}\}$. All items in \mathcal{I} must be included once and only in an attribute cluster. Then, $\mathcal{I} = \bigcup_i A_i \forall$ and $i \neq j: A_i \cap A_j = \emptyset$

The minimum description length (MDL) is used as a measurement to select the best pair of clusters to merge. MDL of a cluster A is the minimum number of bits need to reproduce the cluster A , defined as $L(A)$, and MDL of the whole dataset D is the minimum number of bits need to reproduce all members in clusters and the clustering structure. However, MDL is intractable to calculate as it is hard as Kolmogorov complexity [1]. Then, description length has been used in our paper, instead of MDL.

Definition 2: Description length of an attribute cluster A is calculated as entropy of the attribute cluster.

$$L(A) = - \sum_{I \in A} P(I) \log P(I)$$

where $P(I)$ is the probability that item I appear in the dataset. This entropy has been used to identify the good attribute clusters.

Definition 3: Description length of an attribute clustering \mathcal{C} is the summation of the description length of all clusters in the clustering.

$$L(\mathcal{C}) = \sum_{A \in \mathcal{C}} L(A) + \varepsilon$$

where ε is the number of bits needed to represent elements (i.e., attribute clusters) in the clustering. As we defined in [1], $\varepsilon = \log n + n \log |\mathcal{C}| - \log |\mathcal{C}|!$ where n is the number of items (i.e., attributes) in the dataset.

Definition 4: Description length of the dataset is depended on the attribute clustering. Hence, the description length of the dataset, $L(D, \mathcal{C})$, is defined as:

$$L(D, \mathcal{C}) = L(D|\mathcal{C}) + L(\mathcal{C})$$

Definition 5: The description length of the dataset D encoding by attribute clustering \mathcal{C} , $L(D|\mathcal{C})$, is calculated by the summation of description length of all items in

each transaction. Each item can be encoded as an index of the attribute cluster in the clustering, so it needs only $\log|C|$ bits to represent. Moreover, the summation can be calculated from all attribute clusters, instead of the summation from all transactions. Hence, $L(D|C)$ can be formally defined as:

$$L(D|C) = \sum_{T \in D} \sum_{A \ni T \wedge T \in T} \log|C| = |D| * \sum_{A \in C} freq(A) * \log|C|$$

where $freq(A)$ is the total frequency of any attribute (i.e., item) in an attribute cluster A appeared in the whole dataset.

3 MV Algorithm for Attribute Clustering

In this section, we describe the idea of Mampaey and Vreeken algorithm [1], called MV algorithm for short, as the best-known parameter-free attribute clustering algorithm. This agglomerative clustering algorithm starts by many clusters containing only one attribute. Then, the algorithm will grow each cluster to cover a larger set of attributes. The MDL concept has been applied in [10] to select the best pair of clusters (i.e., groups of attributes) to merge. The key idea of the algorithm is to calculate the description length of the clustering after merging any possible pair. Then, the pair of clusters with the smallest description length will be selected to be merged. This process will continue until the description length of the clustering after merging is larger than the description length of the clustering without merging.

Table 1. MV algorithm

Algorithm: MV Algorithm for attribute clustering
Input: Transactional dataset D , Set of All Attributes
Output: The best minimum attribute clustering C
1. $C \leftarrow$ All Attributes
2. while $ C > 1$ do
3. $min_sim = \infty$
4. $C_{min} = C$
5. for $i = 1$ to $ C $
6. for $j = i+1$ to $ C $
7. $C_{min} \leftarrow C - \{A_i, A_j\} \cup \{A_i \cup A_j\}$
8. if $L(D, C_{min}) < min_sim$
9. $min_sim = L(D, C_{min})$
10. $C_{min} \leftarrow C_{min}$
11. end if
12. end for
13. end for
14. if $L(D, C) \leq min_sim$
15. break ;
16. end if
17. $C \leftarrow C_{min}$
18. end while
19. return C

As shown in **Table 1**, The MV algorithm is the summarizing data by using description length to select the best attribute clustering. Here we show the process. First, the attribute clustering \mathcal{C} will be initialized by setting every single attribute as an attribute cluster (line 1). Then, two clusters will be merged to create a larger cluster until there is only one cluster left (line 2). In this process, all possible pairs of clusters will be temporarily merged (line 7) and the description length of the clustering which contains the new cluster will be calculated (line 8). After trying all possible pairs to merge, the best pair will be selected and merged before starting the next iteration (line 17). If the description length after merging cannot decrease any further, the algorithm will be terminated (line 14-15).

The running time to select the best pair of attribute clusters in each iteration is large as $O(n^2)$ time, where n is the number of attributes. Moreover, selecting the best pair is not only time-consuming but also seems to be too fit to the training data.

4 Entropy-based Attribute Clustering

In this section, we propose a new algorithm for attribute clustering. An entropy-based method can achieve higher performance in term of smaller execution time and higher accuracy compared to the best-known algorithm.

Table 2. Entropy-based attribute clustering algorithm.

Algorithm :Entropy-based Attribute Clustering
Input :A transactional dataset D , Set of All Attributes
Output :The best minimum attribute clustering \mathcal{C}_{\min} .
1. $\mathcal{C} \leftarrow$ All Attributes
2. $\text{max_entropy} = -\infty$
3. while $ \mathcal{C} > 1$ do
4. for $i=1$ to $ \mathcal{C} $
5. if $L(A_i) > \text{max_entropy}$
6. $\text{max_entropy} = L(A_i)$
7. $A_s = A_i$
8. end if
9. end for
10. $\text{min_sim} = \infty$
11. $\mathcal{C}_{\min} = \mathcal{C}$
12. for $j=1$ to $ \mathcal{C} $
13. $\mathcal{C}_{\min} \leftarrow \mathcal{C} - \{A_s, A_j\} \cup \{A_s \cup A_j\}$
14. if $L(D, \mathcal{C}') < \text{min_sim}$
15. $\text{min_sim} = L(D, \mathcal{C}')$
16. $\mathcal{C}_{\min} \leftarrow \mathcal{C}_{\min}$
17. end if
18. end for
19. if $L(D, \mathcal{C}) \leq \text{min_sim}$
20. break;
21. end if
22. $\mathcal{C} \leftarrow \mathcal{C}_{\min}$
23. end while
24. return \mathcal{C}

Entropy-based attribute clustering algorithm, named Entropy-based Attribute Clustering, will be brief as the following steps. First, we select the attribute cluster that has the finest entropy among all attribute clusters, called the selected attribute cluster. Then we use this selected attribute cluster to pair with another attribute cluster to minimize the description length after merge.

Our entropy-based attribute clustering algorithm is shown in **Table 2**, First, as in [1], each attribute will be treated as an attribute cluster in the clustering \mathcal{C} (line 1). Then, the most robust cluster among all clusters in \mathcal{C} will be selected (line 4-9). In this step, the entropy of each attribute cluster will be calculated and the attribute with maximum entropy will be selected (line 5-6).

After the most robust cluster, A_s , has been selected, all other clusters will be temporally paired with the selected cluster (line 13) and the description length after merging these two attribute clusters will be calculated (line 14). The cluster that achieves the smallest description length after pairing to A_s will be selected and merged together in the next iteration (line 15-16). The algorithm will be terminated when the whole clustering contains only one cluster (line 3) or when merging the best pair of clusters cannot decrease the description length any further (line 19-21).

The running time of our entropy-based method to select the best pair reduces from $O(n^2)$ comparisons in the previous algorithm to $O(n)$ comparisons. Moreover, because the cluster selection has been modified our proposed algorithm seems to select the more robust attributes to merge.

5 Experimental Results

In this section, we demonstrate the results on 23 real-world datasets in term of execution time and clustering quality. Two algorithms which are the best-known attribute clustering algorithm [1], called MV algorithm, and our proposed algorithm, called Entropy-based Attribute Clustering, have been compared.

5.1 Experimental Setup.

The algorithms have been written in C++ on Ubuntu. The experiments are measured on 2.6 GHz Intel Core i5 with 8GB of memory. **Table 3** shows the characteristics of all 23 datasets used in our experiments. The performance will be given in terms of compression rate and accuracy.

Table 3. Characteristics of datasets

Dataset	# Attributes	# Rows	# Class
adult	97	48842	2
anneal	71	989	5
auto	135	205	6
breast	17	699	2
car	25	1728	4
connect4	129	67557	3
cylBands	122	540	2
ecoli	34	336	8
flare	38	1389	8
glass	46	214	6
heart	50	303	6
hepatitis	52	155	2
horseColic	83	368	2
ionosphere	157	351	2
iris	19	150	3
mushroom	90	8124	2
nursery	32	12960	5
pageBlocks	44	5473	5
pima	38	768	2
soybean-large	118	683	19
tic-tac-toe	29	958	2
wine	68	178	3
zoo	42	101	7

5.2 Quality in terms of Compression Rate

In **Table 4**, the preliminary results have been proposed in [17]. Hence, both algorithms, MV algorithm and our Entropy-based Attribute Clustering, have been compared in term of execution time and compression ratio (CPR). CPR represents the size of original data (i.e., numbers of attributes) before summarization compared to the size of the summarized data (i.e., number of clusters). The higher CPR, the better compression algorithm.

Table 4. Quality in terms of compression rate

Dataset	MV algorithm			Entropy-based attribute clustering algorithm		
	C	Time (s)	CPR	C	Time (s)	CPR
adult	9	32.65	10.78	5	12.38	19.40
anneal	12	0.20	5.92	6	0.08	11.83
auto	22	0.20	6.14	19	0.12	7.11
breast	3	0.01	5.67	2	0.01	8.50
car	5	0.04	5.00	5	0.02	5.00
connect4	7	84.32	18.43	7	44.76	18.43
cylBands	22	0.35	5.55	25	0.16	4.88
ecoli	6	0.02	5.67	3	0.02	11.33
flare	6	0.07	6.33	5	0.04	7.60
glass	10	0.03	4.60	10	0.02	4.60
heart	14	0.04	3.57	8	0.03	6.25
hepatitis	14	0.03	3.71	13	0.02	4.00
horseColic	21	0.11	3.95	19	0.07	4.37
ionosphere	30	0.37	5.23	33	0.23	4.75
iris	4	0.01	4.75	3	0.01	6.33
mushroom	5	5.16	18.00	3	1.94	30.00
nursery	6	0.63	5.33	7	0.44	4.57
pageBlocks	7	0.27	6.29	6	0.18	7.33
pima	8	0.03	4.75	7	0.02	5.43
soybean	9	0.43	13.11	11	0.21	10.73
tic-tac-toe	9	0.03	3.22	9	0.03	3.22
wine	12	0.05	5.67	13	0.04	5.23
zoo	9	0.02	4.67	8	0.02	5.25

From **Table 4**, our proposed algorithm can achieve smaller CPRs on 14 out of 23 datasets. The experiments demonstrate that the proposed attribute process can select better pairs of attributes to merge. Moreover, because in each selection process, the number of comparison reduce from $O(n^2)$ to $O(n)$, our proposed methods can run twice faster than MV algorithm.

5.3 Quality in terms of accuracy

In this section, we compare the performance between our Entropy-based attributing clustering algorithm and MV algorithm in term of f-Measure, precision, recall and ROC. Different from traditional data clustering, attribute clustering focus on grouping attributes instead of grouping transactions. Hence, decision tree classification has been used to test the model accuracy. The results show that entropy-based attribute clustering algorithm obtains higher accuracy on 10 datasets, comparable accuracy for 13 datasets, and with no loss. Entropy-based attribute clustering algorithm has been resolving the process of selection for grouping the attribute clusters. This result of accuracy shows that the entropy-based attribute clustering algorithm can achieve the better results of attribute clustering as shown in **Table 5**.

Table 5. Quality in terms of Accuracy

Dataset	MV algorithm				Entropy-based attribute clustering algorithm			
	F-Measure	Precision	Recall	ROC	F-Measure	Precision	Recall	ROC
adult	0.66	0.58	0.76	0.50	0.66	0.58	0.76	0.50
anneal	0.73	0.69	0.79	0.70	0.88	0.86	0.90	0.84
auto	0.49	0.49	0.50	0.75	0.53	0.54	0.55	0.80
breast	0.52	0.43	0.66	0.50	0.52	0.43	0.66	0.50
car	0.58	0.50	0.70	0.50	0.58	0.50	0.70	0.50
connect4	0.52	0.43	0.66	0.50	0.52	0.43	0.66	0.50
cylBands	0.60	0.76	0.67	0.61	0.60	0.76	0.67	0.61
ecoli	0.25	0.18	0.43	0.47	0.25	0.18	0.43	0.47
flare	0.77	0.71	0.84	0.50	0.77	0.71	0.84	0.50
glass	0.19	0.13	0.36	0.47	0.20	0.16	0.36	0.49
heart	0.38	0.29	0.54	0.47	0.38	0.29	0.54	0.47
hepatitis	0.70	0.63	0.79	0.47	0.70	0.63	0.79	0.60
horseColic	0.48	0.43	0.59	0.44	0.48	0.42	0.60	0.45
ionosphere	0.82	0.83	0.82	0.82	0.82	0.83	0.82	0.75
iris	0.52	0.46	0.63	0.75	0.56	0.50	0.67	0.83
mushroom	0.71	0.82	0.73	0.72	0.93	0.94	0.93	0.93
nursery	0.32	0.36	0.40	0.56	0.48	0.55	0.53	0.68
pageBlocks	0.85	0.81	0.90	0.50	0.85	0.81	0.90	0.50
pima	0.51	0.42	0.65	0.50	0.71	0.75	0.74	0.63
soybean	0.24	0.20	0.34	0.77	0.31	0.32	0.38	0.81
tic-tac-toe	0.52	0.43	0.65	0.50	0.52	0.43	0.65	0.50
wine	0.50	0.45	0.59	0.69	0.69	0.69	0.69	0.76
zoo	0.23	0.17	0.41	0.44	0.56	0.54	0.64	0.73
Win	10 datasets							
Equal	13 datasets							
Lose	0 datasets							

6 Conclusions

In this paper, a new parameter-free algorithm for attribute clustering has been proposed. Similar to the best-known attribute clustering algorithm proposed in [1], our entropy-based attribute clustering algorithm is an agglomerative clustering, and the MDL concept has been applied to make the algorithm parameter-free. In contrast, our proposed algorithm has improved the process of selecting the best pairs of attribute clusters to merge. This improvement not only reduces the running time but also increases the quality of the clustering.

Our intensive experiments on 23 real-world datasets demonstrate that our proposed algorithm is better than the best-known algorithm in term of accuracy, evaluated by precision, recall, f-measure, and ROC. Our proposed algorithm is about twice faster than the best-known algorithm. Higher compression ratio shows that entropy-based

attribute clustering algorithm will return a smaller number of clusters while obtaining the higher accuracy, compared to the best-known algorithm.

References

- [1] Michael Mampaey and Jilles Vreeken. Summarising Data by Clustering Items. In Proc. of ECML PKDD'10 , pp. 130–173, 2010.
- [2] Wang J, Karypis G SUMMARY: efficiently summarizing transactions for clustering. In: Proc. of the IEEE international conference on data mining (ICDM'04), IEEE, pp 241–248.
- [3] Siebes, J. Vreeken, and M. van Leeuwen. Item sets that compress. In Proc. of SDM'06, pages 393–404, 2006.
- [4] JB. Bringmann and A. Zimmermann. The chosen few: On identifying valuable patterns. In mining (KDD'06). ACM, New York, pp 730–735.
- [5] X. Yan, H. Cheng, J. Han, and D. Xin. Summarizing itemset patterns: A profile-based approach. In Proc. of KDD'05, pp 314–323, 2005.
- [6] T. Calders and B. Goethals. Mining all non-derivable frequent itemsets. In Proc. of ECML PKDD'02, pp. 74–85, 2002.
- [7] T. M. Cover and J. A. Thomas. Elements of Information Theory, 2nd ed. John Wiley and Sons, 2006.
- [8] G. C. Garriga, E. Juntila, and H. Mannila. Banded structure in binary matrices. In Proc. of KDD'08, pages 292–300, 2008.
- [9] Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas. Assessing data mining results via swap randomization. TKDD, 1(3), 2007.
- [10] P. D. Grünwald. The Minimum Description Length Principle. MIT Press, 2007.
- [11] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: Current status and future directions. Data Mining and Knowledge Discovery, 15(1), 55–86, 2007.
- [12] H. Heikinheimo, E. Hinkkanen, H. Mannila, T. Mielikäinen, and J. K. Seppänen. Finding low-entropy sets and trees from binary data. In Proc. of KDD'07, pp 350–359, 2007.
- [13] H. Heikinheimo, J. Vreeken, A. Siebes, and H. Mannila. Low-entropy set selection. In Proc. of SDM'09, pp 569–579, 2009.
- [14] J. Knobbe and E. K. Y. Ho. Maximally informative k-itemsets and their efficient discovery. In Proc. of KDD'06, pp 237–244, 2006.
- [15] J. MacQueen. Some methods for classification and analysis of multivariate observations. In Proc. of the 5th Symposium on Mathematical Statistics and Probability, 1967.
- [16] J. Wang and G. Karypis. SUMMARY: Efficiently summarizing transactions for clustering. In Proc. of ICDM'04, pp 241–248, 2004.
- [17] Adison Khomprasert, Thanawin Rakthamanon and Kitsana Waiyamai . Entropy-based Attribute Clustering. In: Proc. of the ECTI & DAMT NCON 2019 international conference on Application of AI, Thailand, pp 230–233.

Seq2SQL - Evaluating Different Deep Learning Architectures Using Word Embeddings

Kevin Stowers¹ and Dirk Krechel¹

RheinMain University of Applied Sciences, Wiesbaden, Germany

Abstract. Having access to relational databases, which carry an enormous amount of today's knowledge, requires an understanding and application of SQL¹. Learning and utilizing this language can be difficult in addition to requiring a significant amount of time. In order to make this information accessible without having to learn SQL this paper proposes and evaluates different deep learning architectures, which translate questions into SQL statements, using word embedding. These neural networks are trained via supervised learning.

In order to train and evaluate the architectures the dataset WikiSQL[20]² is used. It contains 80654 examples and matching SQL queries as well as an SQLite database with 24241 tables.

Throughout this paper three different base models with different parameters are tested and evaluated regarding their efficiency and accuracy. The result will show which base models and parameters show the most promising results in terms of efficiency and accuracy.

Also this paper introduces Token Selector Encoding as a way of encoding queries using word embeddings, which is closely related to the way [20] encodes and shows promising results.

Keywords: Machine Learning · Deep Learning · Natural Language Processing

1 Introduction

A large amount of today's knowledge is stored in relational databases around the globe. In order to retrieve the stored information it is necessary to use a query language such as SQL. Understanding and using these query languages can be very difficult and takes a lot of time. In order to make this knowledge available without having to learn a query language researchers all over the world try to automate the information retrieving process using different machine learning techniques. During this process deep learning has shown promising results. So trying to find an efficient deep neural network model, which translates natural language into SQL statements, has become a popular topic for natural language researchers as well as machine learning researchers.

¹ Structured Query Language

² <https://github.com/salesforce/WikiSQL>

In order to contribute to the process of finding a promising architecture different deep learning networks are tested and evaluated throughout this paper. These architectures range between simple multilayer perceptrons[4] and complex LSTM networks[16]³.

The data set [20], which is used, contains 80654 different questions, the related SQL statements and an SQLite database. Despite being released 2017 the data set is viewed as the standard data set for generating SQL queries from natural language, because of its size and diversity. Table 1 compares all papers, which have been published using the data set. [22] has the highest accuracy of these models with 98.6 %. It combines different models ([22] and [21]), which have had competitive results.

Model	Training accuracy	Test accuracy
SQLova + Execution-Guided Decoding [22]	90.2 %	89.6 %
IncSQL + Execution-Guided Decoding [17]	87.2 %	87.1 %
Execution-Guided Decoding [21]	84.0 %	83.8 %
SQLova [22]	87.2 %	86.2 %
IncSQL [17]	84.0 %	83.7 %
MQAN (unordered) [13]	82.0 %	81.4 %
MQAN (ordered) [13]	82.0 %	81.4 %
Coarse2Fine[5]	79.0 %	78.5 %
Execution-Guided Decoding[1]	78.5 %	78.3 %
TypeSQL[25]	74.5 %	73.5 %
PT-MAML[9]	68.3 %	68.0 %
SQLNet[23]	69.8 %	68.0 %
[2]	67.1 %	66.8 %
Seq2SQL[20]	60.8 %	59.4 %
Baseline[20]	37.0 %	35.9 %

Table 1. Tested settings

The process of understanding words and its surroundings is difficult. The shortness of questions makes this task even harder. During the process of parsing questions information can be lost or the word can be misinterpreted by parsing one word isolated to the neighboring words. In order to avoid this loss of information this paper proposes using word embeddings. By this approach there are two major advantages. Firstly, the words are set into a context. This idea derives from the distributional hypothesis [27]. This is a major advantage of using word embeddings. Basic words can have a lot of different meanings in different contexts, therefore considering the surroundings of a word is appropriate. Secondly, words, which are important for the query language, are mapped into a similar area. Which means that questions have a similar vector due to the common structure of a question. By training the neural network learns this structure and

³ long short-term memory networks

understands signal words and their meaning. In addition to that [18] proves that using word2vec [19], which is utilized in this paper, leads to improved accuracy at much lower computational costs.

So the process of transforming natural language questions into SQL statements is divided into two steps. Firstly the words of the question are mapped to a vector. Therefore the right vector is extracted from the vector space generated by word2vec. Secondly the sentence is given to the deep neural network, which transforms these vectors into an encoded SQL statement. This process is shown in figure 1 and further elaborated in Section 3.

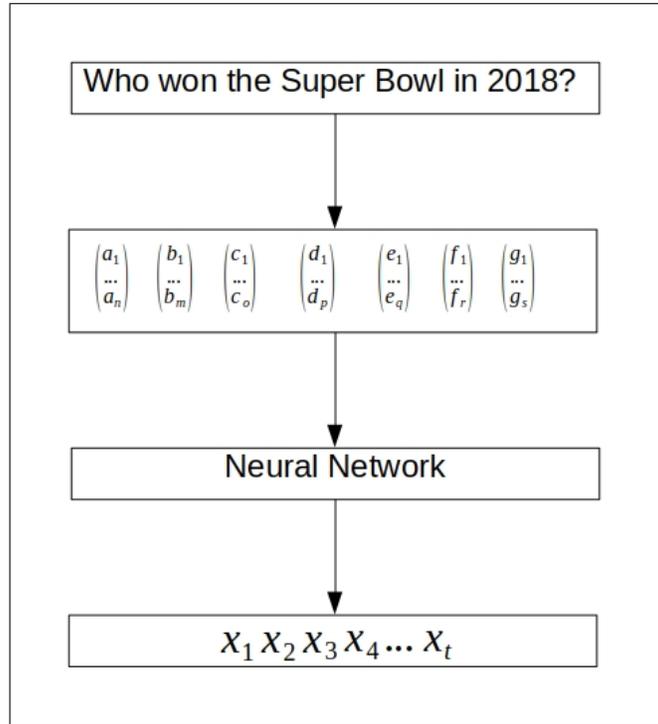


Fig. 1. Process overview example

2 Referenced Work

The data set WikiSQL was released in 2017 by [20]. The data set contains 80654 hand-annotated natural language questions, the corresponding SQL queries and 24241 tables from Wikipedia⁴. To this date WikiSQL is the largest

⁴ <https://www.wikipedia.org>

data set and viewed as the standard data set for the task of creating SQL queries from natural language. The tables as well as the questions exist in raw JSON format. The data set also comes with an SQLite database, which can be used to validate the queries. Additionally [20] proposes a sophisticated way of encoding and decoding SQL queries, which is closely related to the Token Selector Encoding (Section 3), which is introduced by this paper.

The process of creating vectors from words can be difficult. Valuable information can be lost during the process of transforming words to vectors. In order to keep the loss of information as little as possible this paper proposes using word embeddings. Therefore word2vec, which was developed by researchers at Google⁵, is used. [19] introduces word2vec as an autoencoder [15], which transforms words to vectors according to their surrounding by using a skip-gram model. Skip-gram models are robust and often produce state-of-the-art results [8]. Even with sparse sentences this creates word vectors, which contain a lot of information.

By combining both approaches an architecture is created, which creates SQL statements from natural language through word embeddings.

3 Token Selector Encoding

This paper proposes a new way of query encoding, which is closely related to the way [20] encodes queries. Below an JSON-object of the data set is shown.

```
{
  "phase": 2,
  "table-id": "2-1226335-1",
  "question": "What year did Elf Team Tyrrell have 39 points and a Tyrrell 007 Chassis",
  "sql": {
    "sel": 0,
    "conds": [[1, 0, "elf team tyrrell"], [4, 0, "39"], [2, 0, "tyrrell 007"]],
    "agg": 4
  }
}
```

Since the goal is to transform a question to the fitting SQL statement, the JSON-object's important keys are *question* and *sql*. The process of data engineering starts with dividing a question's words into word tokens. Afterwards each token is fitted to the corresponding word embedding v . These word embeddings have the shape $m \times 1$. m is the predefined dimensionality of the vector space created by word2vec. All question's word embeddings are concatenated and a matrix W of the size $|v| \times n$, which represents the question, is created. Where n is the maximal amount of words a question can have⁶. This matrix is fed into the neural network. Representations of v and W are shown in Equation 1 and 2.

⁵ <https://www.google.com>

⁶ in the dataset [20] n is 44

$$v = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \quad \forall x_i \in \mathbb{R}, i, m \in \mathbb{N} \quad (1)$$

$$W = [v_1; v_2; \dots; v_m] = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ x_{m,1} & \dots & \dots & x_{m,n} \end{bmatrix} \quad \forall x_{i,j} \in \mathbb{R}, i, j, m, n \in \mathbb{N} \quad (2)$$

The target vector, which represents the JSON-object's *sql*, is a fourteen-dimensional vector *sel*, which is shown in Equation 3.

$$sel = \begin{bmatrix} a \\ s_1 \\ c_1 \\ s_2 \\ cq_1 \\ \vdots \\ c_4 \\ s_5 \\ cq_4 \end{bmatrix} \quad \forall a, s_i, c_j, cq_k, i, j, k \in \mathbb{N} \quad (3)$$

$$A := \{, MAX, MIN, COUNT, SUM, AVG\} \quad (4)$$

$$S := \{x \in \mathbb{N} | 0 < x \leq n\} \quad n \in \mathbb{N} \quad n \text{ is the amount of columns} \quad (5)$$

$$C := \{=, >, <, OP\} \quad (6)$$

$$Cq := \{x \in \mathbb{N} | 0 < x \leq w\} \quad w \in \mathbb{N} \quad w: \text{ amount of word tokens in question} \quad (7)$$

The target vector's values are connected to sets, which represent different features. *a* corresponds to the set *A* (Equation 4), *s* to *S* (Equation 5), *c* to *C* (Equation 6) and *cq* to *Cq* (Equation 7). So *a* and *c* represent the index of an element from *A* and *C*. Every *s* describes the index of a column of the table, which is searched and every *cq* represents the index of the word token of the question. So *cq* creates a connection between matrix *W* and vector *sel*. Since the question's word tokens are used in order to find the right query, it is important to find an algorithm, which parses the word tokens correctly. The proposed algorithm parses all entries of the database and matches them with the questions. Therefore the tokenizer recognizes if two or more words of a question are one database entry and puts them into one token.

The process of en- and decoding is shown in figure 3. This approach is suitable for word embeddings and sparse information sources. Since questions are sparse and have a more or less common structure, the neural network recognizes the sentence structure, which helps finding the correct query.

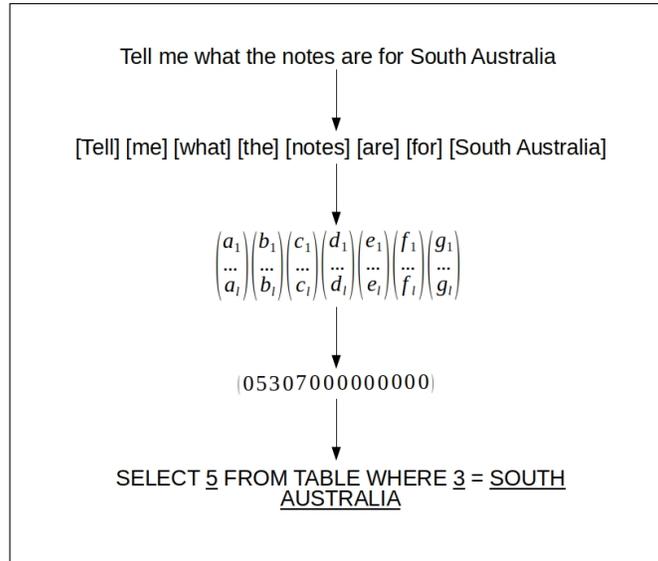


Fig. 2. Encoding process. Step one: tokenization. Step two: Tokens to vectors. Step three: Calculation of query vector. Step four: Query vector to query

4 Comparing the different models

This part of the paper is going to elaborate on the comparison of different architectures. Firstly, the different base models are going to be described. The models will be referred to as base models since they are going to serve as templates. These templates will get different parameters on which they are going to get evaluated. Secondly, the different investigated evaluation parameters and their usefulness are going to be explained.

4.1 Investigated Base Models

The different models were chosen based on complexity and simplicity. In order to combine both premises a multilayer perceptron[4], a convolutional neural network[24] and an LSTM[16] were chosen.

These three base models illustrate a rising complexity. The least complex base model is the multilayer perceptron⁷. A convolutional neural network⁸ is more complex. Apart from these two an LSTM is even more complex. Even though the base models show a rising complexity in themselves, they are basic and therefore modifiable in terms of their depth and used optimization algorithms.

In order to keep the analysis as general as possible two optimization algorithms and two depths were chosen. They were picked based on pre-selection. During

⁷ in the following referred to as MLP

⁸ in the following referred to as cnet

the process of pre-selection one thousand samples of the training set and three hundred samples of the testing set were taken into consideration and evaluated on.

The tested optimization algorithms included a stochastic gradient descent optimizer[10], RMSprop optimizer [7], Adagrad optimizer[11], Adadelata optimizer[26], Adam optimizer[12], Adamax[12] and Nesterov Adam optimizer[6]. The algorithms, which showed the most promising learning results, are Adadelata and RMSProp. But since Adadelata and RMSProp are very similar Adagrad is chosen instead of Adadelata. Another adjustable parameter, which comes hand in hand with the optimization algorithms, is the learning rate.

In order to preselect the depths of the models, depths of three, five, seven, nine and eleven were tested. In every model one of the layers is always a Dropout layer[14] in order to prevent the neural networks from overfitting. The base models with three and five hidden layers showed the most interesting results. Therefore they will be taken into consideration.

The fourth parameter, which is taken into account, is the loss function. In order to find the most interesting loss functions a preselection of five different loss functions was investigated. Those evaluated loss functions are mean squared error⁹, mean absolute error, mean absolute percentage error, mean squared logarithmic error, mean square logarithmic error¹⁰ and logcosh. The results indicated that MSE and MSLE showed the most promising and interesting results. Therefore these two loss functions are taken into consideration.

4.2 Conclusion Of Parametric Choices

Since three base models with seven different settings will be tested and evaluated 24 different settings are possible. In order to be able to find a conclusion and evaluate the different base models, the settings are reduced to twelve. The evaluated settings are shown in table 2. The evaluation will refer to the settings by the numbers in table 2.

4.3 Evaluation Parameters

In order to judge the performance of the different settings, several metrics are taken into account, such as training and evaluation accuracy, training and evaluation time and efficiency.

Efficiency is calculated through Equation 8. This linear Equation describes a subjective perception of the importance of evaluation accuracy (etime) and training time (ttime). The greater r the more important the evaluation accuracy, the lower r the more important training time. In the following r is considered to be

⁹ in the following referred to as MSE

¹⁰ in the following referred to as MSLE

Setting	Base model	Optimizer	Learning rate	Hidden layers	Loss function
1	MLP	Adagrad	0.01	three	MSE
2	cnet	Adagrad	0.01	three	MSE
3	LSTM	Adagrad	0.01	three	MSE
4	MLP	Adagrad	0.005	five	MSLE
5	cnet	Adagrad	0.005	five	MSLE
6	LSTM	Adagrad	0.005	five	MSLE
7	MLP	RMSProp	0.01	three	MSLE
8	cnet	RMSProp	0.01	three	MSLE
9	LSTM	RMSProp	0.01	three	MSLE
10	MLP	RMSProp	0.005	five	MSE
11	cnet	RMSProp	0.005	five	MSE
12	LSTM	RMSProp	0.005	five	MSE

Table 2. Evaluated settings

Setting	Training Acc	Evaluation Acc	Training time	Evaluation time	Efficiency
1	69.25 %	66.44 %	11.51	12.94	320.69
2	71.53 %	69.53 %	1	1	346.65
3	91.50 %	77.94 %	53.17	99.90	336.53
4	69.12 %	67.00 %	11.78	13.15	323.22
5	70.71 %	68.81 %	1.42	1.37	342.63
6	86.80 %	77.91 %	52.73	107.81	336.82
7	7.34 %	20.22 %	13.10	13.03	88
8	1.94 %	1.69 %	1.05	1.08	7.4
9	3.20 %	0.15 %	52.50	132.53	-51.75
10	9.03 %	6.81 %	13.68	13.44	20.37
11	64.60 %	64.58 %	1.30	1.36	321.6
12	80.40 %	69.50 %	30.09	80.50	317.41

Table 3. Evaluation results

five.

$$\begin{aligned}
 X &:= \{\text{all evaluated nets}\} \\
 \text{eff}(x) &= \text{eacc}(x) * r - \text{ttime}(x) \quad x \in X, \text{eff}(x), r, \text{eacc}(x), \text{traintime}(x) \in \mathbb{R}
 \end{aligned} \tag{8}$$

The times, which are evaluated, are relative times. They are brought into relations and therefore in relative values. This is shown in Equations 9 to 11. The advantage is that the times are not dependent on the hardware. The only parameter relative times depend on are calculation steps.

$$T_{xtime} := \{xtime_{abs}(x)\} \quad x \in X, xtime_{abs} \in \{etime_{abs}, ttime_{abs}\} \tag{9}$$

$$\min_{xtime} = \min(T_{xtime}) \tag{10}$$

$$xtime = \frac{xtime_{abs}(x)}{\min_{xtime}} \tag{11}$$

The evaluation result is shown in table 3 and figure 3.

5 Evaluation

The evaluation focuses on finding a relation between all the results. Therefore all five Parameters are evaluated and examined regarding the settings. During evaluation the dimensionality of the word embeddings is 50. Since the training and testing data set contains 61022 different words, this was considered to be a reasonable choice.

The graph in figure 3 illustrates table 3. The graph gives enhances the un-

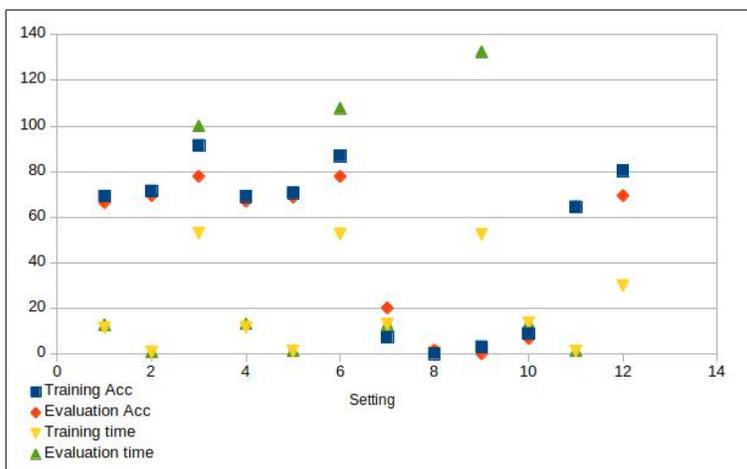


Fig. 3. Illustrated results

derstanding of the parameter combinations. The table indicates that cnets and LSTMs are the preferred base models for the task of transforming natural language to SQL queries. Since a sentence is a sequential information base models, which operate on sequential information are naturally more advanced.

Looking at the evaluation accuracy and the optimization algorithms shows that the accuracy decreases with the use of RMSProp and a high initial learning rate. Both investigated optimization functions, Adagrad and RMSProp, are recommended to be used with data sets larger than 10,000 single cases [7]. Yet both show very different results. Adagrad seems to be working well with MLPs, cnets and LSTMs. Independent from the loss function and learning rate it shows robust results throughout the six different models. On the other hand, RMSProp shows diverse and inconsistent results and seems to be depending on the loss function, the learning rate as well as the base model. For further investigation it is reviewed whether these unstable results are a result of the learning rate or the loss function. Therefore four different settings are tested and evaluated. These settings differ in learning rate and depth. Table 5 illustrates the result.

Comparing setting 8 and setting 1a shows that lowering the learning rate from

Setting	Model	Optimizer	Learning rate	Hidden layers	Loss	Accuracy
1a	cnet	RMSProp	0.001	three	MSLE	70.80 %
2a	cnet	RMSProp	0.001	five	MSLE	69.99 %
3a	cnet	RMSProp	0.005	three	MSLE	65.89 %
4a	cnet	RMSProp	0.005	five	MSLE	60.08 %

Table 4. Further investigation on RMS prop and its unstable results

0.01 to 0.001 increases the accuracy by 68.30 %. This proves that RMSProp needs to have a low initial learning rate, since lowering the learning rate leads to better accuracy. This means that the initial results in table 3 are impacted by the high learning rate. High learning rates can lead to overshooting and therefore unlearning the learnt [3]. [7] even suggests using an initial learning rate of 0.001 for RMSProp.

Additionally table 3 shows that depth does barely impact the models, which use Adagrad as optimization function ¹¹. Results vary in its decimal places. E. g. setting 1 has an evaluation accuracy of 66.44 % and setting 4 an accuracy of 67.00 %. For further analysis eight neural networks ranging from two to nine hidden layers are examined analyzed. The results of this analysis are displayed in table 5.

The results show that the accuracy does improve significantly by using four hidden layers. Using more than four layers does not improve the accuracy notably. Using nine or more hidden layers even decreases the accuracy.

Overall table 3 shows that the accuracy is impacted by the base models, which are used. Depth, learning rate and optimizer have a minor influence on the accu-

¹¹ setting one to six

Setting	Model	Optimizer	Learning rate	Hidden layers	Loss	Accuracy
1b	cnet	Adagrad	0.01	two	MSE	68.41 %
2b	cnet	Adagrad	0.01	three	MSE	70.86 %
3b	cnet	Adagrad	0.01	four	MSE	72.27 %
4b	cnet	Adagrad	0.01	five	MSE	72.40 %
5b	cnet	Adagrad	0.01	six	MSE	72.71 %
6b	cnet	Adagrad	0.01	seven	MSE	72.34 %
7b	cnet	Adagrad	0.01	eight	MSE	72.68 %
8b	cnet	Adagrad	0.01	nine	MSE	71.92 %

Table 5. Further investigation on depth not being an impact

racy. On the one hand LSTMs show the most promising accuracy while dealing with long training and evaluation times, because of the complexity of the calculations. On the other hand cnets and MLPs have a similar accuracy and low evaluation time. This is due to their simplicity.

6 Conclusion

This paper has two major contributions in finding a neural network, which helps encoding natural language to SQL queries. On the one hand it presents Token Selector Encoding or TSE - a new, promising and simple looking way of encoding queries using word embeddings. TSE works by taking the structure of a question into account. The results in table 3 show that it is a sophisticated yet simple way of encoding queries. On the other hand it compares different standard neural networks and shows their value to the task of finding the optimal neural network for this task.

Looking forward we would like to continue our work in the direction of finding an architecture, which transforms natural language questions to SQL queries. Therefore we would like to combine our contributions regarding neural networks with the successful and accurate models [22], [17] and [21]. Since the data set WikiSQL is not as diverse as questions by humans are, we would like to test the models with real world data. Overall there are a lot of contributions yet to be made in order to find a solution, which can be applied in real world scenarios.

References

1. Chenglong Wang, Po-Sen Huang, A.P.M.B.R.S.: Execution-guided neural program decoding (2018)
2. Chenglong Wang, Marc Brockschmidt, R.S.: Pointing out sql queries from text (2017)
3. D. Randall Wilson, T.R.M.: The need for small learning rates on large problems. In Proceedings of the 2001 International Joint Conference on Neural Networks (2001)

4. David E. Rosenblatt, Geoffrey E. Hinton, R.J.W.: Learning internal representations by error propagation. *Parallel distributed processing: Explorations in the microstructure of cognition* (1986)
5. Dong, L., Lapata, M.: Coarse-to-fine decoding for neural semantic parsing. *CoRR* (2018)
6. Dozat, T.: Incorporating nesterov momentum into adam (2016)
7. Geoffrey Hinton, Nitish Srivastava, K.S.: A sepperate, adaptive learning rate for each connection. *Lecture 6d* (2012)
8. Goldberg, Y.: A primer on neural network models for natural language processing (2015)
9. Huang, P., Wang, C., Singh, R., and Xiaodong He, W.Y.: Natural language to structured query generation via meta-learning. *CoRR* (2018)
10. J. Kiefer, J.W.: Stochastic estimation of the maximum of a regression function. *Ann. Math. Statist.* 23 (1952)
11. John Duchi, Elan Hazan, Y.S.: Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12 (2011)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *CoRR* (2014)
13. McCann, B., Keskar, N.S., Xiong, C., Socher, R.: The natural language decathlon: Multitask learning as question answering
14. Nitish Srivastava, Geoffrey Hinton, A.K.I.S.R.S.: Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15 (2014)
15. Pierre Baldi, K.H.: Neutal networks and principal component analyses: Learning from examples without local minima. *Neural Networks* (1989)
16. Sepp Hochreiter, J.S.: Long short-term memory. *Neural Computation* (1997)
17. Shi, T., Tatwawadi, K., Chakrabarti, K., Mao, Y., Polozov, O., Chen, W.: Incsql: Training incremental text-to-sql parsers with non-deterministic oracles
18. Thomas Mikolov, Kai Chen, G.C.J.D.: Efficient estimation of word representations in vector space (2013)
19. Tomas Mikolov, Ilya Sutskever, K.C.G.C.J.D.: Distributed representations of words and phrases and their compositionality (2013)
20. Victor Thong, Caiming Xiong, R.S.: Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR* (2017)
21. Wang, C., Tatwawadi, K., Brockschmidt, M., Huang, P.S., Mao, Y., Polozov, O., Singh, R.: Robust text-to-sql generation with execution-guided decoding
22. Wonseok Hwang, Jinyeong Yim, S.P.M.S.: Achieving 90(2019)
23. Xu, X., Liu, C., Song, D.: Sqlnet: Generating structured queries from natural language without reinforcement learning. *CoRR* (2017)
24. Yann LeCun, Patrick Haffner, L.B.Y.B.: Object recognition with gradient-based learning. *Proceeding Shape, Contour and Grouping in Computer Vision* (1999)
25. Yu, T., Li, Z., Zhang, Z., Zhang, R., Radev, D.R.: Typesql: Knowledge-based type-aware neural text-to-sql generation. *CoRR* (2018)
26. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. *CoRR* (2012)
27. Zellig, H.: Distributional structure. *Word* (1954)

Sparse Multimodal Classification of EEG Signals from Rapid Serial Visual Presentation of Diagnostic Images

Valentina Sulimova¹, Sergey Bukhonov², Olga Krasotkina³, Vadim Mottl^{1,4}, Annette Sterr⁶, Kevin Wells⁶, David Windridge^{5,6}

¹ Tula State University, Tula 300012, 92 Lenin Ave., Russia

² MIPT, Moscow, Russia

³ Markov Processes International, NJ 07901, 475 Springfield Ave, Suite 401, Summit, USA

⁴ Computing Center of the Russian Academy of Sciences, Moscow 119333, Vavilov St. 40, Russia

⁵ Middlesex University, London, UK

⁶ University of Surrey, Guildford, UK

vsulimova@yandex.ru, o.v.krasotkina@yandex.ru, vmottl@yandex.ru

Abstract. Automated identification of target classes within images is the principle aim of supervised computer vision, a key motivation for which is the reduction of human input into the recognition process. An alternative strategy, however, is to exploit the human pattern-recognition within the context of an EEG-based brain-computer interface in such a way as to minimize (or accelerate) the required human input. Observation of target/non-target images usually evokes a differential neural response in trained observers on a 300ms timescale, enabling EEG-based classification of brain signals in response to Rapid Serial Visual Presentation (RSVP) of target/non-target images. RSVP can hence significantly accelerate the recognition process, constituting a two-class problem with regard to the classification of multimodal EEG potentials (i.e. the problem of detecting whether the registered EEG signal contains a response to a target image or not).

In this paper, we propose two types of regularization procedure particularly applicable to SVM-based classification of RSVP event-related potentials that supplement the standard SVM formulation by automatically sparsely selecting the most informative features on the basis of neighborhood proximity.

The effectiveness of the proposed approaches is demonstrated with respect to recognition of mammograms with potential tumor pathology in an RSVP context. We obtain a recognition accuracy of 0.936 on the target data set, significantly improving the previous best result.

Keywords: EEG signals, brain-computer interface, RSVP, ERP-potentials recognition, mammograms with pathology recognition, SVM, regularization, feature selection.

1 Introduction

EEG (Electroencephalogram) signals denote the electrical activity of the brain [1, 2]. Analyzing EEG signals has been key in both supporting the diagnosis of brain diseases and for contributing to a better understanding of cognitive processes. Furthermore, it is actively used as the basis for building systems with a brain-computer interface (BCI) [3]. For instance, BCI is often used to restore sensory and motor functions in patients with motor disabilities. EEG-based BCI, however, extends across many application fields [4]. Recently, the possibility of using BCIs for identifying targets within different image types has emerged [5-7]. In particular, the use of BCI has been proposed in the medical domain as a novel high throughput screening method for mammographic diagnosis [7]. It thereby aims utilize the advanced recognition capacity of the human visual system, which has been found to be able to identify and classify image content as being either target or non-target with a frequency of more than 10 Hz [7]. This problem setting will be the focus of the current paper.

EEG-based BCIs typically operate based on the principle of evoking and detecting event-related potentials (ERPs) – i.e. the neural response of an observer associated with the presentation of certain stimuli or events [8]. Of the numerous methods of eliciting ERPs in BCI applications, in recent years, the rapid serial visual presentation (RSVP) paradigm has been among the most frequently employed. RSVP involves presenting the observer with a series of visual stimuli at a very fast rate (far more rapidly than would be typical for diagnosis), with stimuli being sequentially presented images or image-subcomponents presented in the center of the visual field [9, 10]. Of these stimuli, the ones that are of interest to the subject (depending on the task to be performed) are called the target stimuli. The subject's cognitive response as the target content is flashed up on the screen typically correlates with the enhancement or generation of specific ERP components in the EEG signal, with a much less prominent response in relation to the non-targets. Consequently, when using RSVP, the associated machine learning problem typically reduces to that of binary discrimination i.e. deciding whether the presented stimulus belongs to the target or non-target class via analysis of the EEG signal/detection the target ERP components.

However, the use of ERP detection for efficient detection and target-stimulus recognition in RSVP remains a difficult problem, requiring efficient signal processing and machine learning techniques; this is an active research area [11].

One of the difficulties consists in forming an appropriate feature space (which must naturally satisfy the compactness hypothesis [12] underlying any machine learning technique). Various approaches have sought to directly extract intelligent time-domain features from multiple electrodes [13-15]; however the majority of approaches use finite-length multiple EEG electrode samples as feature-vectors [16]. This involves a potentially large number of samples (as a rule, ~1000) and a significant number of electrodes (for example, 66 in [7]) with a consequent potential for model over-fitting. Non-redundant features are typically extracted by Principal Component Analysis [18], Independent Component Analysis [19], spatial filtering for maximizing the signal-to-signal-plus-noise ratio between target/non-target stimuli classes [20],

determination of the optimal linear combination of data from different electrodes [11] and so on [16]. All are applied before learning.

In this paper, event-related RSVP potential recognition is based on a modified Support Vector Machine (SVM) [20] strategy. To overcome over-fitting we propose two classes of regularization strategy, each expressing particular *a priori* preferences about the decision rule in such a way as to enable the SVM to automatically select the most informative features. Jointly with our approach to preprocessing of EEG signals, such regularization will allow a substantially increased quality of target/non-target signal recognition.

2 Classification of image-evoked EEG potentials as a two class pattern recognition problem

Farwell and Donchin [17] reported the first use of a P300- based brain computer interface, centered on the detection of a positive potential in the EEG about 300 msec after presentation of target images. A key benefit of this method is that images can be presented at high temporal rates (~10 per second), faster than that required for fully conscious detection, facilitating a high throughput of image material (even state-of-the-art computer vision approaches are unable to analyze and understand imagery as successfully as humans in complex domains such as breast cancer mammography). In a typical P300 experiment, participants must classify a series of stimuli which fall into one of two classes: targets and non-target, with targets appear more infrequently than non-targets. Assuming a differential neural response in the observer, the classification of EEG potentials consists in detection whether the registered EEG signal is a response to a target image or non-target; this is consequently a two-class pattern recognition problem.

We represent a single-electrode EEG signal as a real-valued discrete signal $\mathbf{x} = [x_1, \dots, x_m] \in R^m$; for multiple electrodes we concatenate the respective signals such that the signal's feature representation remains a vector. We hence use the same notation for signals from single and multiple electrodes.

Let $(X, Y) = \{\mathbf{x}_j, y_j\}, j = 1, \dots, N$ be a training set of EEG signals $\mathbf{x}_j \in R^m$, which are accompanied by a class label $y_j \in \{+1; -1\}$. $y_j = +1$ indicates that a target image was shown to a participant when EEG signal \mathbf{x}_j was registered; $y_j = -1$ indicates that it was not presented.

The aim of training is to form a decision function $\hat{y}(\mathbf{x})$ such that for any new EEG signals $\mathbf{x} \notin X$ it determines whether \mathbf{x} contains a response to a target image ($\hat{y}(\mathbf{x}) = +1$) or not ($\hat{y}(\mathbf{x}) = -1$).

We consider here three methods for classification of joint EEG signals: first, we use the standard Support Vector Machine approach as a baseline. We then incorporate various types of a priori information regarding the form of EEG signals in order to

make classification more robust. We then include a regularization strategy within the SVM criterion in order to pick-out those components of EEG signals important for target image classification. We also apply various preprocessing methods to the EEG signal in order to increase the signal-to-noise ratio.

3 Methodologies for two-class EEG potentials recognition

3.1 Baseline Support Vector Machine (SVM) method

The Support Vector Machine (SVM) method [21] is a well-established disseminative method for making decisions in a linear (potentially kernelized) feature space.

In this case, the decision rule obtains a linear hyperplane

$$d(\mathbf{x}; \mathbf{a}, b) = \mathbf{a}^T \mathbf{x} + b \quad \begin{cases} \geq 0 \Rightarrow \hat{y}(\mathbf{x}) = +1, \\ < 0 \Rightarrow \hat{y}(\mathbf{x}) = -1, \end{cases}$$

which is completely defined by the direction element $\mathbf{a} = [a_1, \dots, a_m]^T$ and a bias b . It hence separates objects of positive and negative classes, but supposes misclassifications δ_j for some training objects \mathbf{x}_j :

$$\begin{cases} \sum_{i=1}^m a_i^2 + C \sum_{j=1}^N \delta_j \rightarrow \min(\mathbf{a}, b, \boldsymbol{\delta}), \\ y_j (\sum_{i=1}^m a_i x_{ij} - b) \geq 1 - \delta_j, \quad j = 1, \dots, N, \\ \delta_j \geq 0, \quad j = 1, \dots, N. \end{cases} \quad (1)$$

where the parameter C defines a degree of influence of training objects misclassifications on a hyperplane.

3.2 Regularized SVM with decision-rule smoothness constraint

It is evident that in the problem of recognition of image-evoked EEG potentials there are *order relations* between object features. In this connection, it is natural to prefer a decision rule formulated such that its coefficients correspond to neighborhood feature-proximity, i.e. that the decision rule be a smooth one. The introduction of prior information regarding the sought hyperplane should consequently allow us to arrive at a more reliable decision, increasing its generalizing ability.

This requirement corresponds to form of regularization initially proposed the authors in [22]. To supplement the SVM with the potential to choose a smoothed decision rule, we add a specific regularization term $\sum_{i=2}^m (a_i - a_{i-1})^2$ into the optimization criterion in (1). Consequently, the modified SVM problem, incorporating the respective regularization may be formulated as:

$$\begin{cases} \sum_{i=1}^m a_i^2 + \gamma \sum_{i=2}^m (a_i - a_{i-1})^2 + C \sum_{j=1}^N \delta_j \rightarrow \min(\mathbf{a}, b, \boldsymbol{\delta}), \\ y_j (\sum_{i=1}^m a_i x_{ij} - b) \geq 1 - \delta_j, \quad j = 1, \dots, N, \\ \delta_j \geq 0, \quad j = 1, \dots, N, \end{cases} \quad (2)$$

where $\gamma > 0$ is the smoothness coefficient, which determines the degree of influence of the respective regularization member. The decision outcome of problem (2) can be easily found by the gradient descent method.

This order-dominated approach is hence very naturally suited to the recognition of image-evoked EEG potentials.

3.3 Regularized SVM with selectivity of informative EEG elements

Due to spatial and temporal signal localization, not all information from multiple electrodes is expected to be useful for making a decision about the presence or absence of a specific EEG potential. So, to increase recognition quality, it would be useful to have an instrument for automatic selection of the most informative features. For this purpose we introduce a second type of regularization within the SVM optimization.

We thus apply a specific instance of the generalized probabilistic formulation of the SVM initially proposed by the authors [24] for the case of multimodal recognition, wherein each modality is represented by some kernel function, and which was successfully applied to membrane protein prediction [25]. In this paper, we reformulate this approach for the case of recognition in a linear feature space and label it *Supervised Selective Feature Support Vector Machines* (SFSVM).

The training optimization criterion for the considered type of regularization can be written as follows:

$$\begin{cases} J_{SFSVM}(a_1, \dots, a_m, b, \delta_1, \dots, \delta_N, C, \mu) = \sum_{i=1}^m q(a_i | \mu) + C \sum_{j=1}^N \delta_j \rightarrow \min(\mathbf{a}, b, \boldsymbol{\delta}), \\ q(a_i | \mu) = \begin{cases} 2\mu |a_i| & \text{if } |a_i| \leq \mu, \\ \mu^2 + a_i^2 & \text{if } |a_i| > \mu, \end{cases} \\ y_j (\mathbf{a}^T \mathbf{x}_j + b) \geq 1 - \delta_j, \quad \delta_j \geq 0, \quad j=1, \dots, N. \end{cases} \quad (3)$$

The proposed training criterion is thus a generalized version of the classical SVM that implements the principle of *feature selection*.

The threshold μ is named here a "selectivity" parameter because it regulates the ability of the criterion to enact the selection of features. When μ is equal to 0, the criterion is equivalent to the classical SVM (1) with the minimal ability to select features; values much greater than zero are equivalent to the Lasso SVM [26], with increasing selectivity as μ grows up to full suppression of all features.

The solution of problem (3) is equivalent to the solution $(\hat{\xi}_i \geq 0, i \in I = \{1, \dots, m\}, \hat{\lambda}_j \geq 0, j = 1, \dots, N)$ of the dual problem

$$\begin{cases} L(\lambda_1, \dots, \lambda_N | C, \mu) = \sum_{j=1}^N \lambda_j - \sum_{i \in I} \frac{\xi_i}{2} \rightarrow \max(\lambda_1, \dots, \lambda_N, \xi_1, \dots, \xi_N), \\ \xi_i \geq 0, \quad \xi_i \geq \sum_{j=1}^N \sum_{l=1}^N y_j y_l x_{ij} x_{il} \lambda_j \lambda_l - \mu^2, \quad i \in I = \{1, \dots, m\}, \\ \sum_{j=1}^N y_j \lambda_j = 0, \quad 0 \leq \lambda_j \leq \frac{C}{2}, \quad j = 1, \dots, N, \end{cases}$$

and can be expressed in the form

$$\begin{cases} \hat{a}_i = \sum_{j: \hat{\lambda}_j > 0} y_j \hat{\lambda}_j x_i(\omega_j), \\ i \in I^+ = \left\{ i \in I: \sum_{j=1}^N \sum_{l=1}^N y_j y_l x_{ij} x_{il} \hat{\lambda}_j \hat{\lambda}_l > \mu^2 \right\}, \\ \hat{a}_i = \hat{\eta}_i \sum_{j: \hat{\lambda}_j > 0} y_j \hat{\lambda}_j x_i(\omega_j), \\ i \in I^0 = \left\{ i \in I: \sum_{j=1}^N \sum_{l=1}^N y_j y_l x_{ij} x_{il} \hat{\lambda}_j \hat{\lambda}_l = \mu^2 \right\}, \\ \hat{a}_i = 0, \quad i \in I^- = \left\{ i \in I: \sum_{j=1}^N \sum_{l=1}^N y_j y_l x_{ij} x_{il} \hat{\lambda}_j \hat{\lambda}_l < \mu^2 \right\}, \end{cases}$$

where $\{0 \leq \eta_i \leq 1, i \in I^0\}$ are additionally computed coefficients.

It should be noted that criterion (3), in contrast to other criteria of feature selection [26,27], explicitly splits the entire set of features into two subsets: “support” features $I^+ \cup I^0$ (which will participate in the resulting discriminant hyperplane) and excluded features I^- .

As a result, the optimal discriminant hyperplane, which is defined by the solution of the SFSVM training problem (3), can be expressed as

$$\sum_{j: \lambda_j > 0} y_j \lambda_j \left(\sum_{i \in I^+} x_{ij} x_i + \sum_{i \in I^0} \eta_i x_{ij} x_i \right) + b \geq 0,$$

where the numerical parameters $\{0 \leq \eta_i \leq 1, i \in I^0; b\}$ are solutions of the linear programming problem:

$$\begin{cases} 2\mu^2 \sum_{i \in I^0} \eta_i + C \sum_{l=1}^N \delta_j \rightarrow \min(\eta_i, i \in I^0; b; \delta_1, \dots, \delta_N), \\ \sum_{i \in I^0} \left(\sum_{l=1}^N y_j y_l x_{ij} x_{il} \lambda_l \right) \eta_i + y_j b + \delta_j \geq 1 - \sum_{i \in I^+} \left(\sum_{l=1}^N y_j y_l x_{ij} x_{il} \lambda_l \right), \\ \delta_j \geq 0, \quad j = 1, \dots, N, \quad 0 \leq \eta_i \leq 1, \quad i \in I^0. \end{cases}$$

Consequently, the proposed approach has a very significant qualitative advantage over the other methods – it explicitly indicates a discrete subset of support features within the combination, in contrast to other methods that assign some positive (if small) weight to *each* feature and, so, require significantly greater amount of memory, and run the risk of overfitting.

4 High throughput mammographic recognition of image-evoked EEG potentials

4.1 Data description

We seek to demonstrate that the proposed regularization techniques allow for improved recognition quality of image-evoked EEG potentials with respect to high throughput screening for mammography. For this study, we use the same EEG data as in 7 (which contains details of the RSVP methodology). We hence use fragments of EEG signals that were collected from mammography experts while watching a series of RSVP-presented mammograms using 66 electrodes.

For each of the 66 electrodes, EEG fragments have following characteristics:

- 1) Each EEG fragment is a 1100 ms signal corresponding to the presentation of 11 mammograms (100 ms per mammogram).
- 2) For fragments of the target class (target objects) one of the 11 mammograms (randomly from 4 to 7 fragments) contains some pathology.
- 3) For fragments of the non-target class (non-target objects) all of 11 mammograms are free of pathology.

Before separation into fragments, initial EEG signals from each electrode are filtered with a cutoff frequency of 40 Hz. The full set of obtained 755 objects was randomly split into train set (98 target and 98 non target objects) and test set (275 target and 284 non target objects).

4.2 EEG fragment preprocessing

In this work, to improve the speed and quality of recognition we apply 2 types of preprocessing for each fragment of EEG signal: thinning and moving average filtering in a window.

Let $\mathbf{x} = (x_i \in \mathbb{R}, i = 1, \dots, m)$, $m = 1100$ be the initial EEG fragment.

In accordance with the Kotelnikov's theorem [28] initial EEG fragments are redundant, and can be thinned up to 12.5 times without losing information. In this connection, we thin-out each initial EEG fragment in $step = 11$ times. As a result we obtain 100-length EEG fragments instead of 1100-length ones: $\mathbf{x}' = (x'_i \in \mathbb{R}, i = 1, \dots, m')$, $m' = m / step = 1100 / 11 = 100$. Thinning allows us to reduce the dimensionality of the feature space, the amount of required memory, and a number of computation required.

To decrease influence of the noise component, we filter signals by moving average filtering with window size $w = 11$:

$$\begin{cases} x''_i = \frac{1}{w} \sum_{k=i-\lfloor w/2 \rfloor}^{i+\lfloor w/2 \rfloor} x'_k, & i = \lfloor w/2 \rfloor + 1, \dots, m' - \lfloor w/2 \rfloor, \\ x''_i = x'_{\lfloor w/2 \rfloor + 1}, i < \lfloor w/2 \rfloor + 1, & x''_i = x'_{m' - \lfloor w/2 \rfloor}, i > m' - \lfloor w/2 \rfloor. \end{cases}$$

4.3 Experimental results

First of all, we constructed decision rules for each of electrodes separately for 4 modes: with filtering and without filtering, taking into account the decision rule smoothness and not taking this into account. The quality of obtained decision rules was estimated by the area under ROC-curve (AUC) for the test set.

The choice of smoothness coefficient γ for the smoothness regularization was made automatically for each electrode using the leave-one-out procedure for the training set.

The respective results are presented in table 1. The best result for each electrode is displayed in bold font.

As may be seen from table 1, in most cases the best result is obtained when we use moving average filtering in combination with decision-rule smoothness regularization. It should be noted that, in cases when the best quality is observed for modes without moving average filtering and(or) without regularization, the decision rule quality, as a rule, is small - i.e. such situations tend to occur for electrodes that do not contain much useful information. In other cases, such effects may be accounted for by an inappropriate smoothing window width (which was equal to 11 in all cases, and not specifically chosen for each electrode).

Table 1. AUC-values for different electrodes and different training modes

Smoothing	-	-	+	+	smoothing	-	-	+	+		
	smoothness of the decision rule	-	+	-		+	smoothness of the decision rule	-	+	-	+
Electrode's number	1	0,6858	0,6855	0,657	0,6985	Electrode's number	34	0,5012	0,5036	0,4781	0,4944
	2	0,6431	0,6192	0,6009	0,6611		35	0,5012	0,576	0,5493	0,5558
	3	0,6644	0,6758	0,6308	0,6901		36	0,6348	0,6711	0,63	0,6985
	4	0,6683	0,6473	0,6411	0,689		37	0,7203	0,8257	0,7364	0,8451
	5	0,6582	0,6174	0,6124	0,6167		38	0,6085	0,696	0,6259	0,6997
	6	0,6425	0,6791	0,6292	0,6806		39	0,7062	0,6931	0,6811	0,7431
	7	0,6091	0,5954	0,5827	0,5816		40	0,5897	0,5857	0,5464	0,547
	8	0,6766	0,6778	0,6242	0,6873		41	0,6088	0,655	0,6291	0,6742
	9	0,5714	0,6161	0,5782	0,6267		42	0,7708	0,8584	0,7815	0,8672
	10	0,6053	0,555	0,5384	0,5746		43	0,5378	0,5366	0,5268	0,5518
	11	0,5509	0,5613	0,5724	0,576		44	0,545	0,5642	0,5664	0,5735
	12	0,5929	0,5478	0,5504	0,5593		45	0,5924	0,6715	0,6281	0,6961
	13	0,5368	0,5254	0,5541	0,5589		46	0,6428	0,7447	0,6951	0,7671
	14	0,5295	0,5405	0,5226	0,5397		47	0,5491	0,6056	0,5669	0,5759
	15	0,5595	0,6175	0,5876	0,5891		48	0,5628	0,6101	0,5775	0,5869
	16	0,5191	0,5157	0,5171	0,5119		49	0,5713	0,5866	0,5371	0,5455
	17	0,5756	0,5609	0,563	0,5891		50	0,4918	0,5419	0,5268	0,5388
	18	0,5844	0,5601	0,5798	0,6629		51	0,5852	0,603	0,575	0,6005
	19	0,6222	0,5994	0,5969	0,6692		52	0,5622	0,5515	0,5829	0,5787

20	0,5214	0,6301	0,5672	0,6316	53	0,6943	0,7688	0,7287	0,7855
21	0,6377	0,6699	0,6202	0,6944	54	0,5702	0,6184	0,5608	0,5666
22	0,5103	0,508	0,5191	0,5108	55	0,5238	0,5485	0,5354	0,5248
23	0,5471	0,5877	0,5473	0,5952	56	0,6548	0,6617	0,6275	0,6805
24	0,616	0,6358	0,585	0,6594	57	0,6458	0,6638	0,6382	0,6146
25	0,488	0,4853	0,5191	0,5177	58	0,7606	0,7137	0,6748	0,7351
26	0,4931	0,4864	0,5142	0,5072	59	0,7382	0,7115	0,6733	0,6547
27	0,748	0,8083	0,7277	0,8289	60	0,7493	0,7234	0,6854	0,7472
28	0,7385	0,8305	0,7112	0,8351	61	0,5537	0,5049	0,5399	0,5127
29	0,6363	0,6808	0,6716	0,663	62	0,6219	0,6625	0,5983	0,6011
30	0,7717	0,8177	0,7439	0,8252	63	0,7336	0,7465	0,7009	0,7453
31	0,5568	0,5786	0,5422	0,5428	64	0,6066	0,6001	0,5831	0,5783
32	0,6661	0,6812	0,644	0,7248	65	0,4933	0,4814	0,5129	0,5021
33	0,7652	0,8308	0,7312	0,8518	66	0,6191	0,6232	0,5492	0,5475
all	0,764	0,805	0,815	0,811					

In addition, table 1 contains results for simultaneous use of all 66 electrodes. As anticipated, the quality of the decision rule for all electrodes is worse than the quality of decision rules for a specific subset of separate electrodes. This confirms the existence of overfitting in RSVP and the necessity of reducing the dimensionality of the feature space.

For the next stage of experiments, we choose 7 electrodes with the highest recognition quality. These are electrodes 27, 28, 30, 33, 37, 42 and 53. We combined the signals from these electrodes, used the moving average filtering and applied the SVM with decision rule smoothness regularization. Under these assumptions an AUC=0,906 was achieved for the test set. This value exceeds the result for all electrodes and for each individual electrode. It confirms again the overfitting problem and its amelioration by selectively decreasing the number of electrodes and, respectively the number of features.

Finally, we apply the sparse regularization proposed in (section 3.3) for smart dimensionality reduction of the feature space.

In this experiment 13 electrodes were utilized, showing a differential recognition quality (low and high) in accordance with Table 1; the respective electrode numbers can be seen in Figure 1. For this set of electrodes, the training was in accordance with the proposed SFSVM method (section 3.3), and was conducted a number of times for different selectivity values.

For each selectivity value given in Figure 1 the achieved AUC value is presented. Features that remained after training are shown as blue bars.

Figure 1 hence shows that an increase of selectivity value leads to discarding of features at the start and end positions of the EEG fragments. This seems to be because the features are uninformative due to smoothing by the sliding window. We observe that (up to a threshold) an increase of selectivity value leads to an increase in the AUC.

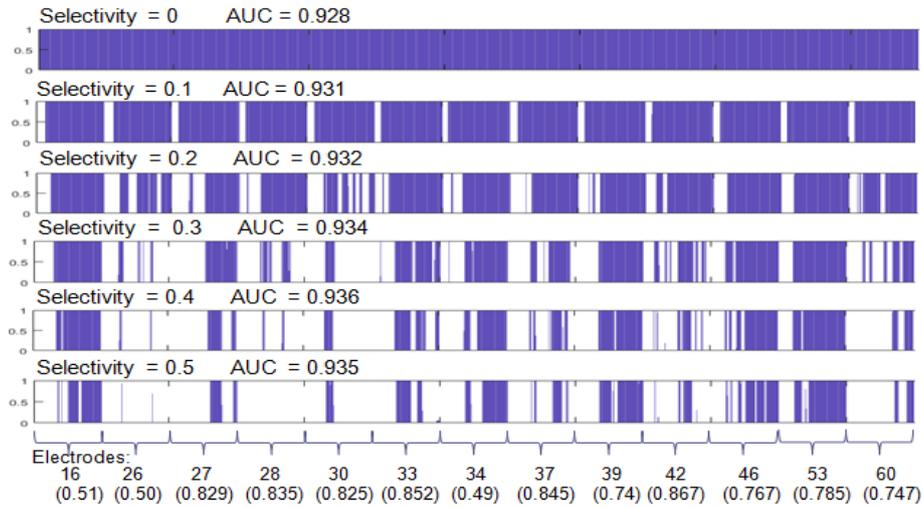


Figure 1. The results of informative feature selection in accordance with SFSVM for different selectivity values

The best recognition quality in this case is 0.936, which significantly improves on results obtained in previous experiments. It should be noted that results for both types of regularization essentially improves on the best result obtained in [7] for joint training with multiple electrodes, which for the considered data was 0.88.

5 Conclusion

In this paper we proposed a specific strategy for preprocessing of EEG signals in a RSVP context, along with two types of regularization for SVMs in order to improve the speed and quality of recognition of image-evoked EEG potentials.

The effectiveness of the proposed approaches was demonstrated on the example of recognition of mammograms with pathology using a brain-computer interface. The best recognition quality obtained on the data set is 0.936, which significantly improves on the best result reported in previous work for joint training with a number of electrodes.

The work supported by the Russian Foundation for Basic Research, projects 18-07-01087, 17-07-00993, 17-07-00436. This work was also supported by the EU 2020 project Dreams4Cars, grant number 731593.

References

1. Zenkov L. R. Clinical electroencephalography. /3-d edition. — Moscow: MEDpress-inform, 2004. — 368 p. ISBN 5-901712-21-8
2. Teplan, M. 2002. Fundamentals of EEG measurement. Measurement Science Review 2(2), pp. 1-11

3. Wolpaw J.R., McFarland D.J., Neat G.W., Forneris C.A., An EEG-based brain-computer interface for cursor control. *Electroencephalography & Clinical Neurophysiology*. Vol 78(3), Mar 1991, 252—259
4. Eizagirre A., Vall A. and D. at al. *EEG/ERP Analysis: Methods and Applications* (2014).
5. Poolman P, Frank R M, Luu P, Pederson S M and Tucker D M 2008 A single-trial analytic framework for eeg analysis and its application to target detection and classification *NeuroImage* 42 787–98 (2008)
6. Sajda P et al 2014 Evoked neural responses to events in video *IEEE J. Sel. Top. Signal Process.* 8 358–65 (2014)
7. C.Hope, A. Sterr,P.E. Langovan, N.Geades, D.Windridge, K.Young, K.Wells.High Throughput Screening for Mammography using a Human-Computer Interface with Rapid Serial Visual Presentation (RSVP) / - *Proc. SPIE 8673, Medical Imaging 2013: Image Perception, Observer Performance, and Technology Assessment*, 867303 (March 28, 2013); doi:10.1117/12.2007557.
8. S. Sur and V. Sinha. Event-related potential: An overview. *Industrial Psychiatry Journal* 18(1), pp. 70. 2009.
9. C. Keysers, D. Xiao, P. Földiák and D. Perrett. The speed of sight. *J. Cogn. Neurosci.* 13(1), pp. 90-101. 2001.
10. O. de Bruijn and R. Spence. Rapid serial visual presentation: A space-time trade-off in information presentation. Presented at *Advanced Vis. Interf.* (2000).
11. Cecotti, H., Eckstein, M.P., Giesbrecht, B. Single-trial classification of event-related potentials in rapid serial visual presentation tasks using supervised spatial filtering. *IEEE Trans. Neural Netw.Learn. Syst.* 25, 2030-2042 (2014)
12. Braverman, E. M. Experiments on training a machine for pattern recognition. PhD Thesis. Moscow (1961)
13. V. Abootalebi, M. H. Moradi and M. A. Khalilzadeh. A new approach for EEG feature extraction in P300-based lie detection. *Comput. Methods Programs Biomed.* 94(1), pp. 48-57. (2009).
14. Z. Amini, V. Abootalebi and M. T. Sadeghi. Comparison of performance of different feature extraction methods in detection of P300. *Biocybernetics and Biomedical Engineering* 33(1), pp. 3-20. (2013).
15. Tran, L. EEG Features for the Detection of Event-Related Potentials Evoked Using Rapid Serial Visual Presentation. PhD Thesis. 63 p.(2014)
16. Lees S., Dayan N., Cecotti H., McCullagh P., Maguire L., Lotte F., Coyle D.. A review of rapid serial visual presentation-based brain-computer interfaces. *J Neural Eng.* 2018 Apr;15(2):021001. doi: 10.1088/1741-2552/aa9817. (2018)
17. Farwell LA, Donchin E: Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *EEG Clin Neuroph.* 1988, 70:510–523.
18. Alpert G. F, Manor R, Spanier A B, Deouel L Y and Geva A B. Spatiotemporal representations of rapid visual target detection: a single-trial EEG classification algorithm *IEEE Trans. Biomed. Eng.* 61 2290–303 (2014)
19. Kumar S. and Sahin F. Brain computer interface for interactive and intelligent image search and retrieval. *High Capacity Optical Networks and Emerging/Enabling Technologies (IEEE)*, pp. 136–40 (2013)
20. Rivet B et al. xDAWN algorithm to enhance evoked potentials: application to brain–computer interface *IEEE Trans. Biomed. Eng.* 56 2035–43 (2009)

21. Vapnik, V.N. *Statistical Learning Theory* / — Wiley-Interscience, 1998. 768 p. (1998).
22. Mottl V.V., Dvoenko S.D., Seredin O.S., Krasotkina O.V. Pattern recognition learning taking into account the criterion of smoothness of the decision rule. // *Control and Inform.: proc. of the chair of autom. and rem. contr. of TSU.* (2000)
23. Tatarchuk, A., Mottl, V., Eliseyev, A., Windridge, D.: Selectivity supervision in combining pattern-recognition modalities by feature- and kernel-selective Support Vector Machines. *Proc. ICPR* (2008).
24. Tatarchuk, A., Urlov, E., Mottl, V., Windridge, D.: A support kernel machine for supervised selective combining of diverse pattern-recognition modalities. In: El Gayar, N., Kittler, J., Roli, F. (eds.) *MCS* (2010).
25. Tatarchuk A., Sulimova V., Mottl V., Windridge D. Supervised Selective Kernel Fusion for Membrane Protein Prediction. M. Comin et al. (Eds.): *Pattern Recognition in Bioinformatics, Lecture Notes in Computer Science Volume 8626*, 2014, pp.98-109. (2014)
26. Bradley P., Mangasarian O.: Feature selection via concave minimization and support vector machines. In *International Conf. on Machine Learning* (1998)
27. Wang, L., Zhu, J., Zou, H.: The doubly regularized support vector machine. *Statistica Sinica*, 01/2006; 16:589–615 (2006)
28. Burachenko D.I., Kluev N.N., Kotjik V.I., Fink V.I. et al. *General communication theory.* / Fink L.M. eds. – L.: BAC, 1970. – 412c.
29. S. J. Luck, *An Introduction to the Event-Related Potential Technique.* Cambridge, MA, USA: MIT Press, (2005)
30. K. E. Hild, M. Kurimo, and V. D. Calhoun, “The sixth annual MLSP competition, 2010,” in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, Kittila, Finland, Sep. 2010, pp. 107–111.
31. Gray, H. M., Ambady, N., Lowenthal, W. T., & Deldin, P. (2004). P300 as an index of attention to self-relevant stimuli. *Journal of Experimental Social Psychology*, 40(2), 216-224

Neural-Attention Multi-Instance Learning for Predicting User Demographics from Highly Noisy Tweets

Qing Chen, Mingxuan Sun, and Jian Zhang

Louisiana State University, Baton Rouge LA 70803, USA
qchen11@lsu.edu {msun, zhang}@csc.lsu.edu

Abstract. A user’s demographic traits such as age, ethnicity and education levels are valuable information for various online services. We consider the problem of predicting these traits from a collection of tweets generated by online users. Our experiments showed that deep neural networks (DNNs) do not necessarily perform better than traditional models for such prediction, despite their wide success in other text-mining tasks. The trait prediction has some unique characteristics: It is a multi-instance learning scenario, where each user is associated with a bag of tweets and the bags are labeled but the individual tweets are not. Furthermore, tweets are often highly unstructured and noisy (i.e., not all tweets are relevant to the target trait). To deal with the noisy multi-instance learning situation, we propose a deep neural network with a neural-attention mechanism. Similar to other DNNs, our model is capable of learning text patterns of different sizes and complexities automatically. On the other hand, it also has advantage over existing DNNs in that it can learn the relevance of each individual tweet and make a prediction based on the selected relevant information. Experiment results on real-world datasets showed that our model outperforms not only traditional text classification methods such as SVM but also other deep neural-networks such as LSTMs and CNNs.

Keywords: User demographic traits · Deep learning · Multi-instance learning.

1 Introduction

User demographic traits such as age, gender and ethnicity information are important attributes for various online services such as personalized recommendation, marketing, public health and social study. Most online applications may explicitly solicit user demographics through user registration. However, online users do not always provide such information due to privacy concern [1]. On the other hand, user interactions such as tweets or comments may provide an alternative way to infer demographic information. We consider the problem of predicting demographic traits from a collection of user tweets.

The prediction problem is usually tackled by learning a model using tweets from annotated users. In previous approaches, the aggregation of each user’s

tweets is converted into a bag-of-words (BoW) feature vector and a model, for example, Support Vector Machine (SVM) or logistic regression, based on this representation is trained to make predictions [23, 24]. In recent years, deep neural networks (DNNs) have been shown to achieve the state of the art performance in many text-mining tasks such as sentiment analysis [6, 10, 26]. It is natural to consider DNNs for the trait prediction problem. Along this direction, we applied two widely used deep neural networks for text classification, the 1-d convolutional neural network (CNN) and the long short-term memory network (LSTM), to predict traits. However, in most cases, the performance of the DNNs were on par or slightly better than that of the traditional SVM, different from many other application scenarios where DNNs outperform traditional models significantly. This leads us to ask the question whether the DNNs can be further improved for trait prediction to show clear advantage over the traditional models.

On a close look at the prediction problem, we notice that the problem is a multi-instance learning (MIL) problem [4]. In MIL, an entity is described by a bag of multiple instances of information (e.g., multiple tweets) and the task is to learn a model that can predict the class label (e.g. some trait) of the entity based on the instances. Only the label of the whole bag is known. There is no label for individual instances (individual tweets). We also noticed that our problem is highly noisy in the sense that not all instances are relevant to the prediction of a label. For example, among the tweets we collected from a user, some may carry the information about the user’s age but most may be completely age-irrelevant. To make an accurate prediction, it is important to select and focus on the relevant instances (tweets) while ignoring the ones not related.

In this paper, we propose a novel neural attention mechanism for better multi-instance prediction using deep neural networks. One of the reasons that make deep neural networks powerful is feature learning. We view the classification done by a deep neural network as a two-step process: the network first extracts learned features from the input data and then makes a prediction based on these features. For the user-trait prediction problem, we use a deep neural network to extract features from each tweet. The neural attention mechanism then determines how relevant each tweet is with respect to the prediction target based on these features. We aggregate the features from a bag of tweets by summation, in which each feature vector is weighted by the relevance of the corresponding tweet. (Effectively, the network pays more attention to features from relevant tweets and less to those irrelevant.) Because a final prediction is made using the aggregation of the weighted features, the attention mechanism ensures that the decision is rooted on the (selected) features from the relevant tweets. Additionally, in our neural network design, the attention mechanism (an attention network) is an integrated part of the deep neural-network architecture. It is not hand-coded. Rather it is trained together with the whole deep network architecture in an end-to-end fashion. We further remark that although attention mechanism is not new with deep neural networks, to our best knowledge, ours is the first that deals with multi-instance learning for text with high noise.

We conducted a set of experiments to evaluate the proposed model and compare it to existing BoW-based approaches and DNNs for text classification. The results show that our model renders the best performance to most prediction tasks on multiple measures. The rest of the paper is organized as follows: In Section 2 we discuss related work. We present the details of our model in Section 3 and the results of our experiments in Section 4. Section 5 concludes the paper.

2 Related Work

2.1 Multi-Instance Learning

Multi-instance learning [4] (MIL) tackles the problem where each training example is regarded as a bag containing multiple instances and associated with one class label. Zhou and Zhang addressed a scene classification problem in MIL method by generating multiple patches as instances from an image as a bag, labelled each instance with the class of the bag, and achieved high performance on image classification by aggregating the classification scores of patches [29]. Dong et al. proposed a CNN-based neural network model to classify images [5]. In this study features from each group of CNN layers were treated as instances, and the image was classified based on the concatenation of the features. Our work tackles text processing and utilizes a neural attention mechanism to weight the relevance of the tweets before fusing tweets features together instead of pure concatenation.

2.2 Demographics Inference

Extensive studies have focused on demographic inference from various online human activities. Studies including [7, 14, 15] demonstrate that it is possible to infer private user attributes from online social networks given a small fraction of users who are willing to provide their private attributes such as locations and interests. Predicting demographic attributes from the content of user writings is often formulated as a classification problem [2, 17–19]. Most of the approaches adopt shallow models with traditional text feature representations such as BoW. For example, Volkova and Bachrach transformed tweets into feature vectors using BoW, and employed SVM to classify the tweets into different emotional tones [24]. However, massive tweets posted by online users are usually highly unstructured and noisy, which makes text analysis with traditional feature engineering inapplicable in this application scenario.

2.3 Deep Learning for Text Processing

Deep learning has also been successfully applied in text classification tasks [6, 10, 13, 26, 28]. The application of deep learning in demographic inference in microblogging system, however, is more challenging. Riemer et al. proposed a deep learning network to extract tweets representations, and achieved better accuracy

for age prediction compared with N-gram based techniques [20]. Liu and Inkpen trained stacked denoising autoencoders for predicting the locations of Twitter users such as their regions, states, and geographical coordinates based merely on the users’ tweets [12]. These works first converted the original texts into feature vectors by bag-of-words or by paragraph vector techniques and then fed the feature vectors into neural networks, which is different from our approach. We obtain the feature vectors from the original texts directly through the neural network. Our framework also discovers text patterns of different sizes and complexities automatically, and learns the relevance of each individual tweet.

3 Neural-Attention Multi-Instance Learning

3.1 Problem Definition

We have a set of N users and for each user u , we have a collection of k tweets $\Gamma_u = \{T_1^{(u)}, T_2^{(u)}, \dots, T_k^{(u)}\}$ that the user tweeted in the past. A tweet is a sequence of words $T = w_1 w_2 \dots$ where the words (w_i) are from a fixed dictionary. A user is also associated with a demographic class (e.g., ethnicity class, income class, etc). Let y_u be the class label of the user u . We would like to construct a model \mathcal{F} that given the collection of tweets Γ_u , predicts the class label y_u , i.e., $\mathcal{F}(\Gamma_u) \rightarrow y_u$.

Since tweets are the input to our model, we describe first the representation of the raw tweets. A matrix is used to represent a tweet. The matrix has the same number of rows as the number of words in the tweet and each row is a one-hot vector for the corresponding word. A one-hot vector for a word is an all-zero vector except at the entry corresponding to the index of the word in the dictionary, where it has a value one. (Hence, the length of the vector is the same as the size of the dictionary.) The matrix representing the tweet has a dimension of $l \times D$ where D is the size of the dictionary and l is the length of the tweet in words. To represent multiple tweets in a uniform fashion, we make all tweets the same length by padding the shorter ones with a null word, whose one-hot vector is all zero. We call such matrix representation of a tweet one-hot matrix and denote by $\mathbf{T}_i^{(u)} \in R^{l \times D}$ the one-hot matrix for the i -th tweet from user u . The collection of one-hot matrices for the user is referred to as $\mathbf{\Gamma}_u$.

3.2 Background

Besides the attention-assisted feature fusion, our model also employs components that performs feature extraction and final decision making. In particular, an inception-like module [21] is used for feature extraction and a max-margin classifier is used for final prediction. We introduce these components in this subsection. We’d like to remark that although our model consists of several modules/components, one does not need to train them separately. The model is an end-to-end solution. There is no intermediate information to calculate and the parameters of the model can be trained simultaneously, given the tweets and the user labels.

Max-Margin Classifier We focus on two-class classification here. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ be the feature vectors representing N data points and $Y = \{y_1, y_2, \dots, y_N\}$, $y_i \in \{-1, 1\}$ be the corresponding labels of the data points. A (linear) max-margin classifier (i.e., SVM) labels a data point \mathbf{x} as class "1" if $\mathbf{w}\mathbf{x} + b > 0$ and class "-1" otherwise. \mathbf{w} and b are the parameters of the classifier that are obtained by training using the following optimization:

$$\min_{w,b} \frac{1}{N} \sum_i [1 - y_i \cdot (\mathbf{w}\mathbf{x}_i + b)]_+ + \lambda \|\mathbf{w}\|^2, \quad (1)$$

where $[\cdot]_+$ is the hinge loss function, i.e.,

$$[x]_+ = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

and λ is the tradeoff parameter between the hinge loss and the L2-regularization term.

Inception-like CNN We use convolutional neural networks (CNNs) to extract features from the tweets. Although CNNs are often utilized for image features, there has been work applying CNN on text data [9, 11] that showed the effectiveness of the CNNs.

Our inference is based on a set of tweets. We first describe the convolutional operation for a single tweet. For a set of tweets, we repeat the same operation on each of them. Let $\mathbf{Z} \in \mathbb{R}^{l \times d}$ be the input to a convolutional layer. In the scenario of a tweet, the input to the first convolutional layer is the embedding matrix of the tweet, i.e., l is the length of the tweet and d is the length of the embedding vector. (The detail of the embedding process is given in the next section.) The input to the successive convolutional layers is the output from the previous layer ($\mathbf{Z}' \in \mathbb{R}^{l \times d'}$). In this case, l is still the length of the tweet and d' is the number of convolution filters. (We use padding on the boundaries to make the output of the convolution operation the same length as that of the input. Hence l stays the same across layers.)

With text data, the convolution is 1-dimensional, i.e., convolving along the word sequence of the tweet. Given a window size s , we denote by $\mathbf{Z}(t)$ the local window at location t , i.e., $\mathbf{Z}(t) \in \mathbb{R}^{s \times d}$ is a part of the matrix \mathbf{Z} starting from row t and spanning s rows. Let $\mathbf{Z}'(t, i)$ be the corresponding unit on the feature map of the i -th filter. We have

$$\mathbf{Z}'(t, i) = \text{relu}(\mathbf{W}^{(i)} \odot \mathbf{Z}(t) + b_i), \quad (3)$$

where relu is the rectified linear function. $\mathbf{W}^i \in \mathbb{R}^{s \times d}$ is the i -th filter's parameter matrix and b_i is the filter's bias. \odot is an operator that takes two matrices \mathbf{U} and \mathbf{V} and outputs $\sum_{i,j} \mathbf{U}_{i,j} \cdot \mathbf{V}_{i,j}$. With padding, the feature map from each filter is a vector of length l . When d' filters are used, the collection of feature maps form a matrix of size $l \times d'$. The feature maps then serve as the input to the next convolutional layer.

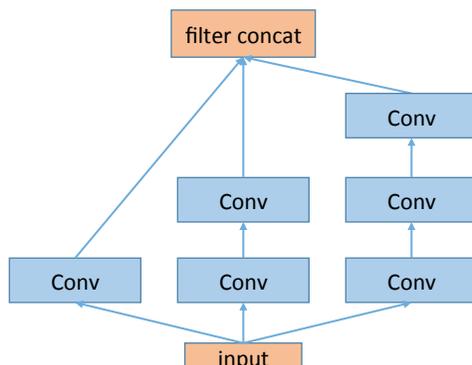


Fig. 1: CNN module used in the model.

A set of convolutional layers are used in our model. Fig. 1 shows the structure of the module that uses these layers. The structure is similar to the inception module in the GoogleNet [21]. It consists of a few stacks of k convolutional layers, with k ranging from 1 to 3. Although the window size is the same for all the filters in the convolutional layers, the neurons in different stacks have effective receptive fields of different sizes. Furthermore, such a structure allows extracting features at different depths and thus can be of different complexities. This is the reason that we choose it as our CNN module.

3.3 Prediction Model

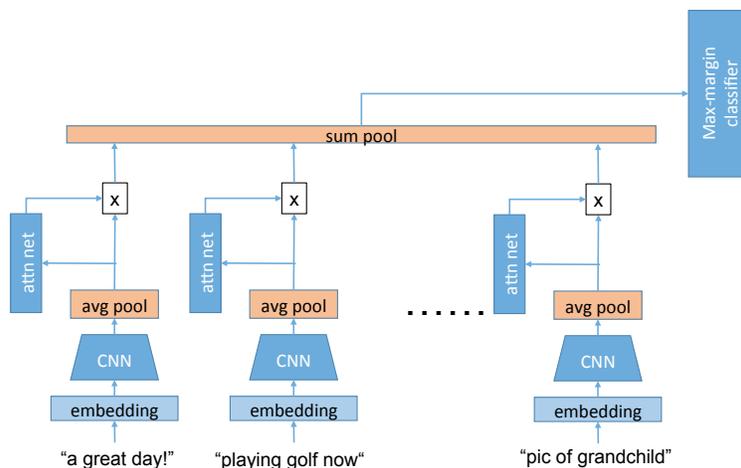


Fig. 2: Model architecture with CNN, attention net and max-margin classifier.

Overall, our model consists of two main stages: at the first stage, using DNNs, we map tweets (the observable) to features (hidden variables). At the second stage, using a max-margin classifier, we decide the label of a user based on the features. This is a discriminative approach. Fig. 2 shows the architecture of the model. The first step in processing a tweet is word embedding. It converts

words in the tweet into embedding vectors. Let \mathbf{E} be the embedding matrix. The process is a linear transformation, i.e., given a one-hot matrix \mathbf{T} of a tweet, the embedding generates: $\mathbf{Z} = \mathbf{T} \cdot \mathbf{E}$. Note that the matrix \mathbf{E} is one of the parameters of the model whose value will be obtained through training.

The result of embedding/linear transformation is forwarded to the CNN module that performs feature extraction. All the CNN components in Fig. 2 are identical and have the same structure as depicted in Fig. 1. As described in the previous section, the output of the CNN module for a tweet is a collection of feature maps with each feature map being a vector of the same length as the tweet. (The collection forms a matrix.) We collapse a feature map into a single value by (average) pooling globally over the whole feature map (the pooling layer in Fig. 2 for each tweet). After this, a tweet is represented by a (aggregated) feature vector whose length is equal to the number of feature maps. We refer to the feature vector of the i -th tweet from the user u as $\mathbf{z}_{u,i}$.

Attention Mechanism For each user, we have a collection of tweets and hence a collection of feature vectors representing the tweets. The goal of the attention mechanism is to learn the relevance of a tweet with respect to the classification. Note that in our scenario, the tweets may not be related to each other in most cases. Hence, relevance (attention) calculation is independent across the tweets, i.e., we compute the relevance (attention) weight for a particular tweet using that tweet only, without considering other tweets from the same user. This is different from many attention mechanisms used with DNNs where attention calculation for an item involves other items related to it [16, 22, 27]. (For example, in these scenarios, attention for a location on an image may be determined by both the content at the location and the content at the neighborhood of the location. Attention for a word in a sentence may be determined by both the word and the words before and after it. This is not the case with our scenario.)

We use a regular multi-layer neural network to learn and compute attention weight. It takes as input the feature vector corresponding to a tweet and output a single attention weight for that tweet. Let \mathcal{A} be the attention network and $a_{u,i}$ be the weight for the i -th tweet from user u . The un-normalized attention weight is computed as $\mathcal{A}(\mathbf{z}_{u,i})$. The weights for all the tweets from the same user is normalized to sum to 1. We perform such normalization by softmax, i.e.,

$$a_{u,i} = \frac{e^{\mathcal{A}(\mathbf{z}_{u,i})}}{\sum_j e^{\mathcal{A}(\mathbf{z}_{u,j})}}. \quad (4)$$

We combine feature vectors from all the tweets for the user u by weighted averaging:

$$\mathbf{x}_u = \sum_i a_{u,i} \mathbf{z}_{u,i}, \quad (5)$$

which produces the final feature vector \mathbf{x}_u for the user. A max-margin classifier is applied on this vector to obtain the user’s demographic class label.

3.4 Model Training

We denote by \mathcal{N} the overall feature computation performed by the DNNs (i.e., all the components in Fig. 2 except the max-margin classifier). Hence $\mathbf{x}_u = \mathcal{N}(\mathbf{\Gamma}_u, \theta)$ is the feature vector representing the user that is computed by \mathcal{N} . We use θ as an umbrella term that refers to the collection of parameters used in the DNNs, i.e., the embedding matrix, the filter weights and bias of the CNNs, and the weights of the attention neural network.

Putting DNN features into Eq. 1, the loss of our model is:

$$\mathcal{L} = \frac{1}{N} \sum_u [\delta - y_u(\mathbf{w} \cdot \mathcal{N}(\mathbf{\Gamma}_u, \theta) + b)]_+ + \lambda \|\mathbf{w}\|^2, \quad (6)$$

where $1 \geq \delta > 0$ is a small constant number. Eq. 6 is a generalized version of Eq. 1 while Eq. 1 sets δ to be 1.

We can perform an end-to-end training to simultaneously train θ , \mathbf{w} , and b using Stochastic Gradient Descent (SGD). For the hinge function in Eq. 2, we have the derivative:

$$\frac{d[x]_+}{dx} = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

Following this, the gradients for the parameters are:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{1}{N} \sum_{u \in U_+} -y_u \cdot \mathcal{N}(\mathbf{\Gamma}_u, \theta) + 2\lambda \mathbf{w}, \quad (8)$$

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{1}{N} \sum_{u \in U_+} -y_u, \quad (9)$$

and

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{1}{N} \sum_{u \in U_+} -y_u \left(\mathbf{w} \cdot \frac{\partial \mathcal{N}(\mathbf{\Gamma}_u, \theta)}{\partial \theta} \right), \quad (10)$$

where $U_+ = \{u | \delta - y_u(\mathbf{w} \cdot \mathcal{N}(\mathbf{\Gamma}_u, \theta) + b) > 0\}$ is the set of users, each u satisfying $\delta - y_u(\mathbf{w} \cdot \mathcal{N}(\mathbf{\Gamma}_u, \theta) + b) > 0$. Using these gradients, one can update the parameters θ , \mathbf{w} , and b together in a gradient-descent fashion to minimize the loss in Eq. 6. When updating is performed in a batched fashion, the set U_+ is limited to only those users in the batch. In Eq. 10, $\partial \mathcal{N}(\mathbf{\Gamma}_u, \theta) / \partial \theta$ refers to the gradients of the outputs of the DNN with respect to the parameters of the DNN. The gradient for each individual parameter in the DNN can be derived in a way similar to backpropagation.

Note that if $\mathbf{x}_u = \mathcal{N}(\mathbf{\Gamma}_u, \theta)$ is given, the optimization to minimize Eq. 6 is the same as the objective function of SVM. Hence it can be written as a quadratic optimization and the optimal solution can be found using a solver for quadratic programming. In other words, if we know \mathbf{x}_u , SGD is not needed to get the optimal values for \mathbf{w} and b . It is the training of θ that requires SGD. From this observation, an alternating method for the optimization arises naturally: given

\mathbf{x}_u , one can obtain \mathbf{w} and b by solving a quadratic programming problem. And given \mathbf{w} and b , one can train θ by SGD following Eq. 10 (with values of \mathbf{w} and b fixed as given). We can randomly initialize θ and use it to compute a value for \mathbf{x}_u . The value in turn allows us to optimize \mathbf{w} and b . We can then alternate between the optimization of \mathbf{w} , b and the optimization of θ .

However, there is a problem with this alternating method, namely, over-commitment. Suppose we start with a random choice of θ and compute \mathbf{x}_u accordingly. At this moment, \mathbf{x}_u is very likely a random vector, far from the useful features we want. Optimizing \mathbf{w} and b based on the random vector will not lead to a good model. In the case of end-to-end training using SGD, \mathbf{w} and b are updated slightly at each step. They and θ evolve together. If we optimize using quadratic programming from the beginning, we may commit \mathbf{w} and b to the global optimal with respect to a random vector, which is not desirable. Hence, the alternating method may be applied only when we have learned a sensible feature vector \mathbf{x}_u , i.e., at the late stage of end-to-end SGD training. At that time, we may use the alternating method to speed up training or to make a final optimization for \mathbf{w} and b . In our experiments, we did not see significant improvement using the alternating method. Hence we report only the results from end-to-end training.

4 Experiment Results

We evaluated the proposed approach on a Twitter dataset and the results demonstrated the effectiveness of the approach for predicting 5 different types of demographics including gender, age, ethnicity, income and education.

4.1 Dataset and Evaluation Criteria

Voldova and Bachrach released 5,000 Twitter user profiles annotated via crowdsourcing[23]. We used a subset of these users. Each user has a unique Twitter user ID, and 10 demographic attributes: gender, age, political preference, ethnicity, and so forth. In the study, we considered 5 demographics: age, ethnicity, education, gender and income. Using the Twitter API, we collected the most recent 200 tweets per each user. After removal of users who tweet in languages other than English, and deletion of tweets which simply contain URL, we have a total of 4,551 user profiles and 150 tweets from each user. We deleted punctuations, URLs and emojis from the tweets, and found that 99% tweets are less than 20 words. Therefore, we limited the length of the tweets to 20 words.

Not all of the 4,551 users have 5 attributes annotated. Hence, for different attributes, we have different number of users. For the gender attribute, the ratio of female user is 57.0%. In the task of age prediction, we followed the convention and set 25 as the threshold. People who below the bar were labeled as young, and the others were labeled as old. The ratio of the young people is 58.2%. For education prediction, we let people who own college degrees and above be in high

education group, and those people only with high school diploma be in low education group. The ratio of low education group is 67.5%. To determine ethnicity classes, we choose Caucasian people as one class, and the rest as the other class. In income classification, we set \$35k (annual income) as the threshold. People who are under this point were labeled as low income group, and the rest people were in high income group. The details of the dataset are shown in Table 1.

Table 1: Dataset description.

Attribute	Class description	Users
Age	Below 25 (58%), Above 25	4,224
Education	High School (67%), College degree	4,551
Ethnicity	Caucasian (48%), Others	4,414
Gender	Female (57%), Male	4,351
Income	Below \$35K (65%), Over \$35K	4,551

For deep models, we split the dataset into 3 parts: training set (75%), validation set (5%), and test set (20%). For regular models (logistic regression and SVM), we only have training set (80%) and test set (20%). To evaluate, we adopt 5-fold cross validation to estimate the performance. In each fold, we withheld the ground truth labels in the test data and measured the classification accuracy using *f-score* and *accuracy*. In addition, we measured the *f-score* in each demographic category.

4.2 Models Used in the Experiments

We use 4 baseline models. We list them together with our proposed model:

LOG: This is the logistic regression model using bag of words as the representation of the tweets. It is a simple yet effective model in text categorization [25].

SVC: This is linear SVM using bag of words representation, another effective model for text classification.

CNN: Convolutional neural network for text classification. We use a model that consists of one convolutional layers, each with 256 neurons. Word embedding is used before the convolutional layer. We use an embedding dimension of 50 and the weights from *tweet2vec* [3] to initialize the embedding matrix. Since each tweet is fed into the model individually, we average the output of the convolutional layer for each user. The last layer is a single sigmoid neuron that computes the class probability.

LSTM: LSTM network. We use a bidirectional LSTM network with 256 neurons. Similar to CNN, we average the output of the bidirectional LSTM layer for each user because we use individual tweet as input. Afterwards, a single sigmoid neuron is used to compute the class probability.

NAMIL: This is our model described in Section 3. We call it Neural-Attention Multi-Instance Learning, and hence the acronym NAMIL. We provide the implementation details of the model here. We use the same embedding matrix as in the CNN and the LSTM models. Each convolution layer in NAMIL’s CNN module uses a window size of 3 words and has 40 filters. The attention

network is a two-layer regular neural network. The first layer has 10 neurons and they use relu activation. The second layer consists of a single linear neuron. We also apply L2 regularization on the weights of the neurons in both layers of the attention network. The tradeoff parameter λ for the max-margin classifier is set to 0.01.

4.3 Results

Table 2 lists the performance results of the five models. First of all, comparing the performance measures, we can see that our model consistently outperformed others in inferring most types of demographic information. The LOG model performed the worst, while SVC performed comparable to CNN/LSTM.

Table 2: The average performance and per-group Fscore for multiple methods inferring different types of user demographics.

Data	Metrics	Methods				
		LOG	SVC	CNN	LSTM	NAMIL
Gender	Fscore	0.8226	0.8459	0.8530	0.8500	0.8612
	Accuracy	0.8235	0.8491	0.8548	0.8518	0.8602
	Female-Fscore	0.8334	0.8302	0.8620	0.8368	0.8726
	Male-Fscore	0.8104	0.8375	0.8441	0.8559	0.8626
Age	Fscore	0.6753	0.7331	0.7002	0.7252	0.7426
	Accuracy	0.6905	0.7291	0.7116	0.7314	0.7544
	Young-Fscore	0.6800	0.7322	0.7120	0.7331	0.7550
	Old-Fscore	0.6418	0.7275	0.7111	0.7007	0.7333
Ethnicity	Fscore	0.7892	0.8228	0.8270	0.8408	0.8426
	Accuracy	0.8022	0.8249	0.8326	0.8392	0.8462
	Group1-Fscore	0.7944	0.8322	0.8320	0.8368	0.8526
	Group2-Fscore	0.7723	0.8129	0.8241	0.8359	0.8316
Income	Fscore	0.6639	0.7252	0.7030	0.7398	0.7486
	Accuracy	0.6797	0.7142	0.7108	0.7276	0.7512
	Low-Fscore	0.6868	0.7322	0.7320	0.7368	0.7526
	High-Fscore	0.6104	0.7575	0.7241	0.7259	0.7426
Education	Fscore	0.7135	0.7450	0.7284	0.7730	0.7862
	Accuracy	0.7185	0.7367	0.7390	0.7660	0.7870
	Low-Fscore	0.7244	0.7322	0.7320	0.7868	0.7926
	High-Fscore	0.7004	0.7505	0.7241	0.7559	0.7726

Some types of demographics are easier to predict than others. For gender prediction task, all methods achieve more than 82% accuracy and ours is the best with an accuracy of 86%. The ethnicity prediction task comes second. The accuracies from all the methods are beyond 80%. For education prediction task, the methods do reasonably well, but are not as accurate as the previous two types of demographic predictions. Finally, the hardest tasks are age and income predictions. For age, one of the reasons is that 75% of the users are in 20s. It is difficult to differentiate the words written by the early 20s from the late 20s. Nevertheless, across the attributes of different levels of prediction difficulties,

our model consistently outperformed the others on most of the performance measures.

Max-Margin v.s. Sigmoid Probability To show that the max-margin classifier can improve model performance, we compare NAMIL with a variant in which the max-margin classifier was replaced by a sigmoid neuron that outputs the class probability. The variant model was trained by minimizing the negative log likelihood (NLL) of the data. (We call it the NLL-Variant.) Fig. 3 gives the accuracy performance of these two models for each demographics. It shows that NAMIL significantly outperformed the NLL-Variant, indicating that max-margin classifier is a better choice for the final layer/classifier than a sigmoid neuron. This observation agrees with the results from a previous work [8], which also showed that max-margin classifier can lead to better performance.

Attention v.s. No Attention To check that our attention mechanism improves model performance, we compare the performance of NAMIL with and without attention. For the no-attention case, the features from each tweet are combined through averaging. Fig. 3 shows that using attention significantly improves the model performance as we expected.

Feature Fusion v.s. Instance Fusion In instance fusion, each instance in a bag is given the label of that bag. A model is trained for the individual instances. The class of a bag is determined by merging the predictions for each instance in the bag. NAMIL is feature fusion since it makes the prediction based on the global information from the bag. We conducted experiments to compare NAMIL with a instance fusion model. The instance fusion model uses the same Inception-CNN architecture as the NAMIL for classifying the instances. And it decides the label of the bag by the majority of the predicted classes of its instances. Fig. 3 shows the accuracy of the instance fusion model and NAMIL for each demographics. The performance improvement of feature fusion can be clearly observed.

4.4 Case Study

To gain a better insight into the attention mechanism, we list in Table 3 a few example tweets that are assigned high attention weights. We select some tweets for the prediction of age and some for the prediction of gender. We observe that there is a great level of diversity in style and topics. For age prediction, one example that gets high attention weight is about health savings account, a topic that may be of more concern to older persons than the young ones. Another example for age prediction carries very little semantic information. It is the style of writing that serves as an indicator for the prediction. In fact, we observe that although acronym is quite common across different classes, the tendency to use acronyms (e.g., idk) excessively is often stronger in young people than the older ones.

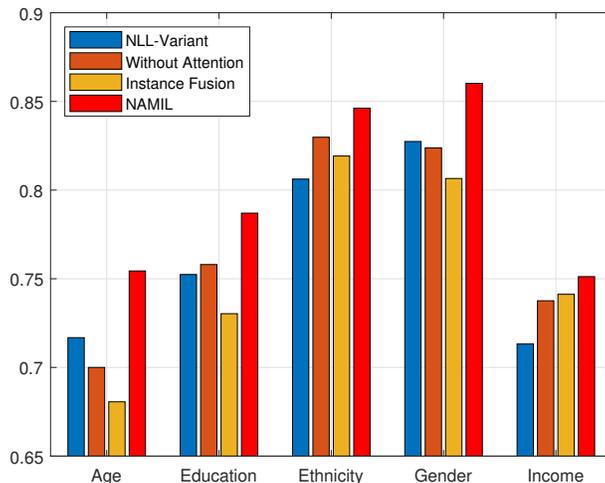


Fig. 3: Comparison of Accuracy between NAMIL and its variants

For the prediction of gender, we observe that a lot of attention is paid to sport-related tweets for the male class. Among the top 100 tweets that get highest attention for this class, a majority is about sports. In the examples in Table 3, we see that there may be a baseball fan who is very excited to go to Fenway Park. For the female class, the tweets that get high attention have quite diverse topics. There are a few tweets about sports too but they are of a much smaller percentage.

Table 3: Example of tweets that were assigned high attention weight for inferring demographic classes.

Trait	Class	Example Tweets
Age	> 25	health save account awesomeness example #2
	> 25	turkey w/ spicy herb mayo pickle onion - grill 'em
	≤ 25	dude like idk . . kinda thirsty
	≤ 25	that's dress rental site great !
Gender	M	worse call nfl history pete carroll never recover
	M	fenway fenway fenway fenway fenway fenway
	F	whatchu know bout english muffin !
	F	don't fight relationship / friendship anymore

5 Conclusion

We have proposed a neural-attention multi-instance learning for predicting user demographic attributes from tweets. The model engages a unique combination of 3 components to deal with the challenges in the particular prediction scenario: an inception-like CNN for feature extraction, an attention mechanism for selecting relevant tweets, and a max-margin classification for regularized model. Our

experiment results show that across different tasks and different performance measures, the proposed model gives the best results in most cases. As future work, we would like to extend our model to predictions of other user attributes and to the multi-instance multi-label scenario where a classifier may be trained to predict multiple traits together.

References

1. Awad, N.F., Krishnan, M.: The personalization privacy paradox: an empirical evaluation of information transparency and the willingness to be profiled online for personalization. *MIS quarterly* **30**(1), 13–28 (2006)
2. Burger, J.D., Henderson, J., Kim, G., Zarrella, G.: Discriminating gender on twitter. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. pp. 1301–1309. Association for Computational Linguistics (2011)
3. Dhingra, B., Zhou, Z., Fitzpatrick, D., Muehl, M., Cohen, W.W.: Tweet2vec: Character-based distributed representations for social media. *arXiv preprint arXiv:1605.03481* (2016)
4. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence* **89**(1-2), 31–71 (1997)
5. Dong, M., Pang, K., Wu, Y., Xue, J.H., Hospedales, T., Ogasawara, T.: Transferring CNNs to multi-instance multi-label classification on small datasets. In: *Image Processing (ICIP), 2017 IEEE International Conference on*. pp. 1332–1336. IEEE (2017)
6. Gui, L., Zhou, Y., Xu, R., He, Y., Lu, Q.: Learning representations from heterogeneous network for sentiment classification of product reviews. *Knowledge-Based Systems* **124**, 34–45 (2017)
7. Hossain, N., Hu, T., Feizi, R., White, A.M., Luo, J., Kautz, H.: Inferring fine-grained details on user activities and home location from social media: Detecting drinking-while-tweeting patterns in communities. *arXiv preprint arXiv:1603.03181* (2016)
8. Jin, J., Fu, K., Zhang, C.: Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems* **15**(5), 1991–2000 (2014)
9. Johnson, R., Zhang, T.: Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058* (2014)
10. Johnson, R., Zhang, T.: Supervised and semi-supervised text categorization using LSTM for region embeddings. *arXiv preprint arXiv:1602.02373* (2016)
11. Kim, Y.: Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014)
12. Liu, J., Inkpen, D.: Estimating user location in social media with stacked denoising auto-encoders. In: *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. pp. 201–210 (2015)
13. Liu, J., Chang, W.C., Wu, Y., Yang, Y.: Deep learning for extreme multi-label text classification. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 115–124. ACM (2017)
14. Mahmud, J., Nichols, J., Drews, C.: Where is this tweet from? inferring home locations of twitter users. *ICWSM* **12**, 511–514 (2012)
15. Mislove, A., Viswanath, B., Gummadi, K.P., Druschel, P.: You are who you know: inferring user profiles in online social networks. In: *Proceedings of the Third ACM International Conference on Web Search and Data Mining*. pp. 251–260 (2010)

16. Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K.: Recurrent models of visual attention. In: *Advances in neural information processing systems*. pp. 2204–2212 (2014)
17. Pennacchiotti, M., Popescu, A.: Democrats, republicans and starbucks aficionados: user classification in twitter. In: *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 430–438 (2011)
18. Rao, D., Yarowsky, D., Shreevats, A., Gupta, M.: Classifying latent user attributes in twitter. In: *Proc. of the 2nd international workshop on Search and mining user-generated contents*. pp. 37–44 (2010)
19. Rao, D., Paul, M.J., Fink, C., Yarowsky, D., Oates, T., Coppersmith, G.: Hierarchical Bayesian models for latent attribute detection in social media. *ICWSM* **11**, 598–601 (2011)
20. Riemer, M., Krasikov, S., Srinivasan, H.: A deep learning and knowledge transfer based architecture for social media user characteristic determination. In: *Proceedings of the third International Workshop on Natural Language Processing for Social Media*. pp. 39–47 (2015)
21. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: *Computer Vision and Pattern Recognition (CVPR)* (2015)
22. Tang, Y., Srivastava, N., Salakhutdinov, R.R.: Learning generative models with visual attention. In: *Advances in Neural Information Processing Systems*. pp. 1808–1816 (2014)
23. Volkova, S., Bachrach, Y.: On predicting sociodemographic traits and emotions from communications in social networks and their implications to online self-disclosure. *Cyberpsychology, Behavior, and Social Networking* **18**(12), 726–736 (2015)
24. Volkova, S., Bachrach, Y.: Inferring perceived demographics from user emotional tone and user-environment emotional contrast. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. vol. 1, pp. 1567–1578 (2016)
25. Wang, S., Manning, C.D.: Baselines and bigrams: Simple, good sentiment and topic classification. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. pp. 90–94. Association for Computational Linguistics (2012)
26. Wang, Y., Huang, M., Zhao, L., et al.: Attention-based LSTM for aspect-level sentiment classification. In: *Proceedings of the 2016 conference on empirical methods in natural language processing*. pp. 606–615 (2016)
27. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: *International Conference on Machine Learning*. pp. 2048–2057 (2015)
28. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pp. 1480–1489 (2016)
29. Zhou, Z.H., Zhang, M.L.: Multi-instance multi-label learning with application to scene classification. In: *Advances in neural information processing systems*. pp. 1609–1616 (2007)

Explainable Artificial Intelligence for Match Analysis in Association Football

Bruno Marques and Dante Barone

Instituto de Informática, Universidade Federal do Rio Grande do Sul, 9500 Bento Gonçalves Ave., Porto Alegre 91501-970, Brazil
{bomarques, barone}@inf.ufrgs.br

Abstract. Explainable Artificial Intelligence (XAI) is an emerging topic that aims to provide, for the users of a learning agent, the means for them to understand the process behind its reasoning with minimal impact on performance or predictive power. One of its possible applications is analyzing the weight that factors, or combinations thereof, have towards the model's outcome, which allows for prioritizing the efforts towards their optimization. This paper applies such an explainable model, with roots in game theory, for the prediction of match outcomes in association football (soccer), able to be further inspected by interested parties such as managers, coaches, and players for the improvement of their tactics and training sessions.

Keywords: Explainable artificial intelligence (XAI) · Shapley Additive Explanations (SHAP) · Machine learning · Association football · Match analysis

1 Introduction

The demand for interpretability in a machine learning model can be justified by the dissonance between its end goals and the mechanisms for evaluation that are available for it [8]. Recent studies on this topic are motivated by concerns regarding the fairness of the resulting criteria applied by these models when they are responsible for significant social consequences [8, 14], in particular given the acknowledgment of several instances where they reproduced the systemic discrimination present in the historical data used as inputs [10, 2]. Another motivation for the development of this area is the introduction of the *right to explanation* for citizens of the European Union with the enactment of the General Data Protection Regulation (GDPR), which guarantees them the right not to be subject to an entirely automated decision with significant (legal or otherwise) consequences without being given an explanation for how it was reached [7].

Meanwhile, there has been a significant increase in the availability of event data from association football matches, which opened a vast array of research opportunities [15]. However, although statistical studies aimed towards finding significant factories behind match outcomes have found success [21, 20], the use of machine learning for building predictive models has been mostly restricted

to search for inefficiencies in the betting market [5, 19, 4] and as an exercise for testing specific techniques [24, 18, 1]. One possible reason is the limited knowledge that can be extracted since better accuracy requires more complex models of very limited comprehension [9].

This study combines the use of black-box machine learning techniques with local surrogate models, which are an interpretability mechanism designed to explain the weights of factors on the outcome of specific test cases [17]. The benefits are twofold: they provide a high-level overview which allows for an easy replacement of the underlying predictive model if desirable [16], and they allow the study of specific matches, such as recent defeats from the team analyzing them or those from the upcoming adversaries and top performers. For that, we build a Shapley Additive Explanations (SHAP) model, which can efficiently show the contribution made by each specific feature on the outcome of either one particular input or any subset of the data [11]. This allows for easier analysis of the trends behind match outcomes with any criteria by a non-technical interested party.

2 Explainable Artificial Intelligence

Explainable Artificial Intelligence (XAI) can be described as an agent with the ability to present, on a human-friendly format, the underlying causes behind the decision-making process of itself or an external agent it acts as a surrogate for. This makes it a *human-agent interaction* problem, which is on the intersection between artificial intelligence, social science, and human-computer interaction [12].

As an incipient topic, XAI suffers from a series of unaddressed issues: the lack of a clear definition for interpretability is a major concern, with specific aims and taxonomies remaining an open issue [6]. Also, despite the acknowledgment of the importance of disciplines such as human-computer interaction and psychology on the matter, the development of the models themselves remains dominated by computer scientists who prioritize raw output generation without considering the issue of how third parties, many of them non-technical, will consume them and extract their knowledge. Since many of the concerns that motivated the emergence of XAI as a discipline are known issues in the social sciences, they accumulated a significant body of work regarding the treatment of explanations by humans that was presented as a possible inspiration [12].

2.1 Model-agnostic methods

As an alternative to the limited array of inherently explainable models, whose necessary simplicity can have a significant impact on accuracy or performance, mechanisms for the interpretation of predictive systems that do not make assumptions about their functioning, called model-agnostic methods, have been developed. The main advantage brought by this visualization paradigm is flexibility; since they act as a high-level abstraction for any kind of predictive model,

they present a consistent interface for end users while the underlying black box can be easily replaced whenever circumstances make it desirable or necessary to [16].

The explanations brought by model-agnostic methods can be categorized into two broad groups: local explanations, which focus on a small region of the feature space centered around a particular input, with the goal of explaining that specific input, and global explanations, which summarize the general behavior of the predictive model. The latter is harder to achieve since it requires a much more robust approximation method, and is farther from the more common goal of explaining individual predictions.

2.2 Shapley Additive Explanations

Among the most popular approaches for prediction model interpretability in recent literature are the additive feature attribution methods, which generate explanation models for individual predictions that consist of linear functions of a simplified mapping of their features [11]. This includes the Local Interpretable Model-agnostic Explanation (LIME), which generates a surrogate model trained with mutations of the input of interest, weighted by their distance to it [17], and estimations of Shapley values, a game theory concept designed to find the individual contribution from each player of a cooperative game towards a payout outcome [22].

The latter option, however, suffers from an exponential computational complexity, because it needs to generate 2^k values for k features. Some approximations were proposed, such as the Shapley sampling values model, which applies a heuristic based on Monte Carlo sampling and other inputs used for training the original model [25].

The Shapley Additive Explanations (SHAP) framework was designed as a combination of both methods, by applying the LIME algorithm with parameters designed to approximate the Shapley values of the features, thus benefiting from the former's performance while getting closer to the latter's desirable mathematical properties [11].

3 Experiments

In this section, we investigate the applicability of SHAP values as a mechanism to identify patterns behind the contributions made by distinct factors to the outcome of association football matches. This experiment compares 10 different supervised learning models, calculates the SHAP values from the presented data and discusses some of the trends evidenced by them.

3.1 Data preprocessing

The data set used is provided by data provider StatsBomb in the form of a GitHub repository, which is publicly available for non-commercial use [23]. It

includes comprehensive event data on 170 matches across three tournaments: the 2018 edition of the FIFA World Cup, the 2018 season of the National Women’s Soccer League, in the United States, and the 2018–19 season of the FA Women’s Super League, in England. It’s the first open data set of association football matches with such granularity, which lowers the entry barrier for any future research in the field and makes results much easier to reproduce.

Table 1. Matches included in the StatsBomb database

Championship	Region	Gender	Season	# of matches
FIFA World Cup	Worldwide	Male	2018	64
National Women’s Soccer League	United States	Female	2018	34
FA Women’s Super League	England	Female	2018-19	72

From this raw data, every match yielded two rows, one per team, with a set of features (presented in tables 2 and 3) to be evaluated and the outcome of that match (win, draw or loss). The *home.advantage* feature was set to “No” for all World Cup matches, as they were played on neutral ground.

Table 2. Quantitative features

Variable name	Minimum	Median	Mean	Maximum	St. dev.
shots	0	12.5	13.659	43	6.731
expulsions	0	0	0.024	1	0.152
corners	0	4	4.471	20	2.922
fouls	2	11	11.756	26	4.741
passes	224	456	468.150	1157	130.937
passes.success.rate	0.533	0.915	0.890	0.989	0.083
avg.recovery.time	8.884	29.824	31.332	66.876	10.246
avg.player.possession	0.627	1.342	1.348	2.298	0.246
avg.team.possession	0.404	0.503	0.521	0.757	0.067

From this input, 8 different classification models were built based on 6 different approaches: support vector machine, naive Bayes (both Gaussian and complement variants), random forest (one with the traditional algorithm and another with Extra-Trees), AdaBoost.SAMME, logistic regression and gradient boosting. These models were then trained, following a 70-30 split between training and test sets, and evaluated according to five metrics: rank probability score (RPS), Cohen’s kappa, logarithmic loss, accuracy, and macro-averaged F1 score.

Table 3. Qualitative features

Variable name	# of categories	Distribution
home.advantage	2	Yes: 106; No: 234
gender	2	Male: 128; Female: 212

The RPS was chosen as the primary metric because it presents desirable properties for evaluating probability forecasts where the order of target variables matters [13]; in our case, both wins and losses are closer from draws than from each other [3].

Table 4. Performance of the evaluated classifiers

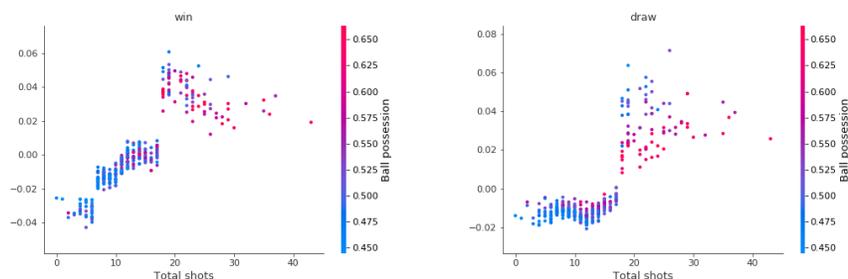
Classifier	RPS	Kappa	Log loss	F1	Accuracy
RF (original)	0.166263	0.442026	0.827025	0.559661	0.661765
Gradient boosting	0.170575	0.390899	0.864263	0.537901	0.632353
RF (Extra-Trees)	0.181417	0.376376	0.886403	0.461806	0.632353
Logistic regression	0.193022	0.415054	0.901471	0.518864	0.647059
Support vector machine	0.211455	0.245796	0.948650	0.441383	0.544118
AdaBoost-SAMME	0.240604	0.434707	1.075846	0.569092	0.647059
Naive Bayes (Gaussian)	0.277561	0.335745	3.304775	0.448029	0.602941
Naive Bayes (complement)	0.388902	0.180424	8.213246	0.376344	0.514706

3.2 Analysis

The analysis of SHAP values uncovered some interesting patterns that warrant further investigation.

Figure 1 shows that the model established a threshold of nearly 20 shots for a positive impact on avoiding defeat. Also, somewhat counterintuitively, the probability for achieving a draw decreases when there are larger disparities in ball possession.

Meanwhile, figure 2 presents a very distinct pattern regarding the number of passes performed by a team, the resulting number of shots and the ultimate match outcome. Exchanging less passes overall has a significantly higher impact on the winning probabilities, even though these passes present lower success rates; this indicates that the model favors teams employing the long ball strategy. On higher amounts, a local minimum for the SHAP values in a win scenario appears, while draw chances show a consistent upward trend. One possible interpretation is that this affirms the common wisdom about the straightforwardness

Fig. 1. SHAP values per number of shots

of direct play in contrast to high-possession tactics such as the *tiki-taka*, which has a "sweet spot" for optimal performance and can stall the game when overdone.

On the other hand, according to figure 3, a high rate of success for passes (somewhere above 90%) has a significant impact on positive outcomes, regardless of factors such as ball possession and defensive pressure, but anything lower makes little, if any, difference. This highlights the popular dichotomy between the passing game and long ball; one is not inherently superior to the other, but it helps to make a choice and commit to it.

The analysis of the SHAP patterns for ball possession in figure 4 reveal a sharply declining trend for both wins and draws below the 50% mark; afterward, the impact on drawing chances, although positive, presents a very high variance, while the influence on winning past the 50-55% range experiences a significant rise before peaking at 60%. This points towards a target rate for possession after which no significant gains are achievable.

Finally, defensive pressure, represented in figure 5 by the average ball recovery time, manifests as a positive factor on victory chances when reduced to under 30 seconds, which increases the lower it gets.

4 Conclusions

As a very recent area of study, XAI presents a vast array of unexplored or underexplored applications, such as sports analytics. By combining the predictive power of complex black box models with the flexibility and consistency of model-agnostic abstractions, it's a powerful mechanism for presenting knowledge from an efficient learner to any parties interested in finding ways to guide towards the desired outcome. The imminent report from DARPA about the first stage of the program of the same name is very promising in this regard since it reveals preliminary results from a large body of research.

Since the field remains far from maturity, several issues remain on the initial stages of research and have not been addressed properly yet. The lack of

Fig. 2. SHAP values per number of passes

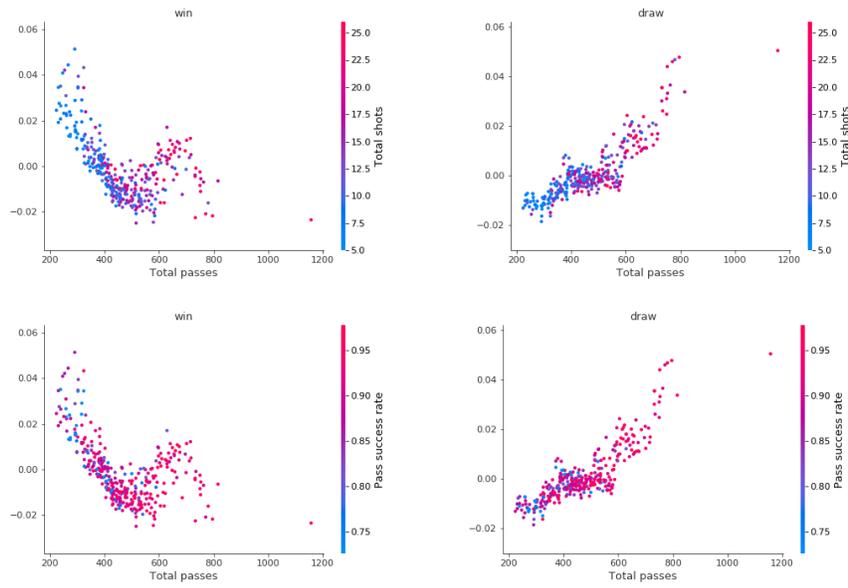


Fig. 3. SHAP values per success rate of passes

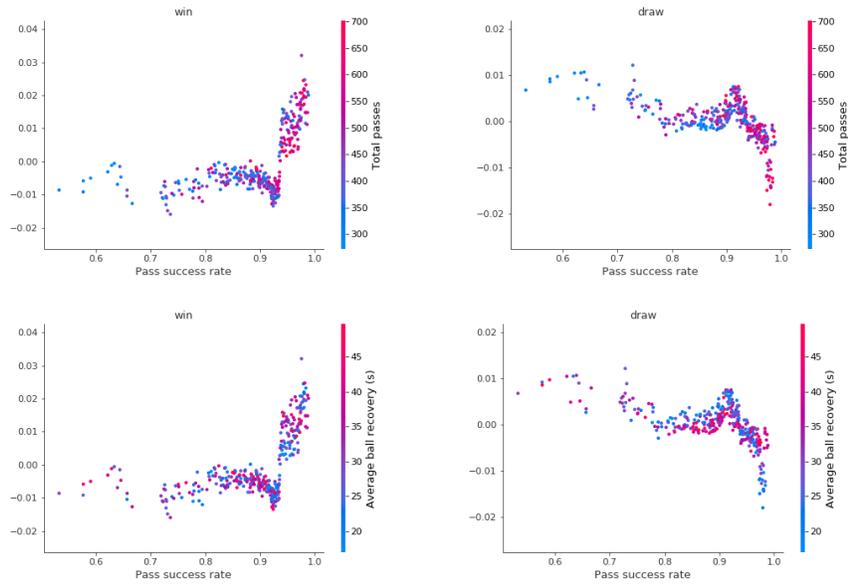
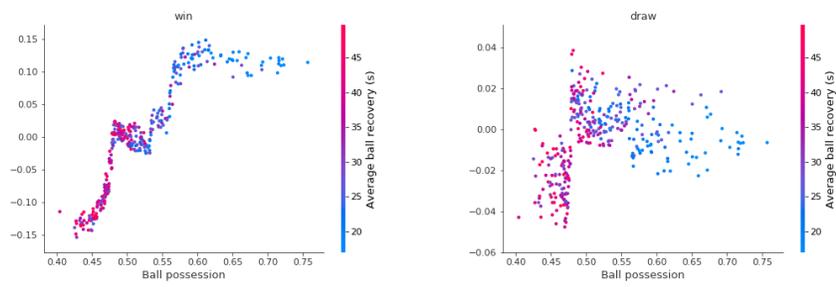
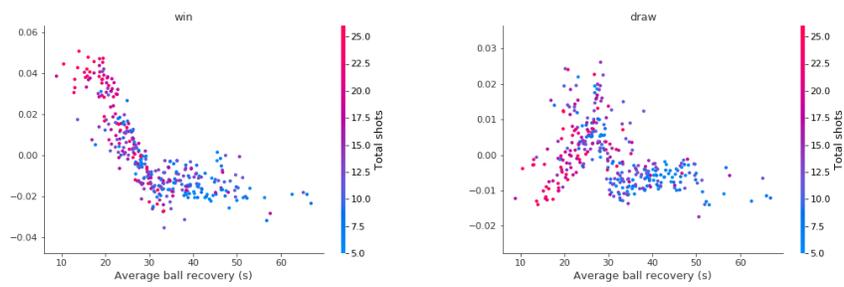


Fig. 4. SHAP values per ball possession**Fig. 5.** SHAP values per defensive reaction time

proper metrics for evaluating interpretability is one of them, and models are developed mostly in an *ad hoc* fashion, despite its interdisciplinary nature and the acknowledgment of the importance of fields such as psychology and human-computer interaction in addressing their ultimate goals.

The temporal aspect of the sport was not explored by the generated models, which only used aggregate match-wise metrics, despite the raw event data allowing for that. One interesting approach could be to calculate a set of metrics for player abilities such as passing, acceleration, and fitness, similar to what is done by video game franchises such as *FIFA* and *Football Manager*. Then, similar predictive models could be applied to all sorts of match events for posterior interpretation.

References

- [1] Baio, G., Blangiardo, M.: Bayesian hierarchical model for the prediction of football results **37**(2), 253–264, <http://www.tandfonline.com/doi/abs/10.1080/02664760802684177>
- [2] Caliskan, A., Bryson, J.J., Narayanan, A.: Semantics derived automatically from language corpora contain human-like biases **356**(6334), 183–186. <https://doi.org/10.1126/science.aal4230>, <http://arxiv.org/abs/1608.07187>
- [3] Constantinou, A.C., Fenton, N.E.: Solving the Problem of Inadequate Scoring Rules for Assessing Probabilistic Football Forecast Models **8**(1). <https://doi.org/10.1515/1559-0410.1418>, <https://www.degruyter.com/view/j/jqas.2012.8.issue-1/1559-0410.1418/1559-0410.1418.xml>
- [4] Constantinou, A.C., Fenton, N.E., Neil, M.: Profiting from an inefficient association football gambling market: Prediction, risk and uncertainty using Bayesian networks **50**, 60–86. <https://doi.org/10.1016/j.knosys.2013.05.008>, <http://www.sciencedirect.com/science/article/pii/S095070511300169X>
- [5] Dixon, M.J., Coles, S.G.: Modelling Association Football Scores and Inefficiencies in the Football Betting Market **46**(2), 265–280. <https://doi.org/10.1111/1467-9876.00065>, <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9876.00065>
- [6] Doshi-Velez, F., Kim, B.: Towards A Rigorous Science of Interpretable Machine Learning <http://arxiv.org/abs/1702.08608>
- [7] Goodman, B., Flaxman, S.: European Union regulations on algorithmic decision-making and a "right to explanation" **38**(3), 50. <https://doi.org/10.1609/aimag.v38i3.2741>, <http://arxiv.org/abs/1606.08813>
- [8] Lipton, Z.C.: The Mythos of Model Interpretability <http://arxiv.org/abs/1606.03490>
- [9] Lou, Y., Caruana, R., Gehrke, J.: Intelligible Models for Classification and Regression. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 150–158. KDD '12, ACM. <https://doi.org/10.1145/2339530.2339556>, <http://doi.acm.org/10.1145/2339530.2339556>
- [10] Lowry, S., Macpherson, G.: A blot on the profession **296**(6623), 657–658, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2545288/>
- [11] Lundberg, S., Lee, S.I.: A Unified Approach to Interpreting Model Predictions <http://arxiv.org/abs/1705.07874>

- [12] Miller, T.: Explanation in Artificial Intelligence: Insights from the Social Sciences <http://arxiv.org/abs/1706.07269>
- [13] Murphy, A.H.: The ranked probability score and the probability score: A comparison **98**(12), 917–924. [https://doi.org/10.1175/1520-0493\(1970\)098;0917:TRPSAT;2.3.CO;2](https://doi.org/10.1175/1520-0493(1970)098;0917:TRPSAT;2.3.CO;2)
- [14] Pedreshi, D., Ruggieri, S., Turini, F.: Discrimination-aware Data Mining. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 560–568. KDD '08, ACM. <https://doi.org/10.1145/1401890.1401959>, <http://doi.acm.org/10.1145/1401890.1401959>
- [15] Rein, R., Memmert, D.: Big data and tactical analysis in elite soccer: Future challenges and opportunities for sports science **5**(1), 1410. <https://doi.org/10.1186/s40064-016-3108-2>, <http://springerplus.springeropen.com/articles/10.1186/s40064-016-3108-2>
- [16] Ribeiro, M.T., Singh, S., Guestrin, C.: Model-Agnostic Interpretability of Machine Learning <http://arxiv.org/abs/1606.05386>
- [17] Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": Explaining the Predictions of Any Classifier <http://arxiv.org/abs/1602.04938>
- [18] Rotshtein, A.P., Posner, M., Rakityanskaya, A.B.: Football Predictions Based on a Fuzzy Model with Genetic and Neural Tuning **41**(4), 619–630. <https://doi.org/10.1007/s10559-005-0098-4>, <http://link.springer.com/10.1007/s10559-005-0098-4>
- [19] Rue, H., Salvesen, O.: Prediction and Retrospective Analysis of Soccer Matches in a League **49**(3), 399–418. <https://doi.org/10.1111/1467-9884.00243>, <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9884.00243>
- [20] Sarmiento, H., Clemente, F.M., Araújo, D., Davids, K., McRobert, A., Figueiredo, A.: What Performance Analysts Need to Know About Research Trends in Association Football (2012–2016): A Systematic Review **48**(4), 799–836. <https://doi.org/10.1007/s40279-017-0836-6>, <https://doi.org/10.1007/s40279-017-0836-6>
- [21] Sarmiento, H., Marcelino, R., Anguera, M.T., Campaniço, J., Matos, N., Leitão, J.C.: Match analysis in football: A systematic review **32**(20), 1831–1843. <https://doi.org/10.1080/02640414.2014.898852>, <http://shapeamerica.tandfonline.com/doi/abs/10.1080/02640414.2014.898852>
- [22] Shapley, L.S.: A Value for N-Person Games. In: Kuhn, H.W., Tucker, A.W. (eds.) Contributions to the Theory of Games, Annals of Mathematics Studies, vol. II, pp. 307–318. Princeton University Press
- [23] StatsBomb: StatsBomb Open Data repository, <https://github.com/statsbomb/open-data>
- [24] Tsakonias, A., Dounias, G., Shtovba, S., Vivdyuk, V.: Soft Computing-Based Result Prediction of Football Games p. 9
- [25] Štrumbelj, E., Kononenko, I.: Explaining prediction models and individual predictions with feature contributions **41**(3), 647–665. <https://doi.org/10.1007/s10115-013-0679-x>, <http://link.springer.com/10.1007/s10115-013-0679-x>

Using Machine Learning Algorithms to Predict the Likelihood of Recurrent Falls in Older Adults

Leeanne Lindsay^{1a}, Sonya Coleman^{2b}, Brian Taylor^{1c},
Dermot Kerr^{2d} Anne Moorhead^{3e}

¹ Institute for Research in Social Sciences, Ulster University, N. Ireland, UK

² Intelligent Systems Research Centre, School of Computing, Engineering and Intelligent Systems, Ulster University, Londonderry, UK

³School of Communication and Media, Institute for Nursing and Health Research, Ulster University, N. Ireland, UK.

^alindsay-l@ulster.ac.uk, ^bsa.coleman@ulster.ac.uk, ^cbj.taylor@ulster.ac.uk,
^dd.kerr@ulster.ac.uk, ^ea.moorhead@ulster.ac.uk

Abstract. It is well known that older adults are more prone to falls due to their cognitive decline with the ageing process. Falls and the consequences of falls can have a huge impact on older adults and their families. Identification and communication of such fall risks must involve the individual, their families and health professionals. This paper explores the use of machine learning algorithms to predict if multiple risk factors including previous falling can lead to recurrent falls. The approach is based on the data collected by The Irish Longitudinal Study on Ageing. Initial results shows that certain risk factors do in fact contribute to older adults falling such as overall health, mental and long-term health as well as black-outs and joint replacements.

Keywords: Falls, Machine Learning, Risk Factors.

1 Introduction

Falling can happen to any of us. However people of 65 years old and over are defined as a vulnerable group of adults prone to falls [1]. Falling can be overlooked but can cause serious injury and unfortunately there are psychological and social consequences that follow, which may have just as high an impact as the fall itself [2]. According to the National Health Service (NHS) in the UK, around one in three adults whom are over the age of 65 and live at home will fall at least once each year [3]. It is estimated that in the United Kingdom by 2030, every one in five people will be 65 years or older [4]. For an older adult, falling can lead to distress, significant pain and injury, loss of confidence and loss of independence [5]. The risk of falling not only affects the individual but may also have an impact on the family, caregivers at home and health and social care professionals. Decisions made about an individual, depending on how a fall has affected them, should ideally be a shared decision about the future of the individual affected to ensure that the best, safe care package is put in place for the older adult [6].

Falling not only impacts the health and quality of life of the patient; health and social care costs related to falls are estimated to cost the NHS in England £2.3 bn per year [5].

There are many causes of falls such as medication, poor eyesight, poor balance, low blood pressure, blackouts due to dizziness, lack of exercise or inactivity, cognitive impairment such as dementia and other medical and mental health issues [3]. There are many potential consequences for older adults if they fall. Falling is known as a ‘vicious circle’; once you have one fall you are more likely to have recurrent falls [8]. Figure 1, illustrates the psychosocial process by which one fall can lead to further falls due to a person’s loss of confidence and inactivity due to the fear of falling.

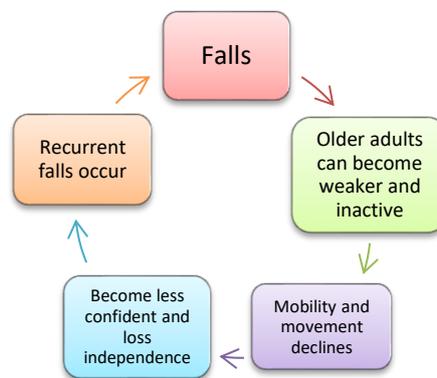


Figure 1: ‘Vicious Circle’ of Falls

Within the health and social care sector, machine learning is beginning to be used to predict adverse outcomes. Big data analytics is permitting the identification of hidden patterns and correlations within large datasets by mining large volumes of data. The smaller, more meaningful pieces of information extracted might be used to assist with the decision-making process for health and social care professionals, by helping to improve care and reduce costs [9].

This paper presents the use of machine learning algorithms to predict the likelihood of falls in older adults and is organized as follows: in Section 2 the dataset used is presented; the methodology and results for this study are described in Section 3; and the paper is concluded in Section 4 with future work proposed.

2 Data Collection

The Irish Longitudinal Study on Ageing (TILDA) dataset was used for this study, comprising data on community-dwelling participants. The dataset collected information from 8000 adults over the age of fifty in the Republic of Ireland (RoI). TILDA collected data in waves every two years. There are three waves in total, Wave One was collected between 2009 to 2011; Wave Two was collected between 2012 and 2013; and Wave Three was collected between 2015 and 2016. The two waves examined in this study

were based on a period of time from 2009 to 2013 which includes Wave One and Wave Two. Wave One consists of a random sample of participants who become part of the framework for the study, Wave Two consisted of a questionnaire and a face to face computer assisted interview using the same random sample of people from Wave One. TILDA collected information from participants about their health and care, mobility, previous education, employment and housing.

Previously, the TILDA dataset has been used to study whether depression correlated with falls in older adults, with the risk of unexplained falls significantly increasing among those with depression [10]. One published study has been identified that examined the performance of machine learning models on the TILDA dataset [11]. In predicting the risk of falls against health risk factors, the Classification via Regression method performed best of those studied, although accuracy between the different models did not vary greatly [11].

3 Methodology

For this study, Waikato Environment for Knowledge Analysis (WEKA) was used to enable a range of machine learning algorithms to be readily applied to the TILDA dataset. The machine learning algorithms were trained to predict if older adults were likely to fall in the next year depending on the number of falls they had previously. The input data for each of the models was taken from the TILDA dataset and included participant characteristics such as Overall Health Description, Emotional Mental Health, Long-Term Health Issues, Previous Blackout or Fainting, Joint Replacements and the Number of Times Fallen. Participants in the study were asked “Have you fallen in the last 12 months?”; this became the target variable to predict if they would have a fall in the future based on data collated prior to that 12 month period. The number of times fallen ranged from zero falls up a maximum of ten falls. For this study, all non-fallers were removed and only those individuals recording one fall to ten falls were used. This allowed for the model to predict based on current fallers having recurrent falls.

4 Results

The dataset used was split into a training and testing set including 90% training samples and 10% testing. Ten-fold cross validation was used to ensure model accuracy throughout. The results from different machine learning algorithms are presented in Table 1, where the models with the highest performance are highlighted in orange for both Wave One and Wave Two. Machine learning algorithms were used with the health risk factors as inputs to predict the likelihood of recurrent falls based on if they had fallen before. Using these health risk factors are important to help aid health and social care professionals to mitigate the risk factors that older adults have to live with in their everyday lives.

A number of different classification algorithms were used including the following. Sequential Minimal Optimization (SMO) which trains a Support Vector Classifier and

transforms nominal attributes into binary ones. Another approach, PART, builds a partial decision tree in each iteration and makes the best leaf into a rule. The Random Forest classifier constructs a forest of random trees used for classification and regression. Decision Tree or sometimes known as J48 in WEKA is used to generate a pruned or unpruned decision tree. Bayes Net is a statistical model that represents a set of variables and their dependencies. The Logistic model is used for building and using a multinomial logistic regression model with a ridge estimator. Multilayer Perceptron uses backpropagation to learn a multi-layer perceptron to classify instances. Simple Logistic builds linear logistic regression models and Classification-via-Regression is a class for doing classification using regression methods. The class is binarized and one regression model is built for each class value. Each algorithm used in this study was tested to determine which approach provided the best accuracy result based on the input data.

Algorithms such as the Decision Trees, Logistic Model, and Sequential Minimal Optimization have all predicted a fall with an accuracy of 90% in Wave One and 77% in Wave Two. Both waves provided different accuracy results based on the data collected in the two different years, due to the potential difference in numbers of people fallen in the first wave and the second. In Wave One there were 139 falls compared to Wave Two there were 230 extra falls with a total of 369 falls which may have impacted on the difference in the algorithmic results. However, the machine learning algorithms that provided the best result were the same in both waves. These results are promising and demonstrate that machine learning can be used with real health and care data in order to predict the risk of falling.

Table 1 Wave 1 machine learning algorithm results

a Classifier	Wave One	Wave Two
	Correctly Classified %	Correctly Classified %
Naïve Bayes	87.90	75.79
SMO	90.02	77.26
PART	89.73	75.91
Random Forest	88.44	74.31
Decision Tree	90.02	77.26
Bayes Net	89.37	76.03
Logistic	90.02	77.14
Multilayer Perceptron	88.15	75.35
Simple Logistic	90.02	77.26
Classification via Regression	89.80	77.26

5 Conclusion

This study has examined the relationship between health and social care risk factors for older people who had a fall to explore if having a previous fall correlated with having further falls. Data from the longitudinal TILDA study were used with several machine learning algorithms, and proved useful for the task. The Decision Tree, Simple Logistic and Sequential Minimal Optimization models performed best in both waves of data. The other models tested gave slightly less accurate results. The results using Wave One data have proven to be more accurate than those using Wave Two data, and we propose this is due to the variation in the number of people that have fallen in the first wave than compared to the second wave. For future research, the English Longitudinal Study of Ageing (ELSA) will be used for similar risk analyses to identify correlations between risk factors and the health and well-being of older adults across different jurisdictions. This study successfully used data collected from health and care risk studies with machine learning algorithms to predict recurrent falls in older adults.

References

- [1] Johansson, I., Bachrach-Lindstrom, M., Struksnes, S. and Hedelin, B. (2009) "Balancing integrity vs. risk of falling - nurses' experiences of caring for elderly people with dementia in nursing homes", 14, pp. 61-73. doi: 10.1177/1744987107086423.
- [2] Salva, A., Bolibar, I., Pera, G. and Arias, C. (2004) "Incidence and consequences of falls among elderly people living in the community", 122(5), pp. 172-176. doi: doi.org/10.1016/S0025-7753(04)74184-6.
- [3] NHS Confederation (2012, Issue 234) "Falls prevention New approaches to integrated falls prevention services", [Online]. Available: https://www.nhsconfed.org/~media/Confederation/Files/Publications/Documents/Falls_prevention_briefing_final_for_website_30_April.pdf. [Accessed: 09-Feb-18].
- [4] Trinity College Dublin (2018) The Irish Longitudinal Study on Ageing (TILDA). Available at: <https://tilda.tcd.ie/> (Accessed: December 2018.)
- [5] NICE (2013) Falls in older people: assessing risk and prevention - NICE Guideline, . doi: nice.org.uk/guidance/cg161.
- [6] Godolphin, W. (2009) Shared Decision Making. *Healthcare Quarterly*, 12, pp. e186 - e190. doi: <http://www.longwoods.com/content/20947>.
- [7] Callis, N. (2016) "Falls prevention: Identification of predictive fall risk factors," *Applied Nursing Research* 29, pp. 53-58.
- [8] Costa, B., Rutjes, A., Mendy, A., Freund-Heritage, R. and Vieira, E. (July 17, 2012) "Can Falls Risk Prediction Tools Correctly Identify Fall-Prone Elderly Rehabilitation Inpatients? A Systematic Review and Meta-Analysis," doi: doi.org/10.1371/journal.pone.0041061.
- [9] Koh, CH., Tan, G. (2005) "Data Mining Applications in Healthcare," *IOSR Journal of Engineering (IOSRJEN)*, V19(2), pp. 64-72.
- [10] Briggs, R., Kennelly, P.S. and Kenny, R.A. (2017) "Does baseline depression increase the risk of unexplained and accidental falls in a cohort of community-dwelling older people?," *Int J Geriatr Psychiatry*. 33e205-e211.
- [11] Lindsay, L., Coleman, S., Taylor, B., Kerr, D. and Moorhead, A. (2019) "Classification of Health Risk Factors to Predict the Risk of Falling in Older Adults," [Submitted to: World Academy of Science, Engineering and Technology Conference 2019]

Tree Based Clustering On Large, High Dimensional Datasets^{*}

Lee A. Carraher, Sayantan Dey, and Philip A. Wilsey^[0000-0002-6562-8646]

Dept of EECS, University of Cincinnati, Cincinnati, OH 45221-0030 USA
leecarraher@gmail.com, deysn@mail.uc.edu, wilseypa@gmail.com

Abstract. Clustering continues to be an important tool for data engineering and analysis. While advances in deep learning tend to be at the forefront of machine learning, it is only useful for the supervised classification of data sets. Clustering is an essential tool for problems where labeling data sets is either too labor intensive or where there is no agreed upon ground truth. The well studied k -means problem partitions groups of similar vectors into k clusters by iteratively updating the cluster assignment such that it minimizes the within cluster sum of squares metric. Unfortunately k -means can become prohibitive for very large high dimensional data sets as iterative methods often rely on random access to, or multiple passes over, the data set — a requirement that is not often possible for large and potentially unbounded data sets. In this work we explore an randomized, approximate method for clustering called *Tree-Walk Random Projection Clustering* (TWRP) that is a fast, memory efficient method for finding cluster embedding in high dimensional spaces. TWRP combines random projection with a tree based partitioner to achieve a clustering method that forgoes storing the exhaustive representation of all vectors in the data space and instead performs a bounded search over the implied cluster bifurcation tree represented as approximate vector and count values. The TWRP algorithm is described and experimentally evaluated for scalability and accuracy in the presence of noise against several other well-known algorithms.

Keywords: Clustering · machine learning · dimensional reduction · locality sensitive hashing.

1 Introduction

The expanding needs for analysis on large data sets has increased as the amount and availability of data continues to grow. Concepts such as the Internet of Things (IoT), social media, digitized medical records, and the aggregation of complex high volume scientific measurements further exacerbated this need. The size and format of this data makes manual analysis infeasible and has motivated the drive for automated methods such as data clustering.

^{*} Support for this work was provided in part by the National Science Foundation under grant ACI-1440420.

Meanwhile big data solutions often find their greatest success on large fast moving data sets. By virtue of the sheer volume of these datasets, supervised labeling is often not possible. Unsupervised learning fills this gap by sacrificing a desired optimization based on a set of ground truths for a solution that, in the case of clustering, tries to optimize the co-similarity of objects in a partition or dissimilarity of object between partitions. Clustering is the process of partitioning data into grouping of related items. It is one of the most fundamental modes of learning and understanding data [21]. Clustering is an intuitive, data analysis method that can provide clear insights into the underlying structure and trends of a dataset. Given its interpretability at high applicability, clustering has been used in a wide variety of disciplines ranging from medical diagnostics and epidemiology to financial prediction and credit fraud detection. As the number of sources and volume of these types of datasets grow deeper insights are desired.

Among the most commonly used clustering algorithms, k -means has been proven as one of the most popular choices that delivers acceptable results in reasonable time [21]. k -means has proven to be statistically efficient and easy to implement. While k -means is widely used for clustering streaming data, it has performance issues when it comes to robustness with noise, parallelism, and working with very large, high-dimensional data sets. In particular, k -Means (and other conventional techniques) for data clustering do not parallelize or scale well with the increasing dimensionality of data.

This paper introduces and evaluates an approximate method for clustering called *Tree-Walk Random Projection Clustering (TWRP)* that is a fast, memory efficient method for finding clusters in high dimensional spaces. TWRP is a tree based clustering method that forgoes storing the exhaustive representation of all vectors of the data space and instead performs a bounded search over the implied cluster bifurcation tree represented as approximate vector and count values. Big data clustering algorithms are not new; in fact many classic algorithms are reasonably well suited to operate on big data with only minor pre-clustering tweaks [26]. Although successful, most methods optimize over the entire data embedding, while TWRP finds not only dense regions in the full embedding, but also allows for the identification of dense low-rank embeddings.

The remainder of this paper is organized as follows: Section 2 provides some background information. Section 3 surveys related work to develop solutions for high-performance big-data clustering. Section 4 describes the Tree Walk Random Projection (TWRP) cluster method and the original RPHash algorithm [8] (on which TWRP is based). Section 5 contains experimental results of our proposed solution against other common clustering methods. Finally, Section 6 gives a summary of our results and findings.

2 Preliminaries

An important tool for overcoming the Curse of Dimensionality *COD* is *dimensional reduction*. In the case of very large datasets, more robust data aware dimensional reduction techniques such as t-SNE and PCA begin to dominate

the computational complexity. For ill-posed problems such as clustering, the optimal subspace embedding that is based on minimizing either the L_2 -norm or Kullback-Leibler divergence is somewhat overkill. Instead, approximate dimensional reduction technique such as Random Projection using the method of Database Friendly Projection by Achlioptas [1] is sufficient.

The JL-Lemma [32] states that the error bound for low dimensional embeddings, is exponentially proportional to the number of objects in the dataset. For large datasets, such as those we are interested in, this bound is still somewhat prohibitive, calling for subspace embeddings on the order of thousands. Thus for $1 - \epsilon = .90$, and $n > 10^9$, the reduced dimension D is bounded below by $D > \Omega(\frac{\log(n)}{\epsilon^2})$.

Locality Sensitive Hash (LSH) functions are employed as probabilistic representation of vector locality to improve the prohibitive subspace embedding dimensionality requirement of the JL-Lemma. An LSH function is any hash function with the property that hashed records with more similar components are more likely to be hashed to the same bucket than records with more fewer similarities. Formally:

Definition 1 (Locality Sensitive Hash Function [11]). *let $\mathbb{H} = \{h : S \rightarrow U\}$ is (r_1, r_2, p_1, p_2) -sensitive if for any $u, v \in S$, where h is the hash function belonging to the hash family \mathbb{H} that maps from the element set S to U and d is the distance metric. Thus, an LSH function operates so that:*

$$\begin{aligned} &\text{if } d(u, v) \leq r_1 \text{ then } Pr_{\mathbb{H}}[h(u) = h(v)] \geq p_1, \text{ and} \\ &\text{if } d(u, v) > r_2 \text{ then } Pr_{\mathbb{H}}[h(u) = h(v)] \leq p_2. \end{aligned}$$

While LSH functions are interesting approaches to quickly compute vector locality, they tend to have some difficulties separating distinct communities of vector data, especially when the vector data is not uniformly distributed throughout the subspace. While considerable work has gone into finding better and near optimal [4] functions for optimizing the signal to noise ratio for locality sensitive hash functions, the best solutions often require multiple passes over the data to build data aware functions.

To switch from the continuous spaces of random projections, discrete space partitioning lattices are also considered. The data space must be partitioned as evenly as possible for an optimal implementation of the clustering. Also to avoid expensive interprocess communication overhead, a universally generative naming scheme must be established. For many known datasets, the Voronoi partitioning [22] can be generated in $\Theta(n \log(n))$ -time [16] for 2D space and produces perfect partitions. However as the dimension increases this algorithm have less favorable run time complexities [17], making them inefficient for partitioning higher dimensional data sets.

3 Related Work

In this section we present related work on clustering large scale datasets. A variety of clustering methods have been proposed in the past that take advantage

of estimation techniques for machine learning, namely: *tree based clustering*, *dimensional reduction*, and *locality sensitive hashing (LSH)*.

The DBScan [13], Clique [3], and CLARANS [28] algorithms represent a successful progression of *density scanning techniques*. Although density scan algorithms are often scalable, they often show weaknesses in accuracy when scaling the number of dimensions.

Proclus [2] explored random projection for clustering. The merits of random projection are discussed in [10]. They suggest that random projection not only compresses sparse data sets, making them computationally more tractable, but also may help overall accuracy by alleviated round-off issues caused by non-homoscedastic variance. This occurs because random projection generates more spherical clusters in a more dense subspace. In addition to Proclus, various other methods and analysis have been proposed for clustering with random projections that provide bounds on the convergence and limits of random projection clustering. Florescu provides bounds on the scaling and convergence of projected clustering [15]. Their results closely follow the logic of Urruty [31] who find that the number of orthogonal projections required is logarithmic in n (where n is the number of vectors to be clustered). Following that, the probability of a random projection plane offering a good partitioning increases exponentially as the number of dimensions in the projected subspace increases. Bingham *et al* provide examples of projected clustering well below the JL bound [7] and Bartal *et al* make these assertions mathematically rigorous showing that the projected subspace is independent of the data's original dimensionality [6].

4 Tree-Walk RPHash

RPHash [8] is an algorithm for dense region and microcluster identification suitable as a precursor to more robust clustering algorithms like k -Means and agglomerative hierarchical clustering or as a standalone approximate clustering algorithm. In *RPHash*, both approximate and randomized techniques are employed to provide a stochastic element to the clustering algorithm. Due to this nature it has the tendency to produce some variation in its outputs. The Tree Walk RPHash (TWRP) extension attempts to mitigate this instability and provide stability to the results.

The basis of the Tree Walk RPHash (TWRP) algorithm is the RPHash clustering algorithm which has capacity to be amenable to distributed and streaming settings [9]. The RPHash algorithm arose from a realization that the degenerative cases for LSH k -nearest neighbor search is a useful method for identifying candidate cluster centers. A problem arises in the query step in which a particular LSH hash is disproportionately over represented. The result is that the algorithm must linearly scan and order all members of that hash bucket, to decide which are the nearest neighbors of the query vector. While bad for LSH K -NN, RPHash uses these outlier buckets as candidates for cluster centers. The original RPHash algorithm is shown in Algorithm 1. Here x_k is a vector from the dataset X . This vector is projected multiple times by a projection matrix p

taken from the larger set P . Then the vector is mapped to a region or bucket following the LSH scheme and the bucket counts are recorded. In the second pass over the data (the second **forall** statement) only those vectors assigned to the buckets with high counts are considered as the probable candidate cluster centroids (c_i).

In the original RPHash description, the LSH function is often chosen to be some generative function that uniformly covers a desired subspace. Ideally, an LSH function distributes vectors into buckets such that the underlying dense and sparse structures of the dataset are identifiable. One way to achieve this is to scale size of the LSH regions (often by the overall data's variance, and cardinality of the LSH mapping) to balance it between the two extremes: too dense or too sparse. To address this issue, a data aware LSH function was developed, called adaptive LSH (Algorithm 2), that attempts to optimize the distribution of hash buckets over the data-space. Adaptive LSH proceeds by taking a simple LSH function similar to the p-stable distribution LSH [20]. A key attribute to this particular type of LSH function is the immediate relationship between adjacent depths of the hash that we used in this algorithm. Composeable LSH allow us to balance the hash ID allocation.

Algorithm 1: 2-Pass RPHash

```

forall  $x_k \in X$  do
  forall  $p_i \in \mathbb{P}$  do
     $\tilde{x}_k \leftarrow \sqrt{\frac{m}{d}} p_i^\top x_k$ 
     $t = \mathbb{H}(\tilde{x}_k)$ 
     $L[k][i] = t$ 
     $C.add(t)$ 
forall  $x_k \in X$  do
  forall  $c_i \in C.top(K)$  do
    if  $L[k] \cap M[i][0] \neq 0$  then
       $\Delta = x_k - M[k]$ 
       $M[k] =$ 
         $M[k] + \Delta/count$ 
       $L[k].add(M[i][0])$ 

```

Result: Candidate Centroids

Algorithm 2: Adaptive LSH

```

 $i = 1$ 
 $ct, ct\_prev =$ 
   $C(\mathbb{H}^{i+1}(x)), C(\mathbb{H}^i(x))$ 
while  $i < n$  and
   $2ct > ct\_prev$  do
   $ct\_prev, i = ct, i + 1$ 
   $ct = C(\mathbb{H}^i(x))$ 
Result:  $\mathbb{H}^i(x)$ 

```

Definition 2 (LSH Composability). An LSH function $\mathbb{H}^n(x)$ that maps $x \in \mathbb{R}^n \rightarrow \mathbb{Z}_2^n$, is composable if there is a related function $\mathbb{H}^{n-1}(x_{n-1})$ that maps $x_{n-1} \in \mathbb{R}^{n-1} \rightarrow \mathbb{Z}_2^{n-1}$ where $\mathbb{H}^{n-1}(x_{n-1}) = (\mathbb{H}^n(x) + 1) \cup (\mathbb{H}^n(x) + 0)$ for all $x_n \in \mathbb{R}^n$

Although a variety of metrics are feasible, one of the simplest is to continue to extend the hash depth, so long as the subsequent hash count is greater than half of the parent hash count. This method comprises the adaptive LSH function (Algorithm 2).

We use the simple LSH function proposed in Indyk and Motwani’s original work on LSH for approximate near neighbor search [20]. We apply the hamming space LSH function to projected euclidean space by observing the signs of the projected vectors with respect to the origin. Formally, a *Signed Based Projected LSH function* is defined as: $H(X) = \sum sign(P(X))2^n$.

The TWRP method begins with the standard RPHash algorithm, but instead of only updating buckets, TWRP also increments the counts of all sub-hashes as well. This bares a slight resemblance to Liu *et al* [23] while adapting their algorithm to work in a streaming and distributed setting and without using any supervised learning. TWRP can use a variety of metrics for the tree splitting condition. Unlike Liu *et al* TWRP does not concern itself with the more complicated calculation to compute C4.5 entropy method. TWRP avoids the splitting of clusters with random hyperplanes by the virtue of its application to high dimensional data that is, the probability of splitting a cluster goes to zero as the dimensionality grows. The theorem below is a consequence of the curse of dimensionality.

Theorem 1. (*Hyperrectangle Splitting*) *The probability of splitting a hyper-rectangular region into two equal mass clusters where subsequent dimensional cuts contain the smaller of the two induced regions region approaches 0 exponentially in d .*

$$\lim_{d \rightarrow \infty} \frac{Vol(R) - Vol_{removed}(R)}{Vol(R)} = 0, R \text{ Rectangle} \in \mathbb{R}^d. \quad (1)$$

Proof.

Let X s.t. $x_j = [0\dots c\dots 0] \in \mathbb{R}^d$ is orthogonal, $c \in [0, 1)$, $\sum_i^n x_i = \mathbb{P}$, is a plane in \mathbb{R}^d , and $Vol(R) = 1$

Let: $S_1(p)$ be the volume of the projection of R on x_p Restrict $S_1(p) + \tilde{S}_1(p) = 1$, $S_1(p) \leq \tilde{S}_1(p)$ for all p

$$V(R_{s(X)}) = \prod_p^n S_1(p) \text{ where } S_1(p) \in U[0, \frac{1}{2})$$

$$V(R_{s(X)}) \leq 2^{-n} \text{ for all } n, \lim_{n \rightarrow \infty} 2^{-n} = 0$$

$$\Rightarrow V(R_{S(X)}) + V(\tilde{R}_{S(X)}) = V(s), V(\tilde{R}_{S(X)}) = V(s) \text{ as } d \rightarrow \infty \quad \blacksquare$$

Algorithm 3 and Algorithm 4 are the core elements of TWRP. The algorithm is linear in the input vector size x . For each vector TWRP must compute the projection and update the counter (Algorithm 3). This algorithm introduces two operations the $+$ to mean population weighted addition, and \gg for the bit shift operation. Projection using the approach of Achlioptas [1] can be performed in $dm/3$ operations where d is the original dimensionality and m is the projection sub-dimension. As each vector comes in it is projected onto the new lower dimension which the user inputs at runtime. There are m levels of hashing and as each level there are 2^{level} of buckets. Each projected vector is hashed onto a vector using the LSH scheme for every level. Thus all the vectors are present at all levels. We bound the memory as we only store a single vector in each bucket. This

is done by storing only the mean of the vectors coming into a specific bucket. We also store the count for each bucket. Whenever a vector is to be inserted into a bucket, we update its vector using weighted merge and increase the count by one. It is to be remembered that at this point we store the vector means in their original dimension. Thus we diminish the dimension only to determine in which bucket each vector goes in and then store the vector means in its original dimension into the buckets.

Algorithm 3: Online Tree Generation	Algorithm 4: Offline Tree Search
<pre> forall $x \in X$ do $\tilde{x} = \sqrt{\frac{m}{d}} p^\top x$ $h := \mathbb{H}(\tilde{x})$ while $h > 0$ do $h = h \gg 1$ $x' = C[h] + x$ $C.add(h, x')$ </pre>	<pre> forall $H \in sort(C.ids)$ do if $2C[H] < C[H \gg 1]$ then $C[H \gg 1] = 0$ $L = []$ forall $h \in sort(C.counts)$ do $L \leftarrow medoid(C[H])$ return L </pre>

The offline step consists of exploring and updating the count records (Algorithm 4). In general it follows a depth first search traversal for candidate clusters with a worse case complexity of exploring all non-leaf nodes, $\theta(2^{m-1})$. This is done by comparing the counts of each bucket at all levels. The hash values form a binary tree with each level having 2^{level} nodes. We compare the count of parent and child nodes/buckets and look for the possible centroids. We also remove the buckets with a threshold count value assuming these are possible noise. Thus only the dense part of the tree are kept for possible centroid values. We then return a overestimate of our weighted candidate centroids and use standard clustering algorithms such as k -means or Hierarchical Agglomerative to reduce these overestimated centroids to the desired number.

5 Experimental Results

TWRP has multiple configuration parameters, namely: projection distance, offline clustering algorithm, and sizes of the overestimated candidate cluster set. During preliminary testing of the TWRP, approximately 800 different configurations were evaluated using synthetically generated labeled data sets. Based on this testing, the best configuration that provided the consistently “best” WCSS result was selected. In particular, a configuration that projects data to 16 dimensions, using the offline k -means clusterer, and an overestimated centroid count of $10 \times k$ (where k is the number of clusters desired) was selected. This configuration is used to collect all the data reported in this section. Because of the stochastic nature of TWRP, the TWRP algorithm is configured to be run six times and the result with the best WCSS value is reported. The hardware used for testing is a 16 core Intel(R) Xeon(R) E5-2670 @ 2.6GHz, supporting 32 threads with 64GB of RAM.

5.1 Data Sets

The initial testing was performed with synthetic data generated as 10,000 vectors from dimensions 100 to 7,000. Each generated data set has 10 Gaussian clusters with labels recorded for all points. An accompanying noise test with synthetic data was performed. In particular, a 10,000 vector data set at dimension 1,000 was generated. We then replaced a percentage of vectors with randomly generated vectors but keeping the original label attached to the noise data. The noise was injected in percentages of 5-40 percent in steps of 5 percent. Finally, a set of scalability of the TWRP algorithm was performed also using synthetic data. In particular, we generated 8 synthetic datasets with 10 clusters and of 1,000 dimensions but varying the total number of vectors from 30,000 to 1,000,000.

Finally We used six real world datasets all available at the UCI machine learning repository. The *Human Activity Recognition Using Smartphones (HAR)* dataset is taken from the [5]. This dataset consists of 10,299 vectors containing 561 features consisting of 6 clusters. The *Smartphone Dataset for Human Activity Recognition (HAR) in Ambient Assisted Living (HARAAL)* dataset is taken from [12]. This data has 5744 vectors and 561 features. It has 6 clusters. The *gene expression cancer RNA-Seq Data Set (RNASEQ)* dataset is taken from [14]. This data has 801 vectors and 20531 features and 4 clusters. The *Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set (HAPT)* dataset is taken from [29]. This data has 10929 vectors and 561 features. This dataset has 12 cluster groups. The *Indoor Location Data (INLOC)* dataset is taken from [30]. This data has 21048 vectors and 529 features. It consists of 3 clusters. The *Gas Sensor Array Drift (GSAD)* dataset is taken from [33]. This data has 13910 vectors and 128 features. It is composed of 6 clusters.

5.2 Algorithms Used for Comparison

We compared the performance of TWRP against six standard well known algorithms, specifically:

- *k-Means*: This is the algorithm of Hartigan and Wong [18] implemented with the function `k-means` in R [24].
- Four methods of *Agglomerative Hierarchical clustering*, namely: Single Linkage, Complete Linkage, Average Linkage and Ward’s minimum variance method. At every stage the inter-cluster distances are recomputed by the Lance-Williams dissimilarity update formula according to the particular clustering method that is being used. For Ward’s [27]) clustering method, the dissimilarities are squared before updating the cluster. We have used the function `hclust` in R for implementing these algorithms.
- *Self-organizing Tree Algorithm (SOTA)*: This is relatively new algorithm based on neural network. It is implemented using the `sota` function in R (found in package `c1Valid`) [24].

These algorithms are selected due of their importance, popularity and availability in R statistical computing framework. These algorithms use FORTRAN, C and

C++ subroutines from R to make them run faster. The implementation language of TWRP is Java.

5.3 Comparison with Other Algorithms

The clustering performance and runtime for TWRP is compared to the six other algorithms described in Section 5.2. The clustering accuracy of TWRP is evaluated using 2 external clustering validation measures: Adjusted Rand Index (ARI) and Cluster Purity. We also use the internal measure WCSS for evaluation. ARI [19] measures the extent to which points from the same ground-truth partition appear in the same cluster, and the extent to which points from different ground-truth partitions are grouped in different clusters. ARI eliminates the chance of misjudging clustering outputs in cases where the output labels could be switched even if the clusters are well identified. The value of ARI lies between -1 and 1; an ARI of 1 denotes a perfect agreement between two partitions (and therefore a perfect clustering). Cluster purity [25] measures how many data points were correctly assigned to its original cluster. WCSS (also called WCSSE or SSE) is the within cluster sum of squared error. A lower value of WCSS indicates better clustering performance. It is basically the objective function that k -means algorithm tries to minimize in order to find suitable clusters.

The measured ARI, PURITY and WCSS for the synthetic datasets are plotted in Figure 1. The TWRP algorithm produces results on all measures are always better than that of k -means algorithm. TWRP has the value of 1 for ARI and PURITY for all these datasets, indicating perfect clustering. TWRP's WCSS also matches the baseline WCSS (*i.e.*, the actual WCSS for a dataset) for these synthetic data.

For the noise injected data, ARI, and WCSS are plotted in Figure 2. The WCSS of TWRP are much lower than single, average and complete linkage and SOTA algorithms. The performance of k -means and TWRP are comparable. We see as the noise grows to 40 percent (when the signal itself is poor) all the algorithms tend to perform poorly. This is because we randomly generated noise vectors and kept the original label.

The results for the real world datasets are summarized in Table 3. The results show that TWRP, k -means, and Wards algorithm have a very similar performance. In contrast, single link perform poorly.

Scalability results are shown in Table 1 using synthetically generated data sets at a range of dimensions between 100 to 7000. TWRP outperforms the other algorithms by a large margin. TWRP is almost 56 faster than k -means and 358 times faster than Agglomerative Hierarchical for synthetic datasets having 7000 dimensions and 10,000 vectors. TWRP shows very little runtime performance degradation as the dimension increases. The same cannot be said of the other algorithms. The runtime at a fixed dimension as the number of vectors was increased of was also studied. In particular Table 2 shows scalability of TWRP on a data set of 1,000 features as the number of vectors is varied between 30K to 1M. This shows a growth in the runtimes that is linear to the number of vectors.

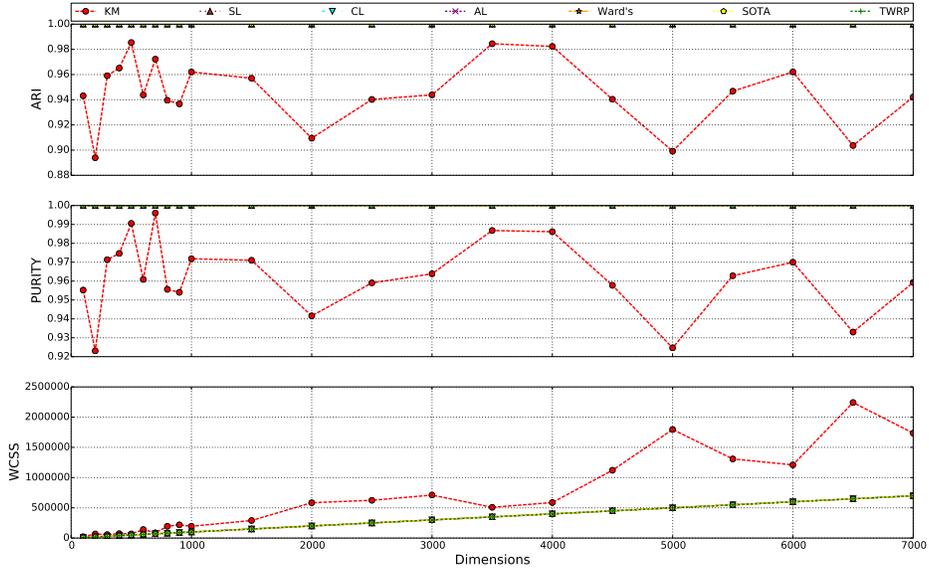


Fig. 1. ARI, Purity and WCSS Plotted Against Increasing dimension

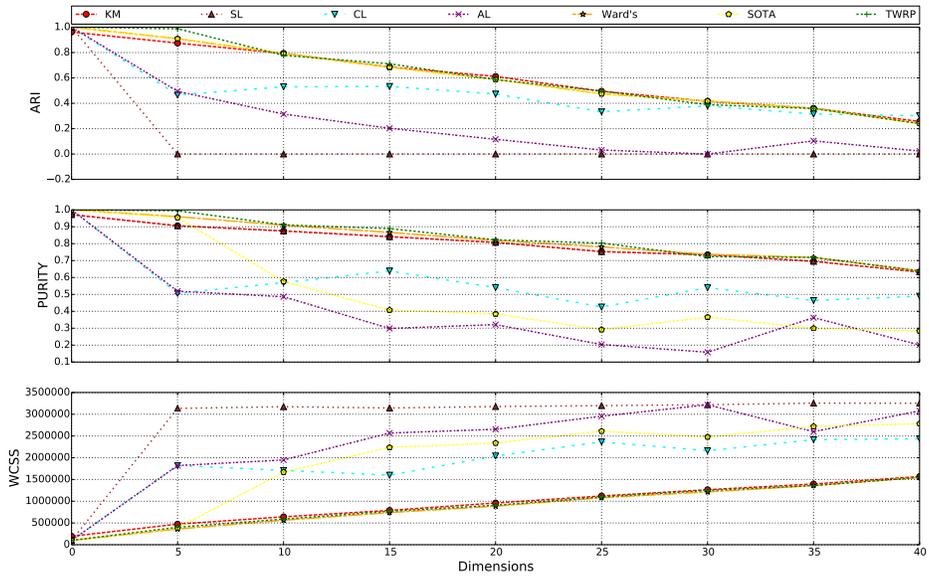


Fig. 2. ARI, Purity and WCSS Plotted Against Noise

Dataset	Measures	ARI	Purity	WCSS	Time
HAR	<i>k</i> -means	0.4610	0.6002	182 169	66.33
	SL	0.0000	0.1890	556 519	493.95
	CL	0.3270	0.3770	222 044	494.47
	AL	0.3321	0.3588	236 143	494.21
	Ward's	0.4909	0.6597	191 441	494.64
	SOTA	0.3143	0.3966	210 490	23.63
	TWRP	0.5125	0.5849	188 552	2.82
HAPT	<i>k</i> -means	0.3988	0.6498	2 498 381	182.90
	SL	0.0003	0.1821	6 023 519	601.98
	CL	0.0488	0.2505	4 584 352	602.42
	AL	0.0055	0.2046	5 491 388	602.04
	Ward's	0.4033	0.6624	2 617 769	602.68
	SOTA	0.3026	0.3848	2 990 195	31.13
	TWRP	0.3541	0.6257	2 593 802	6.54
HARAAL	<i>k</i> -means	0.2461	0.4240	1 618 089	23.55
	SL	0.0000	0.1964	3 166 056	148.43
	CL	0.0003	0.2002	3 043 579	148.53
	AL	0.0001	0.1972	3 097 976	148.45
	Ward's	0.2764	0.3929	1 653 179	148.58
	SOTA	0.2370	0.3785	1 814 593	12.30
	TWRP	0.2352	0.4076	1 636 495	2.64
INLOC	<i>k</i> -means	0.6048	0.8122	9 001 974 483	47.45
	SL	0.0000	0.4637	10 823 661 183	2266.78
	CL	0.7298	0.8413	9 257 876 517	2268.77
	AL	0.0000	0.4637	10 816 420 807	2267.82
	Ward's	0.7452	0.8469	9 040 839 873	2269.25
	SOTA	0.3954	0.7149	9 296 280 113	35.10
	TWRP	0.5322	0.7873	9 055 103 257	3.72
Gas-Sensor	<i>k</i> -means	0.1539	0.4427	27 714 236 160 297	9.05
	SL	0.0000	0.2165	192 076 751 899 323	42.61
	CL	0.0380	0.3474	47 050 045 285 192	42.57
	AL	0.0037	0.2865	75 622 509 139 114	42.45
	Ward's	0.2007	0.4378	31 162 058 051 998	42.87
	SOTA	0.0281	0.3435	46 727 103 818 336	11.93
	TWRP	0.2040	0.4709	31 000 000 000 000	1.86
RNASEQ	<i>k</i> -means	0.6438	0.8402	12 834 131	180.32
	SL	0.0007	0.3783	16 007 266	97.10
	CL	-0.0124	0.3758	15 692 260	97.10
	AL	0.0007	0.3783	16 007 266	97.08
	Ward's	0.5955	0.8202	12 916 461	97.09
	SOTA	0.3205	0.6317	13 632 923	87.72
	TWRP	0.5944	0.8077	20 357 469	7.45

Fig. 3. Performance for real data sets

Dimension	k -means	Average Link	SOTA	TWRP
100	3.9656	47.46	13.758	1.5930
300	13.3796	194.121	23.797	2.2386
500	32.92	454.777	36.046	2.7942
700	77.7298	672.759	48.963	3.7974
900	117.3675	929.6	61.759	4.2552
1000	142.2341	1064.178	68.86	4.6368
1500	237.0007	1789.142	96.795	6.3204
2000	366.0743	2450.813	127.972	7.9176
2500	431.5876	3108.654	155.968	9.6270
3000	542.0223	3771.886	185.23	11.1174
3500	631.8423	4435.349	220.562	12.9984
4000	741.915	5080.802	248.669	14.3292
4500	811.3911	5725.061	274.983	15.3030
5000	909.223	6362.574	301.314	17.2584
5500	975.2703	7006.91	324.314	19.3518
6000	1076.882	7645.155	359.911	20.2080
6500	1187.6062	8278.964	400.767	21.4596
7000	1340.8115	8520.85	428.15	23.7648

Table 1. Scalability with respect to Dimension(seconds)

size	30K	50K	70K	90K	100K	300K	600K	1M
TWRP	10.62	16.53	21.17	27.20	31.52	84.53	157.58	278.29

Table 2. Scalability with respect to size of dataset (seconds)

6 Conclusions

In this work we introduced the Tree-Walk Random Projection (TWRP) algorithm for clustering large high dimensional datasets with log-linear processing complexity. We applied the TWRP algorithm to real data and found that it performs comparably to other commonly used clustering methods. In tests with synthetic data, where clusters are well defined and spherical, we find that TWRP accuracy outperforms k -means and is equivalent to other standard techniques. In addition, TWRP performance is at par with k -means and Ward’s algorithm for noise injected into a dataset. We suspect this is a result of our simplified clustering metric favoring higher support partitions which is unable to distinguish a partition of noise from the true cluster ground truth. We believe choosing a different clustering metric such as C4.5 or within cluster sum of squared error, for the cluster tree bifurcation could potentially extend the effectiveness of TWRP to noisier data sets, however with an additional cost in computation and storage.

The complexity analysis of our method finds that it scales very well both in time and space complexity. The overall scalability is predictable and does not hide large constants.

As TWRP was designed to process very large high dimensional clustering problems, a key requirement is that it be fast. The results show quite clearly that TWRP achieves this goal by comprehensively beating other well known algorithm in its running time and scalability.

References

1. Achlioptas, D.: Database-friendly random projections. In: Proc of the 20th Symp on Principles of Database Systems. pp. 274–281 (2001)
2. Aggarwal, C.C., Wolf, J.L., Yu, P.S., Procopiuc, C., Park, J.S.: Fast algorithms for projected clustering. In: Proc of the 1999 ACM SIGMOD Int Conf on Management of Data (SIGMOD ’99). pp. 61–72 (1999)
3. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proc of the 1998 ACM SIGMOD Int Conf on Management of Data. pp. 94–105 (1998)
4. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: Foundations of Computer Science, 2006. FOCS’06. 47th Annual IEEE Symposium on. pp. 459–468. IEEE (2006)
5. Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.L.: UCI machine learning repository (2012), <https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>
6. Bartal, Y., Recht, B., Schulman, L.J.: Dimensionality reduction: beyond the johnson-lindenstrauss bound. In: Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms. pp. 868–887 (2011)
7. Bingham, E., Mannila, H.: Random projection in dimensionality reduction: Applications to image and text data. In: in Knowledge Discovery and Data Mining. pp. 245–250. ACM Press (2001)
8. Carraher, L.A., Wilsey, P.A., Moitra, A., , Dey, S.: Multi-probe random projection clustering to secure very large distributed datasets. In: 2nd International Workshop on Privacy and Security of Big Data (Oct 2015)

9. Carraher, L.A., Wilsey, P.A., Moitra, A., , Dey, S.: Random projection clustering on streaming data. In: IEEE 16th International Conference on Data Mining Workshops (ICDMW). pp. 708–715 (Dec 2016)
10. Dasgupta, S.: Experiments with random projection. In: Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence. pp. 143–151. UAI'00, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2000)
11. Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S.: Locality-sensitive hashing scheme based on p-stable distributions. In: Proceedings of the twentieth annual symposium on Computational geometry. pp. 253–262. SCG '04, ACM, New York, NY, USA (2004). <https://doi.org/http://doi.acm.org/10.1145/997817.997857>, <http://doi.acm.org/10.1145/997817.997857>
12. Davis, K.A., Owusu, E.B.: UCI machine learning repository (2016), <https://archive.ics.uci.edu/ml/datasets/Smartphone+Dataset+for+Human+Activity+Recognition+%28HAR%29+in+Ambient+Assisted+Living+%28AAL%29>
13. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD '96. vol. 96, pp. 226–231 (1996)
14. Fiorini, S.: UCI machine learning repository (2016), <https://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq>
15. Florescu, I., Molyboha, A., Myasnikov, A.: Scaling and convergence of projection sampling. Tech. rep., Stevens Institute of Technology (2009)
16. Fortune, S.: A sweepline algorithm for voronoi diagrams. In: Proceedings of the Second Annual Symposium on Computational Geometry. pp. 313–322. SCG '86, ACM, New York, NY, USA (1986)
17. Gavrilova, M.L.: Lecture notes in computer science. In: Kumar, V., Gavrilova, M., Tan, C., L'Ecuyer, P. (eds.) Comp. Sci. and Its Applications – ICCSA 2003, vol. 2669, chap. An Explicit Solution for Computing the Euclidean d-dimensional Voronoi Diagram of Spheres in a FP Arithmetic, pp. 827–835. Springer Berlin Heidelberg (2003)
18. Hartigan, J.A., Wong, M.A.: A k-means clustering algorithm. *JSTOR: Applied Statistics* **28**(1), 100–108 (1979)
19. Hubert, L., Arabie, P.: Comparing partitions. *Journal of Classification* **2**(1), 193–218 (1985)
20. Indyk, P., Motwani, R.: Approximate nearest neighbors: towards removing the curse of dimensionality. In: Proc of the 13th Annual ACM Symp on Theory of Computing. pp. 604–613. STOC '98, ACM, New York, NY, USA (1998). <https://doi.org/http://doi.acm.org/10.1145/276698.276876>
21. Jain, A.K.: Data clustering: 50 years beyond k-means. *Pattern Recogn. Lett.* **31**(8), 651–666 (Jun 2010). <https://doi.org/10.1016/j.patrec.2009.09.011>
22. Klein, R.: Abstract voronoi diagrams and their applications. In: *Computational Geometry and its Applications*, vol. 333, pp. 148–157. Springer Berlin Heidelberg (1988)
23. Liu, B., Xia, Y., Yu, P.S.: Clustering through decision tree construction. In: Proceedings of the Ninth International Conference on Information and Knowledge Management. pp. 20–29. CIKM '00, ACM, New York, NY, USA (2000). <https://doi.org/10.1145/354756.354775>, <http://doi.acm.org/10.1145/354756.354775>
24. Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K.: *cluster: Cluster Analysis Basics and Extensions* (2013), r package version 1.14.4
25. Manning, C.D., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press (2008)

26. McCallum, A., Nigam, K., Ungar, L.H.: Efficient clustering of high-dimensional data sets with application to reference matching. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 169–178. KDD '00, ACM, New York, NY, USA (2000). <https://doi.org/10.1145/347090.347123>
27. Murtagh, F., Legendre, P.: Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? *J. Classif.* **31**(3), 274–295 (Oct 2014). <https://doi.org/10.1007/s00357-014-9161-z>
28. Ng, R.T., Han, J.: Clarans: A method for clustering objects for spatial data mining. *IEEE Trans on Knowledge and Data Engineering* **14**(5), 1003–1016 (2002)
29. Reyes-Ortiz, J.L., Oneto, L., SamÁ, A., Parra, X., Anguita, D.: UCI machine learning repository (2015), <https://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions>
30. Torres-Sospedra, J., Montoliu, R., Martínez-Usó, A., Arnau, T.J., Avariento, J.P., Benedito-Bordonau, M., Huerta, J.: UCI machine learning repository (2014), <https://archive.ics.uci.edu/ml/datasets/UJIIndoorLoc>
31. Urruty, T., Djeraba, C., Simovici, D.: Clustering by random projections. In: Perner, P. (ed.) *Adv. in Data Mining. Theoretical Aspects and Applications*, vol. 4597, chap. Lecture Notes in Computer Science, pp. 107–119. Springer (2007). https://doi.org/10.1007/978-3-540-73435-2_9
32. Vempala, S.S.: *The Random Projection Method*. DIMACS Series, American Mathematical Society (2004)
33. Vergara, A., Fonollosa, J., Rodriguez-Lujan, I., Huerta, R.: UCI machine learning repository (2013), <https://archive.ics.uci.edu/ml/datasets/Gas+Sensor+Array+Drift+Dataset+at+Different+Concentrations>

Linear complexity algorithms for high dimensional SVM and regression problems with smart sparse regularization

Vadim Mottl¹, Olga Krasotkina², Valentina Sulimova³,
Alexey Morozov⁴, Ilya Pugach⁴, Alexander Tatarchuk⁵

¹ Computing Center of the Russian Academy of Sciences, Moscow, Russia

² Markov Processes International, Summit, NJ, USA

³ Tula State University, Tula, Russia

⁴ Moscow Institute of Physics and Technology, Moscow, Russia

⁵ Forexis, Moscow, Russia

Abstract. We consider the problems of regression estimation and SVM-based pattern recognition jointly as two particular problems of supervised dependence estimation in linear feature spaces under a complex of additional assumptions. First, it is assumed that the number of available features is huge, whereas that of training examples is moderate, and, thus, the search for a tiny subset in the universe of all the features is absolutely inevitable. As a consequence of this assumption, the computational complexity of the training algorithm has to be linear relative to the number of features, but it may remain polynomial with respect to that of training samples. Second, feature vectors of real-world objects concentrate, as a rule, closely to some small-dimensional feature subspace, which is expected to conserve the variety of objects' regressands or class indexes. Utilization of the latter property by due regularization of the training procedure would essentially improve the generalization performance of the inferred decision rule. We propose a new class of regularization functions and sparse training techniques of linear computational complexity relative to a large number of features, which naturally results from the combination of both above-mentioned assumptions on the nature of the data source.

Keywords: Sparse feature-based dependence estimation from empirical data, computational complexity of the regularized empirical risk minimization, conditions of the linear complexity in the large number of features.

1 Introduction

Supervised regression analysis and pattern recognition are two most typical cases of dependence estimation from empirical data [1]. In both problems it is required to recover the unknown dependence of a hidden variable $y \in \mathbb{Y}$ associated with any real-world object from the observable vector of its numerical features $\mathbf{x} = (x_1 \cdots x_n)^T \in \mathbb{R}^n$. The only difference between regression and pattern recognition is that in regression the target is an arbitrary real number $y \in \mathbb{Y} = \mathbb{R}$, whereas in recognition it is categorical, for instance, one of two real numbers $y \in \mathbb{Y} = \{-1, 1\}$. The commonly adopted approach to these problems implies finding a linear decision rule, respectively,

$$(a) \hat{y}(\mathbf{x}|\mathbf{a}, b) = \mathbf{a}^T \mathbf{x} + b: \mathbb{R}^n \rightarrow \mathbb{R} \quad \text{or} \quad (b) \hat{y}(\mathbf{x}|\mathbf{a}, b) = \begin{pmatrix} 1, & \mathbf{a}^T \mathbf{x} + b > 0 \\ -1, & \mathbf{a}^T \mathbf{x} + b < 0 \end{pmatrix}: \mathbb{R}^n \rightarrow \{-1, 1\}, \quad (1)$$

which would be applicable to any real-world object represented by its feature vector $\mathbf{x} \in \mathbb{R}^n$ and estimate the actual value of the hidden variable $y(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{Y}$.

These are two particular cases of John Nelder's Generalized Linear Model of dependencies [2,3], in which the goal variable of any kind $y \in \mathbb{Y}$ is related to the linear regression via the so-called link function:

$$\begin{cases} z(\mathbf{x}|\mathbf{a}, b) = \mathbf{a}^T \mathbf{x} + b: \mathbb{R}^n \rightarrow \mathbb{R} & \text{-- Generalized Linear Model of the dependence,} \\ q(y, z): \mathbb{Y} \times \mathbb{R} \rightarrow \mathbb{R}^+ & \text{-- link function.} \end{cases} \quad (2)$$

The link function (loss function, in Vladimir Vapnik's terminology [4]) is to be chosen by the observer and is meant to express his/her suggestion on how the Nature would penalize the estimate of unknown y for an object $\mathbf{x} \in \mathbb{R}^n$ represented by its generalized numerical linear feature $z(\mathbf{x}|\mathbf{a}, b)$.

Since the link function is chosen, the hyperplane parameters (\mathbf{a}, b) completely define the decision rule:

$$\hat{y}(\mathbf{x}|\mathbf{a}, b) = \arg \min_{y \in \mathbb{Y}} q(y, z(\mathbf{x}|\mathbf{a}, b)). \quad (3)$$

Particular dependence estimation problems differ from each other only in the choice of the link function, specifically:

- for regression, $q(y, z) = (y - z)^2$, $\hat{y}(\mathbf{x}|\mathbf{a}, b) = \mathbf{a}^T \mathbf{x} + b$; (4)
- for SVM pattern recognition, $q(y, z) = \max(0, 1 - yz)$, $\hat{y}(\mathbf{x}|\mathbf{a}, b) = \begin{cases} 1, & \mathbf{a}^T \mathbf{x} + b \geq 1, \\ -1, & \mathbf{a}^T \mathbf{x} + b < 1. \end{cases}$ (5)

From the viewpoint of the Generalized Linear Approach to dependence estimation, the quality of the hyperplane parameters (\mathbf{a}, b) is the average value of the loss $q(y, \mathbf{a}^T \mathbf{x} + b)$ over all real-world objects $(\mathbf{x}, y) \in \mathbb{R}^n \times \mathbb{Y}$, which is usually called the average risk of error, let it be denoted as $AvR(\mathbf{a}, b)$. However, average risk minimization $AvR(\mathbf{a}, b) \rightarrow \min(\mathbf{a}, b)$ is problematic because the properties of the hypothetical universe may be inexhaustibly complex.

Instead, it is commonly adopted to approximately estimate the average risk from a finite training set of real-world objects

$$\{(\mathbf{x}_j, y_j), j=1, \dots, N\}, \mathbf{X} = (\mathbf{x}_1 \cdots \mathbf{x}_N) = \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_N^T \end{pmatrix} (n \times N), \mathbf{x}_j \in \mathbb{R}^n, \mathbf{x}_i \in \mathbb{R}^n, \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} \in \mathbb{R}^N. \quad (6)$$

The estimate is arithmetic mean of the conditional loss values at all the objects of the training set, which is well known as Empirical Risk, let it be denoted as $EmpR(\mathbf{a}, b)$. This is the famous criterion of Empirical Risk minimization [4], which in our terms has the form

$$EmpR(\mathbf{a}, b) = (1/N) \sum_{j=1}^N q(y_j, \mathbf{a}^T \mathbf{x}_j + b) \rightarrow \min(\mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}). \quad (7)$$

The optimization problem is convex, if the link function $q(y, z)$ is chosen as convex with respect to z , as it is just the case with regression (4) and SVM pattern recognition (5).

When the practical problem originates from a medical or industrial domain, the available amount of data N is usually limited, whereas the observer tries to take into account as many features n as possible in fear of losing important outward exhibitions of entities. Therefore, the amount of features often far dominates that of training objects $n \gg N$. If so, the problem of empirical risk minimization (7) becomes ill posed – there exist a continuum of models (\mathbf{a}, b) that totally approximate the training data:

$$\mathbf{a}^T \mathbf{x}_j + b = z_j = \arg \min_{z \in \mathbb{R}} q(y_j, z), \quad j = 1, \dots, N, \text{ in particular,}$$

$$y_j - (\mathbf{a}^T \mathbf{x}_j + b) = 0 \text{ for regression (4) and } 1 - y_j(\mathbf{a}^T \mathbf{x}_j + b) < 0 \text{ for SVM (5).}$$

Learning as naive minimization $(\hat{\mathbf{a}}, \hat{b}) = \arg \min R(\mathbf{a}, b)$ has bad generalization performance, this effect is well-known as overfitting [4].

Thus, it is absolutely necessary to reduce the freedom of the model (\mathbf{a}, b) so as to align it with the amount of information contained in the training set. The commonly adopted way of enhancing the generalization performance is the additional requirement to minimize a real-valued regularization function $V(\mathbf{a}) \rightarrow \min$ meant as penalty on model complexity, and consider it along with (7) as a combined two-criteria optimization problem. Therefore, the learning problem is usually considered as that of regularized empirical risk minimization

$$\alpha V(\mathbf{a} | \mathbf{X}, \mu) + \text{Emp}R(\mathbf{a}, b) = \alpha V(\mathbf{a} | \mathbf{X}, \mu) + \sum_{j=1}^N q(y_j, \mathbf{a}^T \mathbf{x}_j + b) \rightarrow \min (\mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}). \quad (8)$$

Even if both link function $q(y, z)$ (2) and regularization function $V(\mathbf{a} | \mathbf{X}, \mu)$ are convex piecewise smooth functions of $\mathbf{a} \in \mathbb{R}^n$, the computational complexity of straightforward minimization of (8) cannot be lower than polynomial relative to the number of features n . But this is absolutely unacceptable if n is large.

The paper aims to show that there exists a way of solving the problem of regularization empirical risk minimization in linear time relative to $n > N$ for a class of regularization functions, which is adequate to the majority of practical situations.

The new class of data-dependent regularization functions $V(\mathbf{a} | \mathbf{X}, \mu)$ we propose endows the learning process with abilities that are, in our opinion, extremely useful for many practical problems of dependence estimation.

(a) The regularization function $V(\mathbf{a} | \mathbf{X}, \mu)$ enables the minimum point of the criterion (8) to indicate a subset of most relevant features $\hat{\mathbb{I}} \subset \mathbb{I} = \{1, \dots, n\}$. The preset selectivity parameter $\mu \geq 0$ controls the size of the selected subset from absence of selectivity $|\hat{\mathbb{I}}| = n$ if $\mu = 0$ to complete suppression of all features $|\hat{\mathbb{I}}| = 0$ if $\mu \rightarrow \infty$. The regularized criterion (10) possesses the oracle property [5], i.e., it is equivalent to the “pure” original criterion (7) with truncated direction vector $\mathbf{a} = (a_i, i \in \hat{\mathbb{I}} \subset \mathbb{I})$.

(b) Another parameter of the regularization function $V(\mathbf{a} | \mathbf{X}, \mu)$ is matrix $\mathbf{X} = (\mathbf{x}_1 \cdots \mathbf{x}_N)$ ($n \times N$) of all the feature vectors in the training set (6). It is an inherent property of Nature that real-world objects concentrate in the feature space, as a rule, closely to some low-dimensional subspace. The constellation of the training-set feature vectors in \mathbb{R}^n suggests a priori preferred orientation of the sought-for direction vector of the generalized data model (2), and the regularization coefficient $\alpha \geq 0$ controls the insistency of this preference.

(c) We propose two kinds of problem-oriented regularization functions $V(\mathbf{a} | \mathbf{X}, \mu)$ aimed at two different purposes of learning: to obtain a good estimator of the hidden variable $\hat{y}(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{Y}$ or to estimate the actual hidden parameter of the data source $\mathbf{a} \in \mathbb{R}^n$.

(d) The iterative algorithm of selective empirical risk minimization has linear computational complexity of a single iteration relative to the number of features n and polynomial computational complexity with respect to the size of the training set N . The algorithm converges in a final number of steps n .

The advantages of this methodology are illustrated by experiments on real-world data that originate from two practical problems of stock market data analysis (regression) and evoked EEG potential classification (pattern recognition).

The proofs of the theorems formulated in the paper are available in [6].

2 Selective empirical risk minimization

2.1 Selective ridge regularization

Let's focus first on the very need of regularization. As to the idea of data-dependent regularization $V(\mathbf{a} | \mathbf{X}, \mu)$ (8), we will return to it in Section 3.

The idea of regularization goes back to Tikhonov's mathematical methodology of solving incorrectly formulated problems [7,8,9]. It turned out to be exceptionally useful in mathematical statistics, primarily, for the estimation of regression dependences from noisy data. The simplest kind of quadratic regularization is well known under the name of ridge regularization $\mathbf{a}^T \mathbf{a} = \sum_{i=1}^n a_i^2 \rightarrow \min$ [10,11], often called l_2 regularization. The additive mix of l_2 and l_1 regularizations, well known as Elastic Net [12]

$$\gamma \sum_{i=1}^n (a_i^2 + \mu |a_i|) + \sum_{j=1}^N q(y_j, \mathbf{a}^T \mathbf{x}_j + b) \rightarrow \min (\mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}), \quad (9)$$

endows the criterion with the ability to select a subset of most relevant features $\hat{\mathbb{I}} \subset \mathbb{I} = \{1, \dots, n\}$ and completely suppress the others. However, the minimum point of (9) does not coincide with that of the initial empirical risk criterion (7) when the direction vector consists of only active components $\mathbf{a} = (a_i, i \in \hat{\mathbb{I}} \subset \mathbb{I})$. In other words, Elastic Net does not possess the oracle property [5].

We propose to combine l_2 and l_1 regularizations by the rule of exclusive “or” instead of pure summation “+”:

$$J(\mathbf{a}, b | \mu) = \gamma \sum_{i=1}^n \left(\begin{matrix} 2\mu |a_i|, & |a_i| \leq \mu \\ \mu^2 + a_i^2, & |a_i| > \mu \end{matrix} \right) + \sum_{j=1}^N q(y_j, \mathbf{a}^T \mathbf{x}_j + b) \rightarrow \min (\mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}), \quad (10)$$

where $\mu \geq 0$ is the selectivity parameter. If $\mu = 0$, the regularization function coincides with the usual ridge regularization $\gamma \mathbf{a}^T \mathbf{a} + \text{Emp}R(\mathbf{a}, b) \rightarrow \min$. With small regularization coefficient $\gamma \ll 1$, practically, $\gamma \rightarrow 0$, ridge regression “almost not destroys” the generic idea of empirical risk minimization, and only gives preference to short direction vectors in case of indifference. When the selectivity parameter grows $\mu > 0$, the penalty $\mu |a_i|$ drives to zero the coefficients at redundant features, which weakly contribute to diminishing of the empirical risk. Further growth of the selectivity parameter $\mu \rightarrow \infty$ results finally in complete zeroing of all the coefficients. We will see that it is easy to compute this maximal value of the selectivity parameter from the training set of features $\mathbf{X} = (\mathbf{x}_1 \cdots \mathbf{x}_N)$ (6).

The regularized criterion (10) should be considered as “almost” possessing the oracle property [14] if $\gamma \ll 1$, i.e., $\gamma \rightarrow 0$. In this case, it is equivalent to the “pure” original criterion (7) with truncated direction vector $\mathbf{a} = (a_i, i \in \hat{\mathbb{I}} \subset \mathbb{I})$. However, the computational complexity of it in the number of features is polynomial, just as that of (7), which is inadmissible for high-dimensional problems $n \gg N$.

Since function $q(y, z)$ (4)-(5) is assumed to be convex and piecewise smooth with respect to $z \in \mathbb{R}$ for each $y \in \mathbb{Y}$, the criterion remains convex and piecewise smooth as a whole. If considered as a convex piecewise smooth problem of general kind, problem (10) seems, at first glance, to be unsolvable with lower computational complexity than polynomial with respect to the number of features n [15]. However, the regularization function here is of a special kind, and we will see in the next Section that the computational complexity in n is, in fact, linear.

2.2 Disjoint formulation of the problem of sparse empirical risk minimization

It is obvious that the criterion of the sparse empirical risk minimization (10) is equivalent to the optimization problem with a greater number of variables:

$$\begin{cases} F(\mathbf{a}, b, z_1, \dots, z_n | \mu) = \gamma \sum_{i=1}^n \left(\frac{2\mu |a_i|}{\mu^2 + a_i^2}, |a_i| \leq \mu \right) + \sum_{j=1}^N q(y_j, z_j) \rightarrow \min (\mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}, \mathbf{z} \in \mathbb{R}^N), \\ z_j = \mathbf{a}^T \mathbf{x}_j + b, \quad j = 1, \dots, N. \end{cases} \quad (11)$$

We will call this formulation disjoint, because the objective function is sum of two partial criteria, which are functions of, respectively, the direction vector $\mathbf{a} \in \mathbb{R}^n$ and the generalized features $z_j = z(\mathbf{x}_j | \mathbf{a}, b)$ (2) of the training objects $\mathbf{z} = (z_1 \cdots z_N)^T \in \mathbb{R}^N$, in contrast to the initial unified formulation (10).

The left summand of the disjoint formulation $\gamma \sum_{i=1}^n (\dots)$ is sum of functions each depending on only one component of the direction vector a_i . We will see below in Section 4 that, due to this fact, despite of a greater number of variables, the disjoint form of the selective criterion (10) immediately suggests, under some quite lenient assumption on the link function $q(y, z)$ (2), the way to solve the problem of selective empirical risk minimization with linear computational complexity to the number of features n and polynomial complexity relative to the size of the training set N .

3 Data-dependent regularization of the empirical risk

3.1 Low-dimensional subspace of real-world objects in the feature space

When the number of features n is large, the variety of all feasible points in the feature space $\mathbf{x} \in \mathbb{R}^n$ is inconceivably rich. Albert Einstein's conjecture that "God is subtle but he is not malicious"¹ may be interpreted, in particular, as "the Nature does not create anything extra". At the same time, the set of features is invented by the observer in the fear to lose some important outward exhibitions of natural entities. Thus, there exist insufficient number of objects in the real world to fill such enormous volume that exists only in the observer's imagination.

In the mathematical language, this means that feature vectors of real-world objects $\mathbf{x} \in \mathbb{R}^n$ and, thus, those of the training set $\mathbf{x}_j \in \mathbb{R}^n$, $j=1, \dots, N$, concentrate, as a rule, closely to some small-dimensional manifold [16], whose dimensionality is essentially smaller than the size N of the training set. This assumption is fully consistent with our experience.

In this paper, we restrict our consideration to a particular case when this is affine manifold, i.e. biased linear subspace, which is expected to conserve the variety of objects' regressands or class indexes.

It is actually assumed that the immense variety of real-world objects is "almost completely" exhausted by projections of their feature vectors $\mathbf{x} \in \mathbb{R}^n$ on the small-dimensional subspace defined by few main principal axes of an almost degenerate concentration ellipsoid. The eigenvectors of the matrix

$$\mathbf{X}\mathbf{X}^T = \sum_{j=1}^N \mathbf{x}_j \mathbf{x}_j^T \quad (n \times n) \quad (12)$$

computed from the training set (6) are estimates of these principal axes.

¹ "Raffiniert ist der Herr Gott aber boshhaft ist er nicht" – inscribed in German above the fireplace in the Old Fine Hall of Princeton University.

3.2 Preferred orientation of the direction vector within the concentration ellipsoid in the feature space

In most practical problems, the primary aim of data analysis is to find a good estimator of the hidden variable (1). If so, the a priori preferred location of the direction vector $\mathbf{a} \in \mathbb{R}^n$ is that within the principal subspace as a linear combination of the main eigenvectors of matrix $\mathbf{X}\mathbf{X}^T$. The greater the criterion

$$\mathbf{a}^T \mathbf{X}\mathbf{X}^T \mathbf{a} = \mathbf{a}^T \left(\sum_{j=1}^N \mathbf{x}_j \mathbf{x}_j^T \right) \mathbf{a} \rightarrow \max, \quad \mathbf{a}^T \mathbf{a} = \text{const}, \quad (13)$$

the closer the direction vector to one of the eigenvectors and, so, the more preferable it is.

However, to be consistent with the major learning criterion (8), the regularization criterion must be formulated as minimization requirement. If matrix $\mathbf{X}\mathbf{X}^T$ ($n \times n$) was be of full rank, the role of the regularization criterion could serve the condition $\mathbf{a}^T (\mathbf{X}\mathbf{X}^T)^{-1} \mathbf{a} \rightarrow \min$, but this matrix is singular when $n > N$ (6).

The following Theorem suggests a way-out, that consist in replacing inversion of the singular matrix $\mathbf{X}\mathbf{X}^T$ ($n \times n$) by inversion of the matrix $\mathbf{X}^T \mathbf{X}$ ($N \times N$), which has full rank when $n > N$ and the training-set vectors $(\mathbf{x}_1 \cdots \mathbf{x}_N)$, $\mathbf{x}_j \in \mathbb{R}^n$ are linearly independent.

Theorem 1. The condition (13) $\mathbf{a}^T \mathbf{X}\mathbf{X}^T \mathbf{a} \rightarrow \max$, $\mathbf{a}^T \mathbf{a} = \text{const}$, is fulfilled if

$$\mathbf{a}^T \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{a} \rightarrow \min, \quad \mathbf{a}^T \mathbf{a} = \text{const}. \quad \blacksquare \quad (14)$$

Plain fusion of selective and data-dependent regularizations (10) and (14) results in a slightly more complicated combined unified criterion

$$J(\mathbf{a}, b | \gamma, \mu, \alpha) = \gamma \sum_{i=1}^n \left(2\mu |a_i|, |a_i| \leq \mu \right) + \alpha \mathbf{a}^T \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{a} + \sum_{j=1}^N q(y_j, \mathbf{a}^T \mathbf{x}_j + b) \rightarrow \min (\mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}), \quad (15)$$

where $\alpha \geq 0$ is the regularization coefficient and $\gamma \ll 1$. This criterion is convex, it can be formulated in the disjoint form like (11), and its two left summands $\gamma \sum_{i=1}^n (\dots) + \alpha \mathbf{a}^T \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{a}$ are a more complicated function than sum of a_i .

3.3 Orthogonal preferred orientation of the direction vector to the concentration ellipsoid in the feature space

There exist practical problems in which the aim of the data analysis is not finding the most precise estimator of the hidden variable (1), but the most accurate estimation of the direction vector $\mathbf{a} \in \mathbb{R}^n$ of the unknown dependence $y \cong \mathbf{a}^T \mathbf{x} + b$, which is assumed to exist in reality.

For instance, in econometric problems of investment portfolio analysis [17], the direction vector of regression model $\mathbf{a} \in \mathbb{R}^n$ has the meaning of capital sharing over n stock market assets, usually, under constraint $\mathbf{1}^T \mathbf{a} = \text{const}$, $\mathbf{1}^T = (1 \cdots 1) \in \mathbb{R}^n$. It is assumed that $\mathbf{a} \in \mathbb{R}^n$ is deliberately chosen by a hidden person who was guided, first, by a suggestion on the covariance matrix of the presumably random vector of features $\mathbf{x} \in \mathbb{R}^n$ (asset returns, i.e., relative profitabilities), and, second, the desire to obtain the minimal variance if the resulting random regressand $y \in \mathbb{R}$ (return on the investment portfolio).

If the average value of all the features is zero

$$\sum_{j=1}^N \mathbf{x}_j = \mathbf{0} \in \mathbb{R}^n, \quad (16)$$

it is easy to see that, in this case, the most rational choice of the direction vector is

$$\mathbf{a}^T \mathbf{X} \mathbf{X}^T \mathbf{a} \rightarrow \min, \quad \mathbf{1}^T \mathbf{a} = \text{const}. \quad (17)$$

It is just the inverse requirement relative to (13), i.e., the a priori preferred orientation of the direction vector is that orthogonal to the concentration subspace of the training set in the feature space. As a result, we have the problem

$$J(\mathbf{a}, b | \gamma, \mu, \alpha) = \gamma \sum_{i=1}^n \left(\frac{2\mu |a_i|, |a_i| \leq \mu}{\mu^2 + a_i^2, |a_i| > \mu} \right) + \alpha \mathbf{a}^T \mathbf{X} \mathbf{X}^T \mathbf{a} + \sum_{j=1}^N q(y_j, \mathbf{a}^T \mathbf{x}_j + b) \rightarrow \min(\mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}), \quad (18)$$

that differs from (15) only by the data-dependent regularization function (17) instead of (14).

3.4 Disjoint formulation of the regularized empirical risk minimization problem

Both regularized empirical risk minimization problems (15) and (18) allow for a unified formulation that remains the same as for the preferred orientation of the direction vector within the concentration ellipsoid in the feature space, as well as for orthogonal preferred orientation to the concentration ellipsoid. The disjoint formulation includes also the case of absence of a priori preferences.

Theorem 2. The regularized empirical risk minimization problems (15) and (18) have the common disjoint form, which is slightly more complicated than that of the unregularized selective problem (11), and whose specificity for particular cases (15) and (18) consists only in specific linear transformations of the feature vectors $\mathbf{x}_j \in \mathbb{R}^n \rightarrow \tilde{\mathbf{x}}_{\alpha, j} \in \mathbb{R}^n$:

$$\begin{cases} F(\mathbf{a}, b, r_1, \dots, r_N, z_1, \dots, z_N | \mu) = \gamma \sum_{i=1}^n \left(\frac{2\mu |a_i|, |a_i| \leq \mu}{\mu^2 + a_i^2, |a_i| > \mu} \right) + \alpha \sum_{j=1}^N r_j^2 + \sum_{j=1}^N q(y_j, z_j) \rightarrow \\ \min(\mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}, r_1, \dots, r_N \in \mathbb{R}, z_1, \dots, z_N \in \mathbb{R}), \\ r_j = \tilde{\mathbf{x}}_j^T \mathbf{a}, \quad z_j = \mathbf{a}^T \mathbf{x}_j + b, \quad j = 1, \dots, N \end{cases} \quad (19)$$

where $\tilde{\mathbf{x}}_j \in \mathbb{R}^n$, $j = 1, \dots, N$, are vector columns of matrix $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_1 \dots \tilde{\mathbf{x}}_N)$ ($n \times N$) obtained from $\mathbf{X} = (\mathbf{x}_1 \dots \mathbf{x}_N)$ (6) by the respective linear transformation:

$$(a) \quad \tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_1 \dots \tilde{\mathbf{x}}_N) = \begin{pmatrix} \tilde{\mathbf{x}}_1 \\ \vdots \\ \tilde{\mathbf{x}}_N \end{pmatrix} = \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} (n \times N), \quad \tilde{\mathbf{x}}_j \in \mathbb{R}^n, \quad \tilde{\mathbf{x}}_i \in \mathbb{R}^N, \quad (20)$$

in the case of preferred orientation of the direction vector within the concentration ellipsoid in the feature space (15), Section 3.2;

$$(b) \quad \tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_1 \dots \tilde{\mathbf{x}}_N) = \begin{pmatrix} \tilde{\mathbf{x}}_1 \\ \vdots \\ \tilde{\mathbf{x}}_N \end{pmatrix} = \mathbf{X} = (\mathbf{x}_1 \dots \mathbf{x}_N) = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{pmatrix} (n \times N), \quad \mathbf{x}_j \in \mathbb{R}^n, \quad \mathbf{x}_i \in \mathbb{R}^N, \quad (21)$$

in the case of preferred orientation of the direction vector orthogonal to the concentration ellipsoid in the feature space (18), Section 3.3;

$$(c) \quad \alpha = 0 \quad (22)$$

in the case of no preferred orientation of the direction vector (10). ■

4 Computational complexity of the disjoint problem – linear in the number of features and polynomial in the training-set size

4.1 Properties of the disjoint empirical risk minimization problem

The necessary and sufficient minimum condition for the convex function under $2N$ equality constraints (19) is the saddle point of the respective Lagrangian as function of $2N$ Lagrange multipliers, let them be denoted as (ξ_1, \dots, ξ_N) , $\xi_j \in \mathbb{R}$, at con-

straints $r_j = \tilde{\mathbf{x}}_j^T \mathbf{a}$, and $(\lambda_1, \dots, \lambda_N)$, $\lambda_j \in \mathbb{R}$, at $z_j = \mathbf{a}^T \mathbf{x}_j$. It is just this fact that underlies the Theorem 4 we will formulate in the next Section 4.2.

However, before doing this, we have to introduce an additional notion of function

$$\varphi(y, \lambda | \gamma) = -\inf_{z \in \mathbb{R}} ((1/2\gamma)q(y, z) + \lambda z), \quad \lambda \in \mathbb{R}. \quad (23)$$

Theorem 3. Function $\varphi(y, \lambda | \lambda)$ (23) is convex in $\lambda \in \mathbb{R}$ for any link function $q(y, z)$, $z \in \mathbb{R}$. ■

The definition of this function contains the operation “inf”, whose result depends on the variable $\lambda \in \mathbb{R}$ and is finite, in the general case, not for all its values. Since in this paper function $q(y, z)$ is assumed to be convex with respect to $z \in \mathbb{R}$ for each $y \in \mathbb{Y}$, its derivative is nondecreasing function. If, in addition, $q(y, z)$ is chosen as differentiable by z except a finite number of points in \mathbb{R} , at which the left-hand or right-hand derivative exists, the range of the respective derivatives is known, let it be denoted as

$$g_{\inf}(y) = \inf_{z \in \mathbb{R}} \frac{\partial}{\partial z} q(y, z) \leq \frac{\partial}{\partial z} q(y, z) \leq g_{\sup}(y) = \sup_{z \in \mathbb{R}} \frac{\partial}{\partial z} q(y, z). \quad (24)$$

In particular, it may be that $g_{\inf}(y) = -\infty$, or $g_{\sup}(y) = \infty$, or both.

Now we are ready to formulate the promised theorem on a linear complexity way of solving the problem of selective empirical risk minimization.

4.2 The dual form of the disjoint empirical risk minimization problem

Theorem 4. In all the cases (20)-(22), it is enough to solve the dual convex programming problem

$$\begin{cases} W(\lambda_1, \dots, \lambda_N, \xi_1, \dots, \xi_N | \mathbf{X}, \tilde{\mathbf{X}}, \mathbf{y}, \gamma, \mu) = \\ \left\{ \frac{1}{2} \sum_{i=1}^n \left[\max \left[0, \left(\sum_{j=1}^N (\lambda_j x_{j,i} + \xi_j \tilde{x}_{j,i}) \right)^2 - \mu^2 \right] \right\} + \sum_{j=1}^N \varphi(y_j, \lambda_j) \rightarrow \min, \\ \sum_{j=1}^N \lambda_j = 0, \quad -\frac{1}{2\gamma} g_{\sup}(y_j) \leq \lambda_j \leq -\frac{1}{2\gamma} g_{\inf}(y_j), \end{cases} \quad (25)$$

then, to compute independent solutions for the components $i = 1, \dots, n$ of the direction vector $\hat{\mathbf{a}} = (\hat{a}_1 \dots \hat{a}_n)^T \in \mathbb{R}^n$, and, finally, to find the bias of the hyperplane $\hat{b} \in \mathbb{R}$:

$$\hat{a}_i = \begin{cases} 0, & \left(\sum_{j=1}^N (\hat{\lambda}_j x_{j,i} + \hat{\xi}_j \tilde{x}_{j,i}) \right)^2 \leq \mu^2, \\ \sum_{j=1}^N \hat{\lambda}_j x_{j,i}, & \left(\sum_{j=1}^N (\hat{\lambda}_j x_{j,i} + \hat{\xi}_j \tilde{x}_{j,i}) \right)^2 > \mu^2, \end{cases} \quad \hat{b} = \frac{1}{N} \sum_{j=1}^N \left(\hat{z}_j(\lambda_j) - \sum_{i=1}^n \hat{a}_i x_{j,i} \right), \quad (26)$$

where $\hat{z}_j(\lambda_j) = \arg \min_{z \in \mathbb{R}} \left(\frac{1}{2\gamma} q(y_j, z_j) + \lambda_j z_j \right)$, $j = 1, \dots, N$. ■

Computing of the transformed feature vectors $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_1 \dots \tilde{\mathbf{x}}_N)$ for case (a) is trivial. It has linear computational complexity in the number of features and polynomial complexity in the training-set size:

$$\mathbf{X}^T \mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \mathbf{x}_1 & \cdots & \mathbf{x}_1^T \mathbf{x}_N \\ \vdots & \ddots & \vdots \\ \mathbf{x}_N^T \mathbf{x}_1 & \cdots & \mathbf{x}_N^T \mathbf{x}_N \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n x_{1,i} x_{1,i} & \cdots & \sum_{i=1}^n x_{1,i} x_{N,i} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^n x_{N,i} x_{1,i} & \cdots & \sum_{i=1}^n x_{N,i} x_{N,i} \end{pmatrix} (N \times N)$$

The way of minimizing criterion (19) with linear computational complexity with respect to the number of features n and polynomial computational complexity with respect to the doubled size of the training set $2N$ differs from the unregularized version (11) in only a few details.

In the next Section, we will consider a way of iterative solving the dual convex problem (25). The computational complexity of a single iteration will be linear relative to the number of features n .

5 Iterative algorithms of solving the dual problem

5.1 Piecewise differentiability of the general dual objective function

Let (2) be a parametric generalized family of dependencies defined by a link function $\{q(y, z): \mathbb{Y} \times \mathbb{R}\}$ convex by z , and $\{(\mathbf{x}_j, y_j), j=1, \dots, N\}$ (6) be the given training set. It is additionally assumed that $q(y, z)$ is differentiable by z in \mathbb{R} except a finite number of points. Then, to find the estimates of the hyperplane parameters $(\hat{\mathbf{a}}, \hat{b})$, it is enough to solve the convex dual optimization problem (25) with the given parameters (α, μ) and respective transformed training feature vectors $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_1 \dots \tilde{\mathbf{x}}_N)$ in accordance with the conditions (20), (21) or (22), as formulated in Theorem 4.

Each of n summands in the objective function $W(\lambda_1, \dots, \lambda_N, \xi_1, \dots, \xi_N | \mathbf{X}, \tilde{\mathbf{X}}, \mathbf{y})$ of (25) differs from zero only if $\sum_{j=1}^N (\lambda_j x_{j,i} + \xi_j \tilde{x}_{j,i}) > \mu$. Let $\mathbb{I}(\lambda_1, \dots, \lambda_N, \xi_1, \dots, \xi_N)$ stand for the subset of “active” features:

$$\mathbb{I}(\boldsymbol{\lambda}, \boldsymbol{\xi}) = \mathbb{I}(\lambda_1, \dots, \lambda_N, \xi_1, \dots, \xi_N) = \left\{ i: \sum_{j=1}^N (\lambda_j x_{j,i} + \xi_j \tilde{x}_{j,i}) > \mu \right\} \subseteq \mathbb{I} = \{1, \dots, n\}. \quad (27)$$

It is convenient for us to formulate the dual problem (25) in an equivalent form with respect to this notation:

$$\begin{cases} W(\boldsymbol{\lambda}, \boldsymbol{\xi} | \mathbf{X}, \tilde{\mathbf{X}}, \mathbf{y}, \gamma, \mu) = (1/2) \left\{ \boldsymbol{\lambda}^T \left(\sum_{i \in \mathbb{I}(\boldsymbol{\lambda}, \boldsymbol{\xi})} \mathbf{x}_i \mathbf{x}_i^T \right) \boldsymbol{\lambda} + 2 \boldsymbol{\xi}^T \left(\sum_{i \in \mathbb{I}(\boldsymbol{\lambda}, \boldsymbol{\xi})} \tilde{\mathbf{x}}_i \mathbf{x}_i^T \right) \boldsymbol{\lambda} + \right. \\ \left. \boldsymbol{\xi}^T \left(\sum_{i \in \mathbb{I}(\boldsymbol{\lambda}, \boldsymbol{\xi})} \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^T \right) - \sum_{i \in \mathbb{I}(\boldsymbol{\lambda}, \boldsymbol{\xi})} \mu^2 \right\} + \sum_{j=1}^N \varphi(y_j, \lambda_j | \gamma) \rightarrow \min(\boldsymbol{\lambda}, \boldsymbol{\xi}), \\ \sum_{j=1}^N \lambda_j = 0, \quad -(1/2\gamma) g_{\text{sup}}(y_j) \leq \lambda_j \leq -(1/2\gamma) g_{\text{inf}}(y_j), \end{cases} \quad (28)$$

$$\text{where } \boldsymbol{\lambda}^T \left(\sum_{i \in \mathbb{I}(\boldsymbol{\lambda}, \boldsymbol{\xi})} \mathbf{x}_i \mathbf{x}_i^T \right) \boldsymbol{\lambda} + 2 \boldsymbol{\xi}^T \left(\sum_{i \in \mathbb{I}(\boldsymbol{\lambda}, \boldsymbol{\xi})} \tilde{\mathbf{x}}_i \mathbf{x}_i^T \right) \boldsymbol{\lambda} + \boldsymbol{\xi}^T \left(\sum_{i \in \mathbb{I}(\boldsymbol{\lambda}, \boldsymbol{\xi})} \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^T \right) - \sum_{i \in \mathbb{I}(\boldsymbol{\lambda}, \boldsymbol{\xi})} \mu^2 = \sum_{i \in \mathbb{I}(\lambda_1, \dots, \lambda_N, \xi_1, \dots, \xi_N)} \left\{ \left(\sum_{j=1}^N (\lambda_j x_{j,i} + \xi_j \tilde{x}_{j,i}) \right)^2 - \mu^2 \right\}.$$

The function $W(\boldsymbol{\lambda}, \boldsymbol{\xi} | \dots)$ is differentiable at each point $(\boldsymbol{\lambda}, \boldsymbol{\xi}) \in \mathbb{R}^{2N}$. Moreover, it is even twice differentiable, and, so, Newton’s method is appropriate to find the solution of the dual problem. If $(\boldsymbol{\lambda}^k, \boldsymbol{\xi}^k)$ is the current approximation to the solution, then a supposedly better solution in Newton’s sense $(\tilde{\boldsymbol{\lambda}}^{k+1}, \tilde{\boldsymbol{\xi}}^{k+1})$ is defined by the completely differentiable convex programming problem

$$\begin{cases} (\tilde{\boldsymbol{\lambda}}^{k+1}, \tilde{\boldsymbol{\xi}}^{k+1}) = \arg \min \tilde{W}^k(\boldsymbol{\lambda}, \boldsymbol{\xi} | \mathbf{X}, \tilde{\mathbf{X}}, \mathbf{y}, \gamma, \mu) = \\ \arg \min \left\{ (1/2) \sum_{i \in \mathbb{I}(\boldsymbol{\lambda}^k, \boldsymbol{\xi}^k)} \left\{ \left(\sum_{j=1}^N (\lambda_j x_{j,i} + \xi_j \tilde{x}_{j,i}) \right)^2 - \mu^2 \right\} + \sum_{j=1}^N \varphi(y_j, \lambda_j | \gamma) \right\}, \\ \sum_{j=1}^N \lambda_j = 0, \quad -(1/2\gamma) g_{\text{sup}}(y_j) \leq \lambda_j \leq -(1/2\gamma) g_{\text{inf}}(y_j), \end{cases} \quad (29)$$

which differs from (28) only by the fixed summation domain $\mathbb{I}(\boldsymbol{\lambda}^k, \boldsymbol{\xi}^k)$.

To avoid boring mathematical reasoning in the case of the convex link function of general kind $q(y, z)$, we omit, in this Section, the explanation of how to solve the convex programming problem (29). We will see in the next Section that, in particular

cases of regression (4) and SVM pattern recognition (5), these will be quadratic programming problems easily solvable by traditional computational means.

5.2 Newton's iterative method with variable step length

It is well seen that the approximation $(\boldsymbol{\lambda}^k, \boldsymbol{\xi}^k) = (\lambda_1^k, \dots, \lambda_N^k, \xi_1^k, \dots, \xi_N^k)$ to the sought-for solution at step k occurs in (29) only via the subset of active features (27), let it be denoted as

$$\mathbb{I}^k = \mathbb{I}(\boldsymbol{\lambda}^k, \boldsymbol{\xi}^k). \text{ We will start with the full feature set } \mathbb{I}^0 = \{1, \dots, n\}. \quad (30)$$

Let the supposedly better solution of the dual problem $(\tilde{\boldsymbol{\lambda}}^{k+1}, \tilde{\boldsymbol{\xi}}^{k+1})$ (29) at step k be found. It may happen that the length of Newton's step is too large, and it should be shortened. To check this necessity, it is enough to compare the values of the dual criterion (28) at points $(\boldsymbol{\lambda}^k, \boldsymbol{\xi}^k)$ and $(\tilde{\boldsymbol{\lambda}}^{k+1}, \tilde{\boldsymbol{\xi}}^{k+1})$:

$$\text{If } W(\tilde{\boldsymbol{\lambda}}^{k+1}, \tilde{\boldsymbol{\xi}}^{k+1} | \mathbf{X}, \tilde{\mathbf{X}}, \mathbf{y}, \gamma, \mu) \leq W(\boldsymbol{\lambda}^k, \boldsymbol{\xi}^k | \mathbf{X}, \tilde{\mathbf{X}}, \mathbf{y}, \gamma, \mu), \quad (31)$$

the current iteration is successful, and $(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\xi}^{k+1}) = (\tilde{\boldsymbol{\lambda}}^{k+1}, \tilde{\boldsymbol{\xi}}^{k+1})$;

$$\text{if } W(\tilde{\boldsymbol{\lambda}}^{k+1}, \tilde{\boldsymbol{\xi}}^{k+1} | \mathbf{X}, \tilde{\mathbf{X}}, \mathbf{y}, \gamma, \mu) > W(\boldsymbol{\lambda}^k, \boldsymbol{\xi}^k | \mathbf{X}, \tilde{\mathbf{X}}, \mathbf{y}, \gamma, \mu), \quad (32)$$

the step is to be shortened.

To find the appropriate length of Newton's step, we apply one-dimensional optimization of (28), namely, the golden section algorithm:

$$\begin{cases} \tau^{k+1} = \arg \min W[(\tau \boldsymbol{\lambda}^k, \tau \boldsymbol{\xi}^k) + ((1-\tau)\tilde{\boldsymbol{\lambda}}^{k+1}, (1-\tau)\tilde{\boldsymbol{\xi}}^{k+1}) | \mathbf{X}, \tilde{\mathbf{X}}, \mathbf{y}, \gamma, \mu], \\ 0 \leq \tau \leq 1, \\ (\boldsymbol{\lambda}^{k+1}, \boldsymbol{\xi}^{k+1}) = (\tau^{k+1} \boldsymbol{\lambda}^k, \tau^{k+1} \boldsymbol{\xi}^k) + ((1-\tau^{k+1})\tilde{\boldsymbol{\lambda}}^{k+1}, (1-\tau^{k+1})\tilde{\boldsymbol{\xi}}^{k+1}). \end{cases} \quad (33)$$

Actually, the algorithm iteratively runs over the subsets of regressors $\mathbb{I}^k = \mathbb{I}(\boldsymbol{\lambda}^k, \boldsymbol{\xi}^k) \subset \mathbb{I} = \{1, \dots, n\}$ (27) without cycles because $W(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\xi}^{k+1} | \mathbf{X}, \tilde{\mathbf{X}}, \mathbf{y}, \gamma, \mu) \leq W(\boldsymbol{\lambda}^k, \boldsymbol{\xi}^k | \mathbf{X}, \tilde{\mathbf{X}}, \mathbf{y}, \gamma, \mu)$ at each step. Thus, the stopping condition

$$\mathbb{I}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\xi}^{k+1}) = \mathbb{I}(\boldsymbol{\lambda}^k, \boldsymbol{\xi}^k) \text{ will be achieved after a finite number of steps.} \quad (34)$$

5.3 Numerical realization of an iteration for particular cases of link function

To complete description of the iterative algorithm, it remains only to specify the ways of solving the constrained problem (29) at each step in the particular cases of regression and SVM pattern recognition. The specificity is entirely contained in the functions $\varphi(y, \lambda | \gamma) = -\min_{z \in \mathbb{R}} ((1/2\gamma)q(y, z) + \lambda z)$ (23) and in the inequality constraints $-(1/2\gamma)g_{\sup}(y_j) \leq \lambda_j \leq -(1/2\gamma)g_{\inf}(y_j)$.

5.3.1 Regression

Theorem 5. In the particular case of regression (4), (23) and (24), we have

$$g_{\sup}(y) = \infty, \quad g_{\inf}(y) = -\infty, \quad \varphi(y, \lambda | \gamma) = \frac{1}{2}\gamma\lambda^2 - y\lambda, \quad (35)$$

and the solution to the dual problem (29) at the k th step of the iteration process is defined by the system of linear equations $(N+1) \times (N+1)$

$$\begin{aligned}
& (\tilde{\boldsymbol{\lambda}}^{k+1} = (\tilde{\lambda}_1^{k+1}, \dots, \tilde{\lambda}_N^{k+1}) \in \mathbb{R}^N, \mathbf{1}^T \tilde{\boldsymbol{\lambda}}^{k+1} = 0, \tilde{\boldsymbol{\xi}}^{k+1} = (\tilde{\xi}_1^{k+1}, \dots, \tilde{\xi}_N^{k+1}) \in \mathbb{R}^N): \\
& \underbrace{\begin{pmatrix} \sum_{i \in \mathbb{I}^k} \mathbf{x}_i \mathbf{x}_i^T + \gamma \mathbf{I}_{N \times N} & \sum_{i \in \mathbb{I}^k} \mathbf{x}_i \tilde{\mathbf{x}}_i^T & \mathbf{1}_N \\ \sum_{i \in \mathbb{I}^k} \tilde{\mathbf{x}}_i \mathbf{x}_i^T & \sum_{i \in \mathbb{I}^k} \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^T + \gamma \mathbf{I}_{N \times N} & \mathbf{0}_N \\ \mathbf{1}_N^T & \mathbf{0}_N^T & 0 \end{pmatrix}}_{2N+1} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\xi} \\ \eta \end{pmatrix} = \begin{pmatrix} \mathbf{y} \\ \mathbf{0}_N \\ 0 \end{pmatrix} \quad (36)
\end{aligned}$$

where $\eta \in \mathbb{R}$ is the idle Lagrange multiplier at the constraint $\mathbf{1}^T \boldsymbol{\lambda} = \sum_{j=1}^N \lambda_j = 0$. The final bias estimate (26) has the form

$$\hat{\mathbf{b}} = \frac{1}{N} \sum_{j=1}^N \left(y_j - \sum_{i=1}^n \hat{a}_i \tilde{x}_{\alpha, j, i} \right). \blacksquare \quad (37)$$

It is clear that problem (36) is of polynomial computational complexity with respect to $N+1$ [18].

5.3.2 SVM pattern recognition

Theorem 6. In the particular case of SVM pattern recognition (5),

$$q(y_j, z) = \max(0, 1 - y_j z) = \begin{cases} 1 - y_j z, & z < 1 \\ 0, & z \geq 1 \end{cases}, \quad 0 \leq y_j \lambda_j \leq 1/2\gamma, \quad (38)$$

the solution to the dual problem at the k th step of the iteration process (29) is that of quadratic programming problem

$$\begin{aligned}
& (\tilde{\boldsymbol{\lambda}}^{k+1} = (\tilde{\lambda}_1^{k+1}, \dots, \tilde{\lambda}_N^{k+1}) \in \mathbb{R}^N, \mathbf{1}^T \tilde{\boldsymbol{\lambda}}^{k+1} = 0, \tilde{\boldsymbol{\xi}}^{k+1} = (\tilde{\xi}_1^{k+1}, \dots, \tilde{\xi}_N^{k+1}) \in \mathbb{R}^N): \\
& \begin{cases} \tilde{W}^k(\lambda_1, \dots, \lambda_N, \xi_1, \dots, \xi_N | \mathbf{X}, \tilde{\mathbf{X}}, \mathbf{y}) = (1/2) \sum_{i \in \mathbb{I}^k} \left(\sum_{j=1}^N (\lambda_j x_{j,i} + \xi_j \tilde{x}_{j,i}) \right)^2 - \sum_{j=1}^N y_j \lambda_j \rightarrow \min, \\ \sum_{j=1}^N \lambda_j = 0, \quad 0 \leq y_j \lambda_j \leq 1/2\gamma, \quad j = 1, \dots, N. \end{cases} \quad (39)
\end{aligned}$$

The final bias estimate (26) has the form

$$\hat{\mathbf{b}} = - \frac{\sum_{i=1}^n \left(\hat{a}_i \sum_{j: 0 < y_j \hat{\lambda}_j < (1/2\gamma)} y_j \hat{\lambda}_j x_{j,i} \right) + \sum_{j: y_j \hat{\lambda}_j = (1/2\gamma)} \hat{\lambda}_j}{\sum_{j: 0 < y_j \hat{\lambda}_j < (1/2\gamma)} y_j \hat{\lambda}_j}. \blacksquare \quad (40)$$

This is a standard quadratic programming problem [15,18] of polynomial computational complexity relative to N .

6 Rough regularization path along the selectivity axis

6.1 Active interval of the selectivity parameter

The selectivity parameter $0 \leq \mu < \infty$ is the main hyperparameter of the dependence estimation problem in general (19) and of its dual form in particular (25) or (27)-(28). If $\mu = 0$, the criterions possess no selectivity property at all, and all the estimated components of the direction vector remain active (26).

On the contrary, when the selectivity is large enough $\mu \rightarrow \infty$, all the direction vector components become zero. What is the maximal value of selectivity that completely suppresses all the features? We will denote it as μ_0 because it retains 0 active features.

Let us imagine the subset of active features in (28) to be empty:

$$\begin{cases} \sum_{j=1}^N \varphi(y_j, \lambda_j | \gamma) \rightarrow \min(\lambda \in \mathbb{R}^N, \xi \in \mathbb{R}^N), \\ \sum_{j=1}^N \lambda_j = 0, \quad -(1/2\gamma)g_{\text{sup}}(y_j) \leq \lambda_j \leq -(1/2\gamma)g_{\text{inf}}(y_j). \end{cases} \quad (41)$$

Remember that, in accordance with Theorem 3, functions $\varphi(y, \lambda | \gamma)$ (23) are convex in the respective range. Since ξ does not occur in the objective function, it is always possible to consider the result of minimization by this variable as $\xi^* = \mathbf{0} \in \mathbb{R}^N$.

Let $(\lambda_1^*, \dots, \lambda_N^*)$ be solutions of the truncated problem (41), and μ_0 be defined as

$$\mu_0 = \max_{i=1, \dots, n} \left(\sum_{j=1}^N \lambda_j^* \tilde{x}_{\alpha, j, i} \right). \quad (42)$$

Then, if $\mu = \mu_0$, we have $(\hat{\lambda}_1, \dots, \hat{\lambda}_N) = (\lambda_1^*, \dots, \lambda_N^*)$ in (25)

$$\begin{cases} (\hat{\lambda}_{1, \mu_0}, \dots, \hat{\lambda}_{N, \mu_0}) = \arg \min W(\lambda_1, \dots, \lambda_N | \tilde{\mathbf{X}}_{\alpha}, \mathbf{y}, \gamma, \mu_0), \\ \sum_{j=1}^N \lambda_j = 0, \quad -(1/2\gamma)g_{\text{sup}}(y_j) \leq \lambda_j \leq -(1/2\gamma)g_{\text{inf}}(y_j), \end{cases} \quad (43)$$

and $\hat{a}_i = 0$ for all $i = 1, \dots, n$ in (26), i.e. we obtain the trivial empty model.

Thus, the active interval of the selectivity parameter is $0 \leq \mu \leq \mu_0$.

6.2 The idea of the regularization path and its rough implementation

The number of active features will be growing from 0 to n as μ diminishing from μ_0 to 0. This is just the exact idea of the full regularization path [19,20]. Theoretically, the number of bifurcation points, where the number of active features changes, will not be lesser than n , but in reality it will be much greater than n , because this process is far from being monotonic. As a result, such a procedure would be too time consuming in the case of large number of features.

Here, we consider a rough implementation of this idea. The experience shows that it is expedient to divide the interval $[10^{-8}\mu_0 \approx 0, \mu_0]$ into a number of $m \leq n$ subintervals in logarithmic scale:

$$\mu_l = 10^{-8(l/m)} \mu_0, \quad l = 0, 1, \dots, m, \quad \text{i.e., } \mu_0 = 10^{-0} \mu_0, \quad \mu_m = 10^{-8} \mu_0 \approx 0. \quad (44)$$

The rough regularization path starts with $l = 0$, which corresponds to $\mu = \mu_0$ and the trivial dual problem (43) that yields the empty model $\hat{a}_i = 0$ for all $i = 1, \dots, n$ (26). Nevertheless, the result of the iteration process $(\hat{\lambda}_{1, \mu_0}, \dots, \hat{\lambda}_{N, \mu_0})$ should be stored.

Each next value of the selectivity parameter $\mu = \mu_l$ will almost coincide with the previous value $\mu = \mu_{l-1}$, and the iteration process (Section 5.2) started with the previous solution $(\hat{\lambda}_{1, \mu_{l-1}}, \dots, \hat{\lambda}_{N, \mu_{l-1}})$ will converge after only a few iterations, in most cases, after one or two iterations. The number of non-zero components of the direction vector will gradually grow (26).

Finally, at the last step $\mu = \mu_m \approx 0$, we will have the direction vector with almost all active components.

We will see in the next Section that the entire regularization path will take approximately the same computation time as the iteration process for one single value of the selectivity parameter as (30)-(34) in Section 5.2.

7 Experimental study of the computational complexity of dependence estimation with growing number of features

7.1 The aim of the experimental study

The aim of the study is to experimentally measure the dependence of the processing time of the Newton's iterative algorithm (30)-(34) in Section 5.2, let it be denoted as T , from the dimension of feature vectors n with emphasized attention to the number of iterations.

$$\text{Let } (\mathbf{X}^n, \mathbf{y}) = \{(\mathbf{x}_j^n, y_j), j=1, \dots, N\}, \mathbf{x}_j^n = (x_{j,1} \cdots x_{j,n}) \in \mathbb{R}^n, \quad (45)$$

be a fixed training set (6) that will serve as the basis for the experimental study. Before the detailed description of the experimental setup below in Section 7.3, in the next Section 7.2 we consider two kinds of data $(\mathbf{X}^n, \mathbf{y})$ that served in our experiments as sources of the basic training set (45).

7.2 Real-world training sets

7.2.1 Training set 1: Numerical regression – Returns based analysis of investment portfolios

The first training set originates from the practical problem of stock market data analysis. The set of objects $j=1, \dots, N$ is a succession of time intervals (days, weeks, months) of stock exchange trading, the role of features is played by the so-called returns of a large set $\mathbb{I} = \{i=1, \dots, n\}$ of stock market assets, namely, the relative changes in their monetary price during each respective time interval $x_{j,i} \in \mathbb{R}$.

The object of interest is an investment company that accumulates money of investors, in particular, common people, with the purpose of saving it from devaluation. The entire amount of the accumulated capital is unknown, but it is assumed that it is fully invested in some subset of stock market assets $\mathbb{I}^* \subseteq \mathbb{I} = \{i=1, \dots, n\}$ in unknown proportions

$$(a_i, i=1, \dots, n), a_i = 0 \text{ if } i \notin \mathbb{I}^*, \sum_{i=1}^n a_i = 1.$$

In general, some of these hidden values may be negative $a_i \in \mathbb{R}$ if the investment company is allowed to borrow money from outside sources (the so-called hedge funds). But for socially important companies (mutual or pension funds) this is prohibited, then $a_i \geq 0$. In both cases, the capital share $(a_i, i=1, \dots, n)$ said the portfolio structure is just the subject of public interest, but remains legal secret of the company.

What is especially important is that the full cost of all the assets making the investment portfolio depends on both returns of stock market assets $x_{j,i} \in \mathbb{R}$ and the portfolio structure. The cost is unknown, but the company is obliged to report its periodic returns $y_j \in \mathbb{R}$, namely, the relative changes in the entire cost, just like periodic returns of the assets $x_{j,i} \in \mathbb{R}$.

It was stated by William Sharp, Nobel Prize winner in economics in 1990 [21,22], that the sequence of periodic returns of an investment company is linear combination of daily returns of assets in which the capital is invested:

$$y_t \cong \sum_{i=1}^n a_i x_{t,i}, t=1, \dots, N.$$

The returns of the portfolio $\mathbf{y} = (y_1 \cdots y_N)^T \in \mathbb{R}^N$ have to be regularly reported by the investment company, and the asset returns $\mathbf{X} = (\mathbf{x}_1 \cdots \mathbf{x}_N) (n \times N)$, $\mathbf{x}_t = (x_{t,1} \cdots x_{t,n})^T \in \mathbb{R}^n$, are registered and stored by the stock market data system. William Sharp proposed the technique of Returns Based Style Analysis

$$(\hat{a}_1 \cdots \hat{a}_n) = \arg \min \sum_{t=1}^N \left(y_t - \sum_{i=1}^n a_i x_{t,i} \right)^2, \quad \sum_{i=1}^n a_i = 1.$$

as a means to reconstruct the hidden capital share $(a_1 \cdots a_n)^T \in \mathbb{R}^n$ by joint processing publicly available data.

As the basis

$$\mathbf{X} = (\mathbf{x}_1 \in \mathbb{R}^n \cdots \mathbf{x}_N \in \mathbb{R}^n) = \begin{pmatrix} \mathbf{x}_1^T \in \mathbb{R}^n \\ \vdots \\ \mathbf{x}_N^T \in \mathbb{R}^n \end{pmatrix} (n \times N) \quad (46)$$

of the Training set 1 (6), we used $n=650$ actual time series of monthly stock market indexes having almost zero mean values $\mathbf{x}_i = (x_{t,i}, t=1, \dots, N) \in \mathbb{R}^N, i=1, \dots, n, n=650$, each covering 20 years, i.e., consisting of 240 return values $t=1, \dots, N, N=240$. The eigenvalues of the inner product matrix, having been placed in decreasing ordered, quickly fall

$$\mathbf{X}\mathbf{X}^T = \begin{pmatrix} \mathbf{x}_1^T \mathbf{x}_1 & \cdots & \mathbf{x}_1^T \mathbf{x}_n \\ \vdots & \ddots & \vdots \\ \mathbf{x}_n^T \mathbf{x}_1 & \cdots & \mathbf{x}_n^T \mathbf{x}_n \end{pmatrix} (n \times n), \quad \mathbf{X}\mathbf{X}^T \mathbf{v}_i = \zeta_i \mathbf{v}_i \in \mathbb{R}^n, \quad (47)$$

$$\zeta_1 = 16369.8, \quad \zeta_2 = 1435.2, \quad \zeta_3 = 1076.2, \quad \zeta_{20} = 134.0 < 0.01\zeta_1, \quad \zeta_{50} = 32.2 < 0.002\zeta_1,$$

what is evidence of drastic correlation between the indexes (features).

On the basis of some special assumptions (see [23] for details), we chose a mental portfolio composition $(a_i^*, i \in \mathbb{I}^*)$, $\mathbb{I}^* \subseteq \mathbb{I} = \{i=1, \dots, n\}$, $a_i^* = 0$ if $i \notin \mathbb{I}^*$, $\sum_{i=1}^n a_i^* = 1$,

that included $|\mathbb{I}^*| = 13$ active assets. The returns $\mathbf{y} = (y_1 \cdots y_N)^T \in \mathbb{R}^N$ of the respective hypothetical portfolio (regressands of the training set) were generated as independent zero-mean Gaussian random values

$$y_t = \sum_{i \in \mathbb{I}^*} a_i^* x_{t,i} + \xi_t \quad \text{with } 10\% \text{ noise variance } \sigma^2(\xi_t) = 0.1 \left((1/N) \sum_{i \in \mathbb{I}^*} a_i^* x_{t,i} \right). \quad (48)$$

The combination $(\mathbf{X} = (\mathbf{x}_1 \cdots \mathbf{x}_N), \mathbf{y})$ is just the Training set 1.

7.2.2 Training set 2: SVM pattern recognition – Classification of evoked potentials in Electro Encephalo Grams

Electroencephalography is a method of testing the electrical activity of the brain by jointly processing several electrical signals registered in parallel at several points on the surface of the skull. It was originally invented and is broadly used as a means to study mechanisms by which human behavior is generated, in particular, for brain diseases diagnosis.

However, in the past decades, electroencephalography has become the basis of many brain-computer interfaces, which decode neural response to different stimuli into commands that, for instance, operate external devices [24].

The experiments we refer to in this paper [25,26] are concerned with another purpose of analyzing responses of a multi-channel electroencephalogram (EEG) to outward stimuli. It is assumed that the person whose EEG is processed is an experienced mammologist able to reliably distinguish between X-ray mammograms of women with breast cancer and those of healthy women. These studies pursue the aim to essentially improve productivity of rare pronounced experts by way of, first, accelerating the screening of mammographic images up to ten pictures per second, and, second, immediately detecting the eventual potentials evoked in the expert's EEG by a target (cancer) image among a crowd of non-target ones before the expert becomes aware of this fact.

In our experiments, we analyzed 66-channel EEG signals registered in parallel at 66 points on the scalp of an expert. Initial EEG signals from each electrode are filtered with a cutoff frequency of 40 Hz, see [25,26] for details.

The diagnostic session of a set of mammograms is organized as follows. The expert is shown a sequence of mammograms at a speed of 10 Images per second, namely, 100 ms per mammogram. The sequence was divided into groups, each of 11 images. There are two kinds of groups called target and non-target ones. A non-target group consists entirely of healthy mammograms, whereas each target group contains exactly one cancer image at a random place surrounded by healthy ones at both sides. Thus, the time duration of each group is 1100 ms. The EEG signal was originally registered at a frequency of 1000 Hz, but we applied 11 times thinning, so, one signal fragment corresponding to one group of mammograms, target or non-target ones, finally consists of 100 samples. Since 66 channels are registered, $n = 6600$ is the entire dimension of the ‘‘EEG feature vector’’ $\mathbf{x}_j = (x_{j,1} \cdots x_{j,n})^T \in \mathbb{R}^n$, which relates to the j -th image group and is built as concatenation of all the 66 channels.

The classification of EEG potentials consists in detection whether the registered EEG signal $\mathbf{x}_j \in \mathbb{R}^n$ is a response to a target image $y_j = 1$ or not target one $y_j = -1$. From the mathematical point of view, this is a two-class pattern recognition problem, which was formulated in [26] as that of selective SVM pattern recognition (5), (11):

$$\begin{cases} F(\mathbf{a}, b, z_1, \dots, z_n | \gamma, \mu) = \gamma \sum_{i=1}^n \left(\frac{2\mu |a_i|}{\mu^2 + a_i^2}, |a_i| \leq \mu \right) + \sum_{j=1}^N q(y_j, z_j) \rightarrow \min(\mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}, \mathbf{z} \in \mathbb{R}^N), \\ q(y_j, z_j) = \max(0, 1 - y_j z_j), \\ z_j = \mathbf{a}^T \mathbf{x}_j + b, \quad j = 1, \dots, N. \end{cases} \quad (49)$$

In [26], the full set of 755 experimental objects (target and non-target image groups) was randomly split into the training set of $N = 196$ objects (98 target and 98 non target ones) and the test set of $M = 559$ objects (275 target and 284 non target objects).

In this paper, we study only the computational complexity of the training problem, therefore, only the size of the training set $N = 196$ is of interest. Thus, the basis of Training set 2 has the same structure as the basis of Training set 1 (46)

$$\mathbf{X} = \left(\mathbf{x}_1 \in \mathbb{R}^n \cdots \mathbf{x}_N \in \mathbb{R}^n \right) = \begin{pmatrix} \mathbf{x}_1^T \in \mathbb{R}^n \\ \vdots \\ \mathbf{x}_N^T \in \mathbb{R}^n \end{pmatrix} (n \times N),$$

with the number of features $n = 6600$ and number of objects $N = 196$.

Just as in Training set 1 (47), the 6600 EEG features turned out to be tightly correlated – the eigenvalues of the inner product matrix quickly fall:

$$\zeta_1 = 16369.8, \quad \zeta_2 = 1435.2, \quad \zeta_3 = 1076.2, \quad \zeta_{20} = 134.0 < 0.01\zeta_1, \quad \zeta_{50} = 32.2 < 0.002\zeta_1.$$

7.3 Chronometry of the learning process on data sets of growing dimensionality with fixed value of the selectivity parameter

In all the experiments we assumed that there exists an insistent a priori preference of the direction vector orientation within the concentration ellipsoid in the feature space (20). Such a preference is expressed by quite a large value of the regularization coefficient, so that we assumed $\alpha = 10$:

$$\tilde{\mathbf{X}}_\alpha = \tilde{\mathbf{X}}_{10} = \mathbf{X} \left(\mathbf{I}_N + 10(\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{X})^{-1} \right).$$

Let $i = 1, \dots, n$ be some natural numeration of features. We transformed the given basic training set (45) into a succession of partial training sets with growing dimensionality of the feature space

$$(\mathbf{X}^m, \mathbf{y}) = \{(\mathbf{x}_j^m, y_j), j=1, \dots, N\}, \mathbf{x}_j^m = (x_{j,1} \cdots x_{j,m}) \in \mathbb{R}^m, m=1, \dots, n,$$

and, so, into a succession of transformed partial training sets

$$(\tilde{\mathbf{X}}_\alpha^m, \mathbf{y}) = \{(\tilde{\mathbf{x}}_{\alpha,j}^m, y_j), j=1, \dots, N\}, \tilde{\mathbf{x}}_{\alpha,j}^m = (\tilde{x}_{\alpha,j,1} \cdots \tilde{x}_{\alpha,j,m}) \in \mathbb{R}^m, m=1, \dots, n. \quad (50)$$

Let $0 < \mu < \infty$ be a fixed value of the selectivity parameter that suppresses about three quarters of features, i.e., one fourth of them remains active. For the fixed value of the selectivity parameter $\alpha = 10$, each of these partial training sets (50) defines a succession of convex training criteria $J(\mathbf{a}^m, b | \mu, \alpha)$

$$J(\mathbf{a}^m, b | \mu, \alpha) = \gamma \sum_{i=1}^m \left(\frac{2\mu |a_i|}{\mu^2 + a_i^2}, |a_i| \leq \mu \right) + \sum_{j=1}^N q \left(y_j, \sum_{i=1}^m a_i \tilde{x}_{\alpha,j,i} + b \right) \rightarrow \min(\mathbf{a}^m \in \mathbb{R}^m, b \in \mathbb{R}) \quad (51)$$

as functions of the direction vector of growing dimension $m=1, \dots, n$. On the other hand, each of the partial training sets (50) defines the respective succession of dual criteria (25)

$$\begin{cases} W(\lambda_1, \dots, \lambda_N | \tilde{\mathbf{X}}_\alpha^m, \mathbf{y}, \mu) = \\ \frac{1}{2} \sum_{i=1}^m \left\{ \max \left[0, \left(\sum_{j=1}^N \lambda_j \tilde{x}_{\alpha,j,i} \right)^2 - \mu^2 \right] \right\} + \sum_{j=1}^N \varphi(y_j, \lambda | \mu) \rightarrow \min, \\ \sum_{j=1}^N \lambda_j = 0, \quad -(1/2\gamma) g_{\text{sup}}(y_j) \leq \lambda_j \leq -(1/2\gamma) g_{\text{inf}}(y_j), \end{cases} \quad (52)$$

which are functions of the same fixed number of Lagrange multipliers $\lambda_1, \dots, \lambda_N$.

First, we applied a standard iterative convex programming procedure available in Matlab to each of the full criteria $J(\mathbf{a}^m, b | \mu, \alpha)$ (51), and registered the run time $T_{\text{full}}(m)$. Then, we applied the iterative procedure (30)-(34) from Section 5.2 to each of dual criteria (52) and registered the run time $T_{\text{dual}}(m)$.

We repeated this experiment two times for real-world Training set 1 (Section 7.2.1) and real-world Training set 2 (Section 7.2.2). Each time, we used the respective version of the algorithm for regression (Section 5.3.1) in the case of Training set 1, or for SVM pattern recognition (Section 5.3.2) in the case of Training set 2.

The results are shown in Figure 1. As it is seen from both plots, the numerical computational complexity of the initial regularized empirical risk minimization problem (23) relative to the number of features remains polynomial and extremely high – the run time T_{full} swiftly grows as m increases. Numerical solving of this problem in the disjoint formulation (25)-(26) qualitatively reduces the computational complexity.

The slope of the growth of the run time T_{dual} as function of m is determined by two factors – the run time of a single iteration and the number of iterations.

In the case of SVM pattern recognition, the average slope is less as half of that in the case of regression. This must be a consequence of the very principle of SVM (39), which takes into account only “support” objects $y_j \lambda_j > 0$ at each iterative step.

In both cases, the plots of the run time look as noise-like functions of the number of features m , because the computation process for different m required not only different number of iterations (31)-(32) but also different run time of the golden section algorithm (33) when adjusting the length of Newton’s step at each iteration.

Recall that that the learning process always converges to the strong minimum point of the regularized empirical risk criterion in a finite number of steps (Section 5.2).

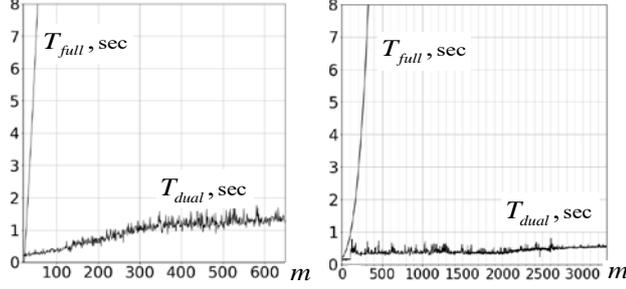


Figure 1. Chronometry of the learning process with growing number of features $m=1, \dots, n$.

Training set 1: Numerical regression – real-world data Training set 2: SVM pattern recognition – real-world data

7.4 Experimental selectivity path

In this experiment, we applied the rough selectivity path algorithm (Section 6) to the real-world Training set 1 (Section 7.2.1).

First of all, we found the maximal value of the selectivity parameter μ_0 (zero active features) in accordance with the technique outlined in Section 6.1. Since the respective dependence estimation problem is that of regression estimation, the truncated dual objective function (41) has the simple form on the force of (23) and (35):

$$\begin{cases} \sum_{j=1}^N ((1/2)\gamma\lambda_j^2 - y_j\lambda_j) \rightarrow \min, \\ \sum_{j=1}^N \lambda_j = 0. \end{cases}$$

The solution is the saddle point of the Lagrangian relative to the Lagrange multiplier η at the equality constraint:

$$L(\lambda_1, \dots, \lambda_N, \eta) = \sum_{j=1}^N ((1/2)\gamma\lambda_j^2 - y_j\lambda_j) - \eta \sum_{j=1}^N \lambda_j \rightarrow \begin{cases} \partial/\partial\lambda_1 = 0, \dots, \partial/\partial\lambda_N = 0, \\ \partial/\partial\eta = 0. \end{cases}$$

These conditions result in the system of $N+1$ linear equations:

$$\begin{pmatrix} \gamma\mathbf{I}_N & -\mathbf{1}_N \\ \mathbf{1}_N^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \eta \end{pmatrix} = \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix}, \text{ solution } (\lambda_1^* \dots \lambda_N^*) \in \mathbb{R}^N \text{ and idle number } \eta^* \in \mathbb{R}.$$

In accordance with (42), the solution determines the maximum reasonable selectivity, more precisely, the minimum selectivity that completely suppresses all the features when processing the given training set:

$$\mu_0 = \max_{i=1, \dots, n} \left(\sum_{j=1}^N \lambda_j^* \tilde{\alpha}_{\alpha, j, i} \right).$$

Finally, the condition (44) evaluates tentative gradually diminishing values of the selectivity parameter $\mu_0 > \mu_1 > \dots > \mu_{m-1} > \mu_m = 10^{-8}$. In our study, we accepted $m = 200$.

The experiment itself consists in consecutive application of the iterative algorithm (30)-(34), as in Section 6.2, starting for each μ_t with the previous solution $(\hat{\lambda}_{1, \mu_{t-1}}, \dots, \hat{\lambda}_{N, \mu_{t-1}})$. Since the Training set 1 relates to the regression estimation problem, solving the succession of dual problems boils down to solving the system of linear equations (36).

At each step, we registered the number of active features and the number of iterations. The result is shown in Figure 2.

The decrease of the number of active features is not surprising, it is just this what was to be expected, however, this process is not completely uniform. But the fact that the number of iterations at the slowly diminishing tentative values of the selectivity parameter remains minimally small is very notable. At most points of the selectivity axis the stopping condition (34) was achieved after one iteration, and only at a few points 2-3 iterations turned out to be required.

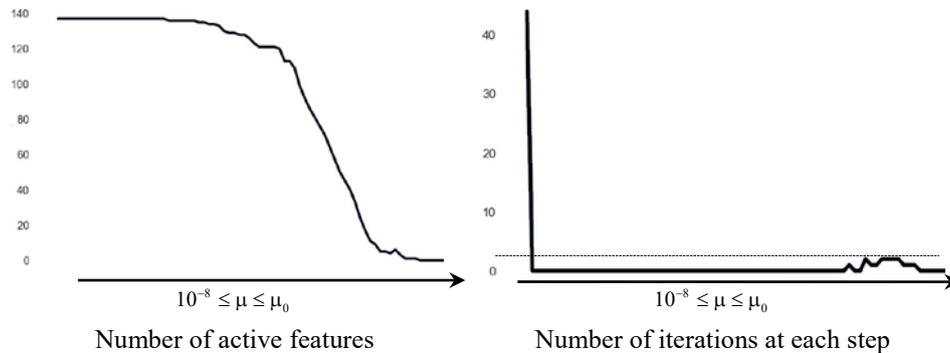


Figure 2. Rough regularization path through the selectivity axis

8 Conclusions

We have considered a class of generalized linear models of feature-based dependence estimation from empirical data, which covers, in particular, numerical regression and two-class SVM pattern recognition. Two additional assumptions, which are adequate to the overwhelming majority of practical applications, are that, first, the number of features n far exceeds that of objects in the training set N and, second, the features are tight interdependent, so that the effective dimension of the concentration ellipsoid of feature vectors is essentially smaller than the number of features.

Under some quite lenient assumptions, the traditional formulation of the generalized linear dependence estimation problem results in the convex problem of regularized empirical risk minimization. This problem inevitably has polynomial computational complexity in the number of features, what is in crucial conflict with the assumption on the huge dimension of the feature vectors $n \gg N$.

Therefore, we proposed an alternative disjoint formulation of the generalized linear dependence estimation problem, which allows for its numerical solution in two consecutive stages. First, a convex dual minimization problem of N variables is to be solved, which have the sense of Lagrange multipliers associated with the objects of the training set. Such a problem is of polynomial computational complexity relative to the assumingly modest size of the training set N . After that, it remains only to independently compute the estimates of the coefficients at the features in the generalized linear model of the sought-for dependence. It is clear that this procedure is not only of linear computational complexity in the number of features n , but also easily parallelizable.

9 Acknowledgement

We acknowledge the support from grants of the Russian Foundation for Basic Research 16-57-52042, 17-07-00436, 17-07-00993 and 18-07-01087.

References

1. Vapnik, V. Estimation of Dependences Based on Empirical Data. Springer-Verlag New York, 1982.
2. Nelder, J, Wedderburn, R. Generalized Linear Models. Journal of the Royal Statistical Society. Series A (General). Vol. 135, Issue 3, 1972, pp. 370-384.
3. McCullagh, P., Nelder, J. Generalized Linear Models, Second Edition. Chapman and Hall, 1989, 511 p.
4. Vapnik, V. Statistical Learning Theory. Wiley, 1998.

5. J. Fan, R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, December 2001, Vol. 96, No. 456, Theory and Methods, pp. 1348-1360.
6. <https://drive.google.com/open?id=1xhGgf7AXrKmSd3ZEIFus-48yHE8C6p8s>
7. A.N. Tikhonov. On the stability of inverse problems. *Doklady Akademii Nauk SSSR*, 1943, 39 (5), pp. 195–198. On the stability of inverse problems
8. Tikhonov A.N. Solution of incorrectly formulated problems and the regularization method. *Soviet Mathematics*, 1963, 4, pp. 1035–1038.
9. Tikhonov A.N., Arsenin V.Y. *Solution of Ill-posed Problems*. Washington: Winston & Sons, 1977.
10. Hoerl, A.E. Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 1962, 58, pp. 54-59.
11. A.E. Hoerl, R.W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, Vol. 12, No. 1, February 1970, pp. 55-67.
12. Zou, H., Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society*, 2005, Vol. 67, pp. 301-320.
13. D.W. Marquardt, R.D. Snee. Ridge Regression in Practice. *The American Statistician*, Vol. 29, No. 1, February 1975, pp. 3-20.
14. J. Fan, R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, December 2001, Vol. 96, No. 456, Theory and Methods, pp. 1348-1360.
15. Y. Nesterov. *Lectures on Convex Optimization*. Springer, 2018.
16. Bernstein A., Kuleshov A., Yanovich Y. (2015) Manifold learning in regression tasks. In: Gammerman A., Vovk V., Papadopoulos H. (eds). *Statistical Learning and Data Sciences*. LNCS, Vol. 9047. Springer, 2015, pp. pp 414-423.
17. O. Krasotkina, M. Markov, V. Mottl, D. Babichev, I. Pugach, A. Morozov. Constrained regularized regression model search in large sets of regressors. *MLDM in Pat. Rec. LNAI*, Vol. 1035, Springer, 2018, pp. 1-15.
18. R. Fletcher. *Practical Methods of Optimization*. Wiley, 2000, 450 p.
19. M. Park, T. Hastie. L1-Regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 2007, Vol. 69, Part 4, pp. 659–677.
20. J. Friedman, T. Hastie, R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journ. of Stat. Soft*, 2010, Vol. 33, Is. 1, pp. 1-22.
21. Sharpe W.F. Determining a fund's effective asset mix. *Investment Management Review*, September/October 1988.
22. Sharpe W.F. Asset allocation: Management style and performance measurement. *The Journal of Portfolio Management*, Winter 1992, pp. 7-19.
23. O. Krasotkina, M. Markov, V. Mottl, D. Babichev, I. Pugach, A. Morozov. Constrained Regularized Regression Model Search in Large Sets of Regressors. *MLDM 2018, New York, Part I. LNAI*, Vol. 1035, Springer, 2018, pp. 1-15.
24. M. Bucolo, L. Fortuna, M. Frasca. Robot control through brain-computer interface for pattern generation. *Complex Systems*, 2012, Vol. 20, Is. 3, pp. 243-251.
25. C.Hope, A. Sterr, P.E. Langovan, N.Geades, D.Windridge, K.Young, K.Wells. High Throughput Screening for Mammography using a Human-Computer Interface with Rapid Serial Visual Presentation (RSVP). *Proc. of SPIE 8673, Med. Imaging 2013: Image Perception, Observer Performance, and Technology Assessment*, 867303, 28 March 2013, 8 p.
26. V. Sulimova, S. Bukhonov, O. Krasotkina, V. Mottl, D. Windridge. Regularized SVMs for classification of image evoked EEG potentials captured from an observer. *MLDM 2019, New York, NY, USA, July 20-25, 2019*.

Authors Index

Alam	Ashraful			255
AlHamad	Ahmad	Qasim		1
Almugren	Nada			270
Alomari	Khaled	Mohammad		1
Alshamlan	Hala			270
Andrade	Luciano	C	M	201
Barone	Dante	A.	C.	28
Barone	Dante			382
Bhatele	Pushpraj			63
Braga	Dieinison	J.	F.	176
Braz S.S.	Leodecio			176
Bukhonov	Sergey			355
Carraher	Lee	A.		397
Carvalho	Andr'e	C P L F		201
Castro	Henrique	Carlos	de	28
Chang	Hyun-chul			296
Chen	Qing			367
Cisty	Milan			104
Coelho da Silva	Ticiana	L.		176
Coenen	Frans			187
Coleman	Sonya			392
Cyprich	Frantisek			104
Dahal	Animesh			319
de Almeida Lima	Marília	Nayara	Clemente	132
de Araújo Fagundes	Roberta	Andrade		132
de Faria	Elaine	Ribeiro		146
de Souza	Criston	Pereira		176
Dey	Sayantan			397
dos Santos	Wellington	Pinheiro		132
ElSherif	Hatem	M.		1
Faruque	Mohammad	O.		117
Farzana	Sheikh	Mastura		255
Ghahremannezhad	Hadi			117

Goodman	Eric	L.		161
Guarnizo	Jose	Guillermo		49
Guo	Terry			319
Gupta	Nidhi			63
Hatamizadeh	Ali			39
Hosseini	Hamid			39
Hudson	Corey			161
Jiang	Zhengyong			187
Kamareddine	Fairouz			78
Kerr	Dermot			392
Khan	Sharowar	Md.	Shahriar	255
Khanna	Pritee			63
Khanom	Aniqa	Zaida		255
Khomprasert	Adison			333
Krasotkina	Olga			355
Krasotkina	Olga			412
Krechel	Dirk			343
Krishnan	Mythili			229
Li	William			93
Lin	Xuan	Xiong		216
Lin	Xuan	Xiong		282
Lindsay	Leeanne			392
Liu	Chengjun			16
Liu	Zhengyuan			39
Liu	Chengjun			117
Maharana	Satyajeet			304
Marques	Bruno			382
Moorhead	Anne			392
Morozov	Alexey			412
Mottl	Vadim			355
Mottl	Vadim			412
Nascimento	Francielle	M.		28
Nino	Luis	Fernando		49
Pandey	Pradumn	Kumar		304
Park	Sungbum			296
Pugach	Ilya			412

Rahman	Tahsinur		255
Rakthamanon	Thanawin		333
Rao	T.	Ramalingeswara	304
Schwartz	Steven	D.	39
Shaalán	Khaled		1
Shi	Hang		16
Silva	Iago	Richard	Rodrigues 132
Siraj	Ambareen		319
Soldanova	Veronika		104
Srinivasan	Madhan	Kumar	229
Sterr	Annette		355
Stower	Kevin		343
Sulimova	Valentina		355
Sulimova	Valentina		412
Sun	Mingxuan		367
Tatarchuk	Alexander		412
Taylor	Brian		392
Terzopoulos	Demetri		39
Vieira	Eldane		146
Waiyamai	Kitsana		333
Wang	Xiaochun		216
Wang	Xia	Li	216
Wang	Aozhong		243
Wang	Xiaochun		243
Wang	Xia	Li	243
Wang	Xiaochun		282
Wang	Xia	Li	282
Wells	Kevin		355
Wilsey	Philip	A.	397
Windridge	David		355
Zhang	Jian		367
Zhao	Hua		78
Zhong	Junmei		93
Zimmerman	Chase		161

Announcement

World Congress DSA 2020

The Frontiers in Intelligent Data and Signal Analysis
July 12 - 23, 2020, New York, USA

www.worldcongressdsa.com

We are inviting you to our fourth World congress on the Frontiers of Signal and Image Analysis DSA 2020 to New York, Germany.

This congress will feature three events:

- the 16th International Conference on Machine Learning and Data Mining MLDM (www.mldm.de),
- the 20th Industrial Conference on Data Mining ICDM (www.data-mining-forum.de),
- and the 15th International Conference on Mass Data Analysis of Signals and Images in Artificial Intelligence&Pattern Recognition MDA-AI&PR (www.mda-signals.de).

Workshops and Tutorial will also be given.

Come to join us to the most exciting event on Intelligent Data and Signal Analysis.

Sincerely your,
Prof. Dr. Petra Perner

MLDM

www.mldm.de

icdm

www.data-mining-forum.de

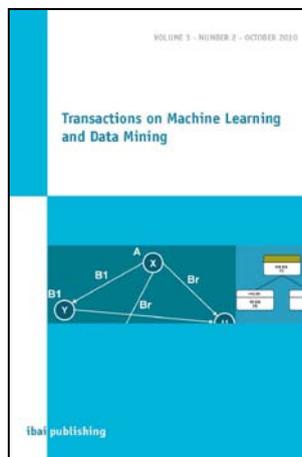
mda

www.mda-signals.de

Journals by ibai-publishing

The journals are free on-line journals but having in parallel hardcopies of the journals. The free on-line access to the content of the paper should ensure fast and easy access to new research developments for researchers all over the world. The hardcopy of the journal can be purchased by individuals, companies, and libraries.

Transactions on Machine Learning and Data Mining (ISSN: 1865-6781)



The International Journal "Transactions on Machine Learning and Data Mining" is a periodical appearing twice a year. The journal focuses on novel theoretical work for particular topics in Data Mining and applications on Data Mining.

Net Price (per issue): EURO 100
Germany (per issue): EURO 107 (incl. 7% VAT)

Submission for the journal should be send to:
info@ibai-publishing.org

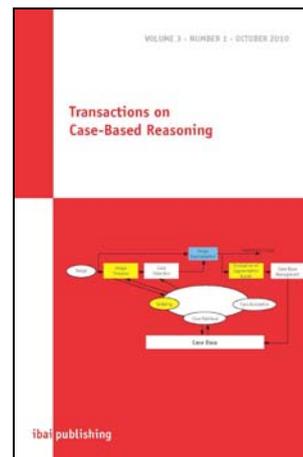
For more information visited: www.ibai-publishing.org/journal/mldm/about.html

Transactions on Case-Based Reasoning (ISSN:1867-366X)

The International Journal "Transactions on Case-Based Reasoning" is a periodical appearing once a year.

Net Price (per issue): EURO 100
Germany (per issue): EURO 107 (incl. 7% VAT)

Submission for the journal should be send to:
info@ibai-publishing.org

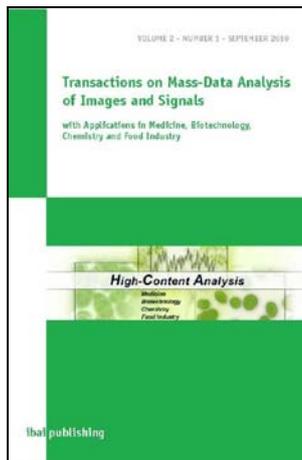


For more information visited: www.ibai-publishing.org/journal/cbr/about.html

Transactions on Mass-Data Analysis of Images and Signals (ISSN:1868-6451)

The International Journal "Transactions on Mass-Data Analysis of Images and Signals" is a periodical appearing once a year.

The automatic analysis of images and signals in medicine, biotechnology, and chemistry is a challenging and demanding field. Signal-producing procedures by microscopes, spectrometers and other sensors have found their way into wide fields of medicine, biotechnology, economy and environmental analysis. With this arises the problem of the automatic mass analysis of signal information. Signal-interpreting systems which generate automatically the desired target statements from the signals are therefore of compelling necessity. The continuation of mass analyses on the basis of the classical procedures leads to investments of proportions that are not feasible. New procedures and system architectures are therefore required.



Net Price (per issue): EURO 100

Germany (per issue): EURO 107 (incl. 7% VAT)

Submission for the journal should be send to:
info@ibai-publishing.org

For more information visited: www.ibai-publishing.org/journal/massdata/about.php