

Computer Science & Information Technology 170

Artificial Intelligence

David C. Wyld,
Dhinaharan Nagamalai (Eds

Computer Science & Information Technology

- 2nd International Conference of Education (CONEDU 2022), June 18~19, 2022, Sydney, Australia
- 8th International Conference on Computer Science, Information Technology and Applications (CSITA 2022)
- 3rd International Conference on Machine learning and Cloud Computing (MLCL 2022)
- 8th International Conference on Image and Signal Processing (ISPR 2022)
- 5th International Conference on Natural Language Processing and Trends (NATAP 2022)
- 8th International Conference on Artificial Intelligence (ARIN 2022)

Published By



AIRCC Publishing Corporation

Volume Editors

David C. Wyld,
Southeastern Louisiana University, USA
E-mail: David.Wyld@selu.edu

Dhinaharan Nagamalai (Eds),
Wireilla Net Solutions, Australia
E-mail: dhinthia@yahoo.com

ISSN: 2231 - 5403

ISBN: 978-1-925953-69-5

DOI: 10.5121/csit.2022.121001 - 10.5121/csit.2022.121014

This work is subject to copyright. All rights are reserved, whether whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the International Copyright Law and permission for use must always be obtained from Academy & Industry Research Collaboration Center. Violations are liable to prosecution under the International Copyright Law.

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

2nd International Conference of Education (CONEDU 2022), June 18~19, 2022, Sydney, Australia, 8th International Conference on Computer Science, Information Technology and Applications (CSITA 2022), 3rd International Conference on Machine learning and Cloud Computing (MLCL 2022), 8th International Conference on Image and Signal Processing (ISPR 2022), 5th International Conference on Natural Language Processing and Trends (NATAP 2022), 8th International Conference on Artificial Intelligence (ARIN 2022) was collocated with 8th International Conference on Artificial Intelligence (ARIN 2022). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The CONEDU 2022, CSITA 2022, MLCL 2022, ISPR 2022, NATAP 2022 and ARIN 2022. Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically.

In closing, CONEDU 2022, CSITA 2022, MLCL 2022, ISPR 2022, NATAP 2022 and ARIN 2022 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the CONEDU 2022, CSITA 2022, MLCL 2022, ISPR 2022, NATAP 2022 and ARIN 2022.

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

David C. Wyld,
Dhinaharan Nagamalai (Eds)

General Chair

David C. Wyld,
Dhinaharan Nagamalai (Eds)

Organization

Southeastern Louisiana University, USA
Wireilla Net Solutions, Australia

Program Committee Members

Aarthi Alamelu,
Abdel-Badeeh M. Salem,
Abdelhadi Assir,
Abderrahmane Ez-Zahou,
Abdessamad Belangour,
Abdullah,
Abdur Rasool,
Ajay Anil Gurjar,
Akhil Gupta,
Alejandro Garces,
Ali H. Wheeb,
Allel Hadjali,
Amal Azeroual,
Amel Ourici,
Amine Achouri,
Amir H Gandomi,
Ana Luísa Varani Leal,
Anand Nayyar,
Ankur Singh Bist,
António Abreu,
Archit Yajnik,
Aridj Mohamed,
Artyom Grigoryan,
Ashok Sutgaundar,
Ashraf Hossain,
Assem Abdel Hamied Moussa,
Assem mousa,
Ayyad Maafiri,
Azeddine Wahbi,
Bala Modi,
Bashir Ido,
Benyettou Mohammed,
Beshair Alsiddiq,
Bo Wei,
Bouhouch adil,
Brahmi Menaouer,
Brahim Lejdel,
Carlos Becker Westphall,
Chandrasekar Vuppapalapati,
Charalampos Karagiannidis,
Cheng Siong Chin,
Chi Zhou,
Chris Panagiotakopoulos,
Chrissanthi Angeli,

SRM Institute of Science and Technology, India
Ain Shams University, Egypt
Hassan 1st University, Morocco
Mohammed V University, Morocco
University Hassan, Morocco
Chandigarh University, India
Shenzhen Institute of Advanced Technology, China
Sipna College of Engineering and Technology, India
Lovely Professional University, India
Jaume I University, Spain
University of Baghdad, Iraq
LIAS/ENSMA, France
Mohammed V University, Morocco
Badji Mokhtar University of Annaba , Algeria
University of Tunis, Tunisia
University of Technology, Australia
University of Macau, China
Duy Tan University, Vietnam
AI Scientist at Signy Advanced Technology, India
ISEL – Polytechnic Institute of Lisbon, Portugal
Sikkim Manipal Institute of technology, India
Hassiba Benbouali University, Algeria
The University of Texas at San Antonio, USA
Basaveshwar Engineering College, India
National Institute of Technology (NIT), India
Chief Eng egyptair, Egypt
University of Bristol, England
Ibn Tofail University, Morocco
Hassan II University, Morocco
Gombe State University, Nigeria
Arsi University, Ethiopia
Univesity Center of Relizane, Algeria
Riyad Bank, Saudi Arabia
Northumbria University, UK
University Mohammed-V Rabat, Morocco
National Polytechnic School of Oran, Algeria
University of El-Oued, Algeria
Federal University of Santa Catarina, Brazil
San Jose State University, USA
University of Thessaly, Greece
Newcastle University, Singapore
Illinois Institute of Technology, USA
University of Patras, Greece
University of West Attica, Greece

Christian Mancas,
 Daniel Hunyadi,
 Dario Ferreira,
 Dariusz Jacek Jakóbczak,
 Debjani Chakraborty,
 Derya Malak,
 El Habib Nfaoui,
 El murabet Amina,
 Elena Pelican,
 Elżbieta Macioszek,
 Endre Pap,
 Faheem A. Khan,
 Fang Wang,
 Felix J. Garcia Clemente,
 Fernando Zacarias Flores,
 Fitri Utaminigrum,
 Francesco Zirilli,
 Friday Zinzendoff Okwonu,
 Froilan D. Mobo,
 G. Rajkumar,
 Gabriela Grosseck,
 Gajendra Sharma,
 Gang Wang,
 Garcia Clemente,
 Geraldo Pereira Rocha Filho,
 Giambattista Bufalino,
 Glaoui hachemi,
 Grigorios N. Beligiannis,
 Grzegorz Sierpiński,
 Guezouli Larbi,
 Guillermo E. Atkin,
 Gülден Köktürk,
 Hamid Ali Abed AL-Asadi,
 Hamidreza Rokhsati,
 Hatem Mohamed Abdelkader,
 Hatem Yazbek,
 Hedayat Omidvar,
 Hicham Gueddah,
 Hlaing Htake Khaung Tin,
 Iancu Mariana, Bioterra
 Ibrahim Hamzane,
 Ilango Velchamy,
 Isa Maleki,
 Islam Tharwat Abdel Halim,
 Israa Shaker Tawfic,
 Issa Atoum,
 Jagadeesh HS,
 Jakhongir Shaturaev,
 Janusz Kacprzyk,
 Jawad K. Ali,
 Jesuk Ko,
 José Alfredo F. Costa,

Ovidius University, Romania
 Lucian Blaga University of Sibiu, Romania
 University of Beira Interior, Portugal
 Koszalin University of Technology, Poland
 Indian Institute of Technology, India
 University of Minnesota, UK
 Sidi Mohamed Ben Abdellah University, Morocco
 Abdelmalek Essaadi University, Morocco
 Ovidius University of Constanta, Romania
 Silesian University of Technology, Poland
 Singidunum University Belgrade, Serbia
 University of Huddersfield, UK
 Wilfrid Laurier University, Canada
 University of Murcia, Spain
 Universidad Autonoma de Puebla, Mexico
 Brawijaya University, Indonesia
 Sapienza Università Roma, Italy
 Universiti Utara Malaysia, Malaysia
 Philippine Merchant Marine Academy, Philippines
 Department of Computer Applications, India
 West University of Timisoara, Romania
 Kathmandu University, Nepal
 University of Connecticut, USA
 University of Murcia, Spain
 University of Brasília, Brazil
 University of Catania, Italy
 Tahri Mohammed University, Algeria
 University of Patras, Greece
 Silesian University of Technology, Poland
 Higher National School of Renewable Energy, Algeria
 Illinois Institute of Technology, USA
 Dokuz Eylül University, Turkey
 Iraq University college, Iraq
 Sapienza University of Rome, Italy
 Menofia university, Egypt
 Broadcom Inc., Israel
 Research & Technology Dept, Iran
 Mohammed V University, Morocco
 University of Computer Studies, Myanmar
 University of Bucharest, Romania
 Hassan II University of Casablanca, Morocco
 CMR Institute of Technology, India
 Science and Research Branch, Iran
 Nile University, Egypt
 Ministry of Migration and Displaced, Iraq
 The World Islamic Sciences and Education, Jordan
 APS College of Engineering (VTU), India
 Tashkent State University, Uzbekistan
 Systeme Research Institute, Poland
 University of Technology, Iraq
 Universidad Mayor de San Andres (UMSA), Bolivia
 Federal University, UFRN, Brazil

Joydev Ghosh,	National Research Tomsk Polytechnic, Russia
Jun Hu,	Harbin University of Science and Technology, China
Karim El Moutaouakil,	FPT/USMBA, Morocco
Keneilwe Zuva,	University of Botswana, Botswana
Khang Lam,	Can Tho University, Vietnam
Khelifi mustapha,	Tahri Mohammed Bechar University, Algeria
Kiril Alexiev,	Bulgarian Academy of Sciences, Bulgaria
Krzysztof Kulpa,	Warsaw University of Technology, Poland
Lan Truong,	University of Cambridge, UK
Lin Cai,	Illinois Institute of Technology, USA
Ljubomir Lazic,	Belgrade Union University, Serbia
Loc Nguyen,	Loc Nguyen's Academic Network, Vietnam
Luisa Maria Arvide Cambra,	University of Almeria, Spain
Mahmoud R. Delavar,	University of Tehran, Iran
Manal Abdulaziz Abdullah,	King Abdulaziz University, Saudi Arabia
Manish Kumar Mishra,	University of the People, USA
Mansour,	University Salah Boubenider, Algeria
Manuel Jesús Cobo Martín,	Universidad de Cádiz, Spain
Mario Versaci,	Associate Professor - Electrical Engineering, Italy
Marius Cioca,	Lucian Blaga University of Sibiu, Romania
Masoomah Mirrashid,	Semnan University, Iran
Michail Kalogiannakis,	University of Crete, Greece
Micheline Al Harrack,	Marymount University, USA
Mohammad A. M. Abushariah,	The University of Jordan, Jordan
Mohammad Jafarabad,	Qom University, Iran
Mohammad Talib,	University of Botswana, Botswana
Mohd Norazmi bin Nordin,	Universiti Kebangsaan Malaysia, Malaysia
Mounir Zrigui,	University of Grenoble-Alpes, France
Mu-Chun Su,	National Central University, Taiwan
Müge Karadağ,	İnönü University, Türkiye
Muhammad Naveed Anwar,	Northumbria University, UK
Mu-Song Chen,	Da-Yeh University, Taiwan
MV Ramana Murthy,	Osmania University, India
Nadia Abd-Elasabour,	Cairo university, Egypt
Nadine Akkari,	Lebanese university, Lebanon
Naren.J,	Senior Faculty, Karnataka, India
Natheer K Gharaibeh,	Taibah University, Saudi Arabia
Naziah Abd Kadir,	Universiti Selangor, Malaysia
Nidal M. Turab,	Al-Isra University, Jordan
Nikola Ivković,	University of Zagreb, Croatia
Nur Eilayah Wong,	Senior Lecturer/ Researcher, Malaysia
Oleksii K. Tyshchenko,	University of Ostrava, Czech Republic
Oliver L. Iliev,	FON University, Republic of Macedonia, Macedonian
Omid Mahdi Ebadati,	Kharazmi University, Tehran
P.V.Siva Kumar,	VNR VJIT, India
Paul Bogdan,	University of Southern California, USA
Petra Perner,	Futurelab Artificial Intelligence IBal-2, Germany
Pietro Guccione,	Politecnico di Bari, Italy
Pokkuluri Kiran Sree,	Sri Vishnu Engineering College for Women, India
Priyanka Srivastava,	Banaras Hindu University, India
Przemyslaw Falkowski-Gilski,	Gdansk University of Technology, Poland
R. S. Balagadde,	Kampala international University, Uganda

Raghad Ghalib Alsultan,	Northern Technical University, Iraq
Rajeev Kanth,	University of Turku, Finland
Ramadan Elaiees,	University of Benghazi, Libya
Ramana Murthy,	Osmania University, India
Ramgopal Kashyap,	Amity University Chhattisgarh, India
Rodrigo Pérez Fernández,	Universidad Politécnica de Madrid, Spain
Ruhaidah Samsudin,	Universiti Teknologi Malaysia, Malaysia
Rung-Ching Chen,	Chaoyang University of Technology, Taiwan
Saad Al Janabi,	Al- hikma college university, Iraq
Sabyasachi Pramanik,	Haldia Institute of Technology, India
Sahar Saoud,	Ibn Zohr University, Morocco
Sahil Verma,	Chandigarh University, India
Said Nouh,	Hassan II university of Casablanca, Morocco
Saikumar Tara,	CMR Technical Campus, Hyderabad
Sami Ahmed Haider,	University of Worcester, UK
Samir Kumar Bandyopadhyay,	University of Calcutta, India
Santosh Kumar Bharti,	Pandit Deendayal Energy University, India
Sathyendra Bhat J,	St Joseph Engineering College, India
Sean Mc Grath,	University of Limerick, Ireland
Selenge,Mongolian,	University of Pharmaceutical Sciences, Mongolia
Shah Alam,	University of Rajshahi, Bangladesh
Shahid Ali,	Manukau Institute of Technology, New Zealand
Shahram Babaie,	Islamic Azad University, Iran
Shashikant Patil,	ViMEET Khalapur Raigad MS India
Shing-Tai Pan,	National University of Kaohsiung, Taiwan
Siarry Patrick,	Universite Paris-Est Creteil, France
Sidi Mohammed Meriah,	University of Tlemcen, Algeria
Sikandar Ali,	University of Haripur, Pakistan
Smain Femmam,	UHA University, France
Sofiane Bououden,	University Abbes Laghrour Khenchela, Algeria
Soumya Sen,	Seacom Engineering College, India
Stefano Cirillo,	University of Salerno, Italy
Subhendu Kumar Pani,	Krupajal Group of Institution, India
Suhad Faisal Behadili,	University of Baghdad, Iraq
T V Rajini Kanth,	SNIST, India
Tasher Ali Sheikh,	Residential Girls' Polytechnic, India
Tran Cong Manh,	Le Quy Don Technical University, Vietnam
Tsung-Jung Liu,	National Chung Hsing University, Taiwan
Uranchimeg Tudevdagva,	Chemnitz University of Technology, Germany
V.Ilango,	CMR Institute of Technology, India
Vyacheslav Tuzlukov,	Belarussian State Aviation Academy, Belarus
Xiao-Zhi Gao,	University of Eastern Finland, Finland
Yas Alsultanny,	Arabian Gulf University, Bahrain
Yazid Basthomi,	Universitas Negeri Malang, Indonesia
Yousef Farhaoui,	Moulay Ismail University, Morocco
Yousfi Abdellah,	University Mohamed V, Morocco
Youssef Fakir,	Sultan Moulay Slimane University, Morocco
Yuan-Kai Wang,	Fu Jen Catholic University, Taiwan
Zayar Aung,	National Research University, Russia
Zeljen Trpovski,	University of Novi Sad, Serbia
Zeynep Yucel,	Okayama University, Japan
Zoran Bojkovic,	University of Belgrade,Serbia

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)



Artificial Intelligence Community (AIC)



Soft Computing Community (SCC)



Digital Signal & Image Processing Community (DSIPC)



2nd International Conference of Education (CONEDU 2022)

Video Content Development Guides based on Teaching Experiences.....01-18
Zolzaya Badamjav and Uranchimeg Tudevdaeva

**The Opportunities and Challenges of Learning Online During the Pandemic:
Thai High School Students' Perspective.....19-30**
Miss Pichsinee Oimpitiwong

8th International Conference on Computer Science, Information Technology and Applications (CSITA 2022)

**Media Legitimacy Detection: A Data Science Approach to Locate Falsehoods
And Bias using Supervised Machine Learning and Natural-Language
Processing.....31-41**
Nathan Ji and Yu Sun

Enhancing Networking Cipher Algorithms with Natural Language.....43-54
John E. Ortega

3rd International Conference on Machine learning and Cloud Computing (MLCL 2022)

**An Intelligent Sensor Mobile Phone Assisting System using AI and Machine
Learning.....167-178**
Ruilang Liang and Yu Sun

8th International Conference on Image and Signal Processing (ISPR 2022)

**AI_Birder: An Intelligent Mobile Application to Automate Bird Classification
using Artificial Intelligence and Deep Learning.....55-66**
Charles Tian and Yu Sun

5th International Conference on Natural Language Processing and Trends (NATAP 2022)

**An Intelligent Mobile Application for Depression Relief using Artificial
Intelligence and Natural Language Processing.....67-81**
Zhishuo Zhang, Yu Sun and Ryan Yan

**A Context-Aware Vocabulary Management and Reading Assistance System
using Machine Learning and Natural Language Processing.....83-93**
Zhanhao Cao and Yu Sun

8th International Conference on Artificial Intelligence (ARIN 2022)

Deep Multiple Instance Learning for Forecasting Stock Trends using Financial News.....	95-111
<i>Yiqi DENG and Siu Ming YIU</i>	
CalixBoost: A Stock Market Index Predictor using Gradient Boosting Machines Ensemble.....	113-128
<i>Jarrett Yeo Shan Wei and Yeo Chai Kiat</i>	
An Introductory Review of Spiking Neural Network and Artificial Neural Network: From Biological Intelligence to Artificial Intelligence.....	129-145
<i>Shengjie Zheng, Lang Qian, Pingsheng Li, Chenggang He, Xiaoqi Qin and Xiaojian Li</i>	
An Intelligent News-based Stock Pricing Prediction using AI and Natural Language Processing.....	147-155
<i>Sirui Liu and Yu Sun</i>	
Identifying a Default of Credit Card Clients by using a LSTM Method: A Case Study.....	157-166
<i>Jui-Yu Wu and Pei-Ci Liu</i>	
Transformer based Ensemble Learning to Hate Speech Detection Leveraging Sentiment and Emotion Knowledge Sharing.....	179-194
<i>Prashant Kapil and Asif Ekbal</i>	

VIDEO CONTENT DEVELOPMENT GUIDES BASED ON TEACHING EXPERIENCES

Zolzaya Badamjav¹ and Uranchimeg Tudevdaeva²

¹Department of Didactics,
Mongolian National University of Education, Ulaanbaatar, Mongolia

²Faculty of Computer Science, Chemnitz University of Technology,
Chemnitz, Germany

ABSTRACT

This paper describes a research study on video content development. Due to the COVID-19 pandemic, all kinds of education and training switched from traditional classroom teaching to online and distance learning. The effect of e-learning will be the integral part of the higher education's primary structure. The challenge of online teaching in higher education is to prepare learning materials for students with corresponding quality in various types. The video contents are one of important type of teaching and learning materials. This is one of most welcomed learning materials by students during online and distance teaching. Advantages of video contents are easy to follow focus of lesson, can hear and watch simultaneously, or just can hear if want, or just can watch if not possible to hear, more realistic, gives feeling like takes lesson in classroom. But, to prepare video contents requests a lot of time and preparation. It needs corresponding skills from teacher and it is costly. To support video content with high quality can be offer well defined guidance which helps to prepare good video contents. In this study authors are explained experience-oriented guidance for video content development.

KEYWORDS

Online teaching, distance teaching, higher education, quality of video lessons, SURE model, evaluation.

1. INTRODUCTION

In a UNESCO document on the theory and methodology of e-learning: online teaching is the way of acquiring knowledge and skills using tools that are mostly based on web and computer devices. It may take place in or out of the classroom. It can be referred as a counterpart of distant learning that may be engaged in an online platform used for educational purposes [1].

The process of using a set of technological tools which have a function of sharing or exchanging any kind of information with an intention of enhancing student's knowledge and skills creates the foundation of the technology enhanced learning. It allows students to discover the relevant information for their studies using diverse electronic resources. Integration of learners with web-based applications gives access of engaging through online platforms and enables exchange of information. Using the relevant electronic tool as a teaching methodology gives access to improve the training experience and makes it more creative which is one of the advantages of digital learning concept [2].

E-learning involves four main components such as: a person who delivers the information or skills, a person who receives it, a technological device and, data that is shared during this process. This has become a substantial part of learning for students as it can be achieved from any distance and time [3].

The main difference of e-learning from the distance learning is the use of electronic device and internet. In other words, it can be referred as a channel that delivers information from one point to another numerous destinations at any time. In the beginning of introduction of digital learning, people did not fully acknowledge it because most of the part of this process includes machinery which made an impression that it has a deficiency in human contribution in education system [4].

Because of the existence of coronavirus pandemic, a mass gathering is strongly restricted in many areas of the globe. This has led educational institutions to make relevant changes into the standard education system making it more digitalized. The most effective method of proceeding teaching and learning steps is e-learning because it can be achieved from anywhere and at any time. However, there are obstacles and statistical indicators, that show the low level of effectiveness in gaining new information and skills [5].

Adekola and Dale (2017) show in the study that E-learning involves two main elements and six components as shown in the Figure 1 [6]. In the first element of this framework, the author referred three counterparts related with engagement of learners with the technology. In the second dimension, there are three components which involve activities concerning learning and processing information and skills [7].

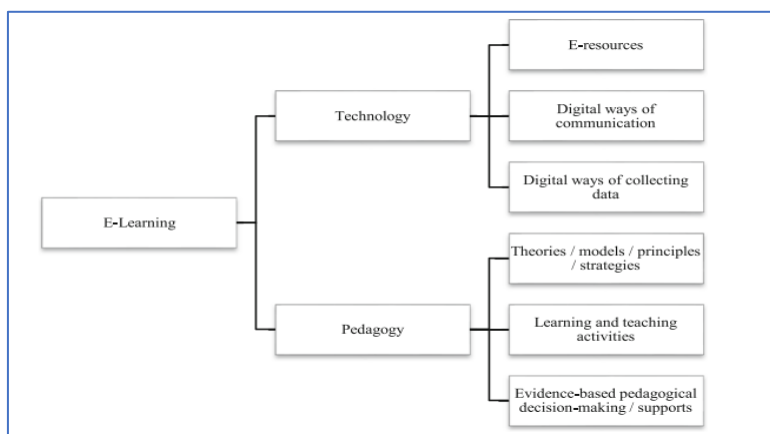


Figure 1. An e-Learning framework in school education consisting of two dimensions and six components

When it comes to enhancement of learning and verification of the performance, assessment is one of the primary tools for educational institutions [8].

Assessment that can be proceeded through digital environment is called e-assessment [9]. This can enable various opportunities which makes it more distinguish than the traditional assessment process. The digitalization of assessment can simplify the scoring process making it more time-saving. On the other hand, with the help of e-assessment, teachers can gather essential data from the students works [10] and can get a guaranteed answers and marks that will confirm the final score[11].

Digitalization and automatization have brought a considerably massive changes in our thoughts of how education system can be reformed. However, there are still a lack of deeper

understandings of assessment in this regard. Hence, the majority is discussing about the assessment that comprehends genuine assessment and encourages the users [12-13].

Numerous institutions are currently studying the diverse technology-based applications that use personal data for more reliable e-assessments[14-18]. Regarding the outcomes, e-assessment offers more time-saving and rational options and makes the process more innovating[19]. However, when using e-assessment, there is a small chance of avoiding fraud which makes it weaker. To prevent such issues, professionals from the TeSLA corporation (<https://tesla-project.eu>) created a tool that can identify face and voice of an individual, any information that is considered as a transcript, etc [20].

One of the integral parts of higher education is a concept of video. It is often used both in traditional and e-learning environments as a main source for sharing an essential information. A number of studies have presented the statistical proofs of the effectiveness of e-learning [21-22] and other works also have shown the importance of the use of video in teaching and learning methodology. [23-28].

2. STATE OF THE ART

The video content is one of the important teaching elements for e-learning. There are many different tips around how to create video content for teaching. Vyond Studio defined 25 different hints to create video content[29]. These hints consist of five different dimensions: Planning, Writing, Storyboarding, Creating and Distributing. Each dimension included sub sections where explained what should care during preparation of video content. Gretchen Vierstra shared learned lessons and advantages of video lessons [30]. Gretchen Vierstra recommended to care “working memory” of students, that means video lessons should be short. Next hints from her is the cognitive load which should support students who cannot concentrate on different focuses. Further hints directed to care various ways to access the information for students. Patrick Lowenthal with his colleagues noted that one of the methods to engage students during e-learning is video-based discussions [31].

Teaching and learning during COVID-19 pandemic requested to try different engagement methods to keep student's motivation to study. The synchronous video-based discussions can give opportunities to students feel in social presence together with class mates and offers chance to illustrate and demonstrate how to solve learning problems [32]. Rahmatika in cooperation with his colleagues published the paper about effectiveness of Youtube videos as one of the useful video contents for teaching and learning [33]. Main focus of Rahmatika is how can be fill missing gap of teacher's skill to work tools for video contents. Due to the COVID-19 each single teacher must be teaching his/her course online, from distance. But not all teachers had corresponding skills to work with video tools. Usually quality of prepared video contents could not meet expected results of students and parents. To solve in some case this problem his team offered to use free Youtube videos for teaching. B.D. Collier and M.J. Scott described in their paper about teaching case of mechanical engineering courses based on video games [34].

As a result of their study found out that students spend twice more hours on video game with learning material comparison with traditional learning materials. In case of Peter D. Wiens and his co-authors study focus directed to application of video content to assessment of teacher education [35]. Zanelidin E. et. al. used as teaching tool video contents for engineering courses and did survey to investigate the effectiveness, benefits, and students' satisfaction of students [36]. In total 67 students attended to survey and survey result shows that students prefer to receive learning materials in video content and study in blended type of study.

3. GUIDANCE FOR VIDEO CONTENT DEVELOPMENT

Why do we need a videocontent on this topic? How to make a video? How to reach your students? What is your experience? How to evaluate the video content? will answer the question. The quality of video content for teaching depends a lot from preparation and development phases. Not all teachers have enough experiences and skills to work with video contents. But today's teaching materials for students requests to include various type of teaching tools and methods for student's engagements. One of basic method for this is to use video content in teaching. To support teachers who has not big experience with video content our team developed special guidance for video content development. The guidance consists of three phases.

3.1. Pre-phase

Not everything should be taught through video, so be strategic about why you're using the medium to deliver a particular topic [37]. Therefore,first and important phase of preparation of video content is Pre-phase.

What can be done to help you reach your goals? think about and prepare video content. For example, what is the student's learning style? How to choose the content of the topic? What elements (text, pictures, diagrams, graphs, tables, sounds, recordings, animations, illustrations) should be used to introduce the concept? etc. Didactic solutions for teaching methods need to be considered and planned in order to achieve and achieve the objectives of the lesson.

At this stage, video content planning should be done according to the following instructions.

- *Research and analysis of course materials*

Study and analyse video content course materials. This may include the syllabus, pictures, tables, diagrams, audio, and additional videos used in the lesson. Course files can be placed on cloud-based devices such as google drive, one drive, and Dropbox for sharing. For example, in our experiment, we used Figure 2 to upload the file to Google Drive and OneDrive.

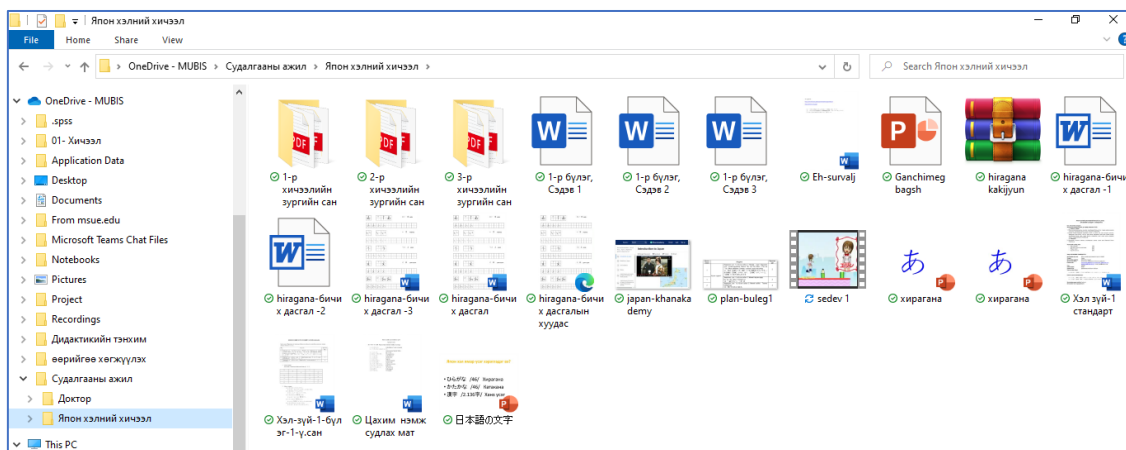


Figure 2. File library located on OneDrive

- *Plan which elements to use. Make a storyboard*

Plan a storyboard before developing a video content. In other words, plan the script for the screenplay. For example, plan the script for the screenplay. A storyboard is a series of images and symbols that show the background image of each frame (slide for ppt), the main scene, appearance, who and where to do, what elements and information to use, and what activities take place. In addition to drawing the storyboard by hand on paper, you can design it using a program that suits you. The following example shows a detailed Storyboard design for each screen (Figure 3 and 4).

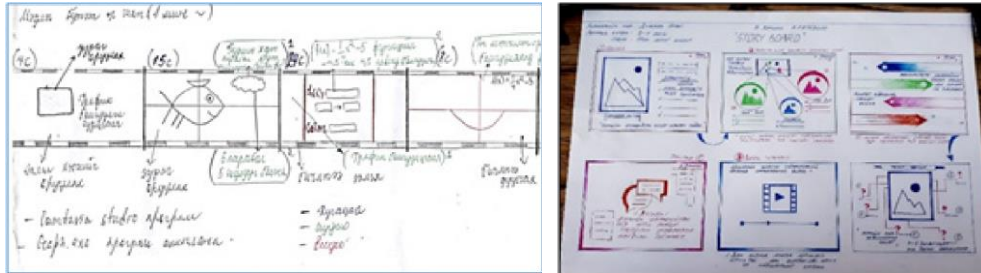


Figure 3. An example of a storyboard



Figure 4. The process of developing a storyboard

If you prefer to hand-draw your storyboard, you don't need to draw boxes for every frame use this free template [29]. When designing a storyboard, select the type of video using the video tutorial templates in Figures 5 and 6.

Hansch [38] also provide a catalogue of video production styles 'as a method of providing a current overview of the field'. Their division is based on what they refer to as the production style's 'different affordances of learning'. The production styles they define can also be combined in various forms. They list the following 18 production styles:

3.2. Development phase

After preparation of above defined elements man can start to develop video contents.

- *Select technology and develop materials*

Video contents can be made professionally, in the studio, or in a quiet environment using simple software. Table 1 shows the possibilities of working with video product development software information. In addition, video editing software can be used online. For example: <https://www.loom.com>, <https://www.vidyard.com>

Table 1. Ability to work with video processing software information

No	Activity	Power Point	Moviemaker	Camtasia	iSpring suite	SnagIt	Adobe Premier
1	Write text	+	+	+	+	+	+
2	Upload a photo	+	+	+	+	+	+
3	Insert audio	+	+	+	+	+	+
4	Audio editing			+			+
5	Upload video	+	+	+	+	+	+
6	Video editing		+	+			+
7	Insert shapes	+			+	+	+
8	Insert effects	+	+	+	+	+	+
9	Record Audio...	+	+	+	+	+	+
10	Screen Recording	+		+		+	
11	Add Quiz/Survey			+	+		
12	Save in video format	+	+	+	+	+	+

3.3. Evaluation phase

Ready video contents should go through local evaluation. For evaluation of prepared video contents team applied structure-oriented evaluation SURE model [39].

This method is called SURE (StrUcture oRIented Evaluation) and is designed to contribute to the space for evaluating e-learning with a science-based methodology (Figure 7). The model was first discussed at its first meeting in 2011 to assess e-learning in a multi-dimensional space with the participation of all stakeholders. One of the features of this methodology was that the goals of the evaluation were defined by the main and sub-goals, and then these goals were expressed in a logical scheme [40-42].

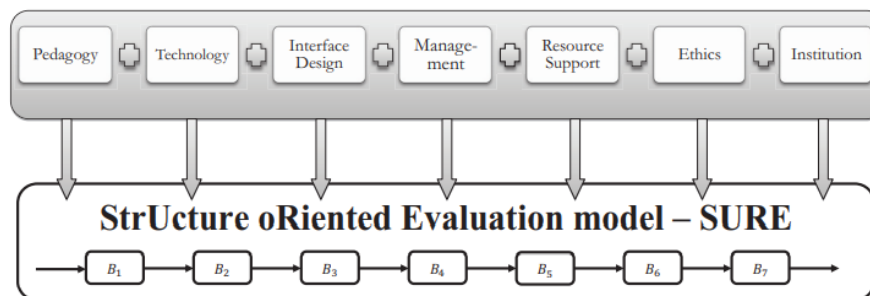


Figure 7. Evaluation and e-learning framework [42]

Step 1. Definition of key goals

The main aim of the first step of the SURE model is to define key goals of evaluation process. All defined key goals should reach its target and the general evaluation result will be successful or bigger than zero only if all key goals reached their target successfully. If one of the key goals is failed, then the e-learning process will be evaluated as failed. The main goals of the evaluation:

- Planning (B_1)
- Technologists (B_2)
- Contents (B_3)
- Performance (B_4) dependence, implementation

consists of goals(Figure8).



Figure 8. Goal structures of video content

Step 2. Definition of sub goals

Determining the steps that need to be taken to achieve the main goals is the step that defines the sub-goals.

In order for Planning (B_1) to be successful, the following sub-goals must be met. These include:

- Course content planning (A_{11})
- Lesson material planning (A_{12})
- Lesson methodological planning (A_{13})
- Screen design (A_{14})
- Course file compilation (A_{15})

The following sub-goals must be met in order for Technology (B_2) to achieve its primary objectives. These include:

- Environment (A_{21})
- Video (A_{22})
- Speaker's voice (A_{23})
- Accompanying sound (A_{24})

Content (B_3) in order to achieve the main goal, the following sub-goals must be met. These include:

- Topic content (A_{31})
- Object (A_{32})
- Record type (A_{33})
- Information Ethics (A_{34})
- Spelling and grammar (A_{35})
- Didactic solution (A_{36})

Performance (B_4) in order to achieve the main goal, the following sub-goals must be met. These include:

- Video content Troubleshooting (A_{41})
- Video content distribution (A_{42})

Figure 9 shows the structure of the sub-goals, which consists of five, four, six, and two sub-objectives.

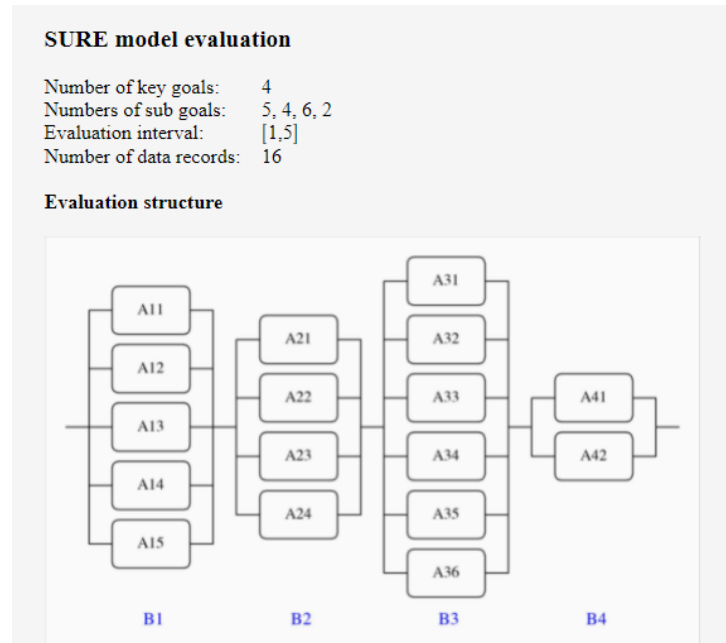


Figure 9. Evaluation sub-goalsstructure

Step 3. Confirmation of evaluation goalstructures

We defined four goals as key goals. These key goals should reach its target all together. This is essential important for goal structure definition. First key goal consists of five, second key goal consists of four, third key goal consists of six and fourth key goal consists of two sub goals. Difference of sub goals from key goal is if just one of sub goal is evaluated as successful the corresponding key goal will be receiving successful evaluation score.

The purpose of this evaluation was presented and approved by the student who developed the video content, the student who was the teacher, the student who developed the SURE methodology, and the leaders of the video content development teams.

Step 4. The checklist of evaluation

The checklist of evaluation should create based on defined goal structures of the SURE model. Basically, sub goal definition can become fundamental for checklist questions. In Table 2 show example questions which developed for key goals Planning and Technology.

Table 2. Development of questions related to the main goalsof Planning (B₁) and Technology (B₂)

		Corresponding points	5	4	3	2	1
		Evaluation indicators	Strongly agree	Agree	Don't know	Disagree	Strongly disagree
Planning (B ₁)	1	The content of the lesson is designed in a simple, clear and interesting way					
	2	The course materials are designed to fully express the content, and the type and format appropriate to the video is selected.					
	3	The teaching methodology is designed to be motivating and engaging					
	4	Each frame was clearly mapped, each element was formulated in detail, and explanatory notes were made					
	5	The file directory and resources used in the lesson are well established					
Technology (B ₂)	1	The video should be played on any player and on devices such as a mobile phone or tablet					
	2	The video is well timed, uninterrupted, non-vibrating, and recorded normally.(Not more than 6 minutes)					
	3	The teacher's speech and explanations are recorded clearly and audibly					
	4	Accompanying sounds and audios are written in accordance with the content					

The question is developed in the form of a final definition, and the data are collected by measuring the extent to which the evaluator agrees with the definition. Here:

- Strongly agree - 5 point
- Agree - 4point
- Don't know - 3point
- Disagree - 2point
- Strongly disagree - 1point

Step 5. Acceptance of checklist

The checklist should control by all involved groups in evaluation process and only checked and confirmed checklist should be apply for data collection. Clear formulation of questions is only one aspect of checklist. Each member of the evaluation team has to check that before confirmation and, if necessary, appearance and design of checklist has to be improved.

Step 6. Data collection

There are several techniques for data collection: surveys and questionnaire, tests and assessments, interviews, focus groups, action plans, case studies, and performance records [43]. Evaluation team can use any of these techniques. Most attractive and fashion technique is online checklist. Data collection via online checklist increases quality of collected data avoiding human error during paper-based data collection.

In this study, we used the Google form free online survey form (Figure 10).

Figure 10. Questionnaire example in Google form

Step 7. Data processing

Another advantage of the SURE method is the online data processing calculator for the method [44]. The online calculation program starts with the main window called SURE model evaluation.

- Enter checklist data - enter the collected data in the appropriate format
- In the Color scale type section, select from 4 color options to adjust the color of the result table.
- Checklist data display format –collected data can be show in result table in three different view
- In the Evaluation table - select the format in which the SURE evaluation results will be displayed

Figure 11. The upper part of the data calculation program

Step 8. The evaluation reports

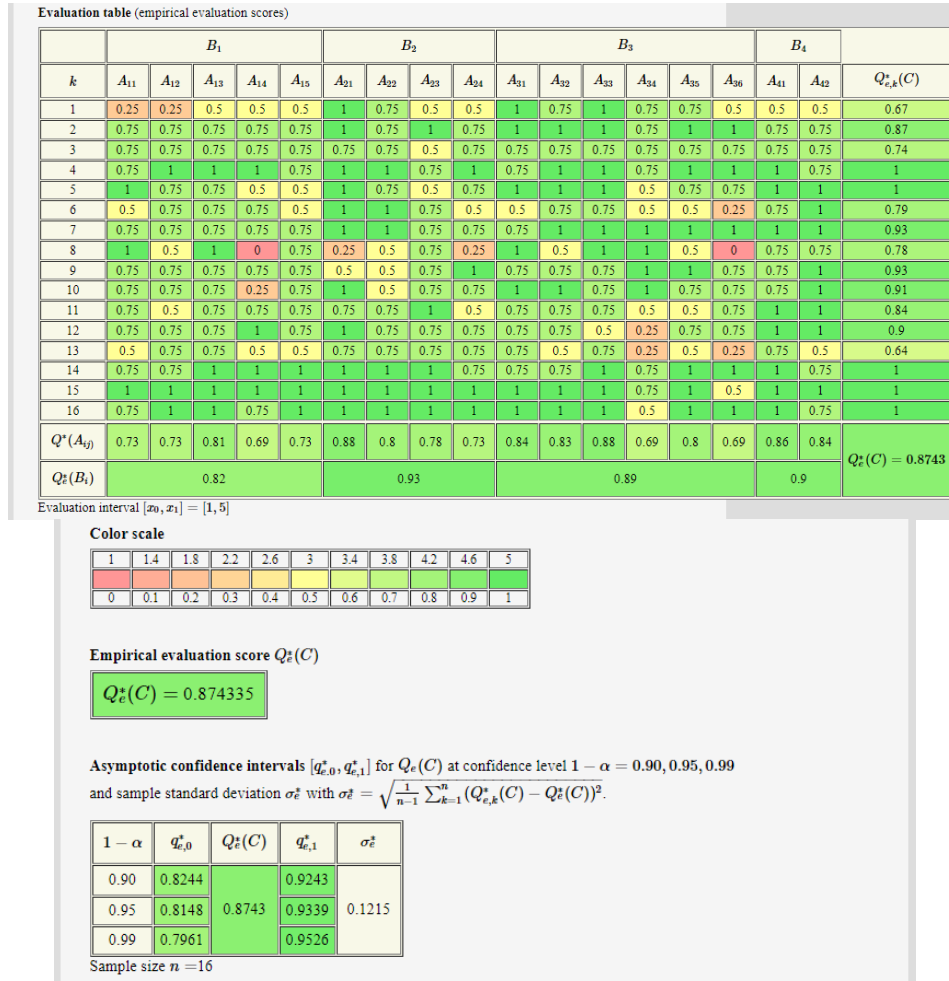


Figure 12. The evaluation reports of video content

The SURE model delivered four different scores after data processing.

- First one is general evaluation score.

- $Q_e^*(C) = 0.87$

The result of SURE data processing shows general evaluation score as 0.87. If transfer it to percent (maximum is 100%) the evaluation result can be explained as the 87% succeeded by responses data.

- Second result is scores for key goals.

- $Q_e^*(B_1) = 0.82;$
 - $Q_e^*(B_2) = 0.93;$
 - $Q_e^*(B_3) = 0.8;$
 - $Q_e^*(B_4) = 0.9.$

These scores show how good reached targets for key goals. Best score received the fourth key goal and worst one is third key goal. But all scores are reached its target as over 80%.

- Third result is scores for sub goals.

- $Q^*(A_{11}) = 0.73;$
- $Q^*(A_{12}) = 0.73;$
- $Q^*(A_{13}) = 0.81;$
- $Q^*(A_{14}) = 0.69;$
- $Q^*(A_{15}) = 0.73;$
- $Q^*(A_{21}) = 0.88;$
- $Q^*(A_{22}) = 0.8;$
- $Q^*(A_{23}) = 0.78;$
- $Q^*(A_{24}) = 0.73;$
- $Q^*(A_{31}) = 0.84;$
- $Q^*(A_{32}) = 0.83;$
- $Q^*(A_{33}) = 0.88;$
- $Q^*(A_{34}) = 0.69;$
- $Q^*(A_{35}) = 0.8;$
- $Q^*(A_{36}) = 0.69;$
- $Q^*(A_{41}) = 0.86;$
- $Q^*(A_{42}) = 0.84.$

These scores are show success of sub goals. Best scores received the sub goals A_{21} and A_{33} . Worst scores are linked to sub goals A_{34} and A_{36} .

- Fourth result is evaluation score of each participant in the evaluation process.

- $k_1 = 0.67;$
- $k_2 = 0.87;$
- $k_3 = 0.74;$
- $k_4 = 1;$
- $k_5 = 1;$
- $k_6 = 0.79;$
- $k_7 = 0.93;$
- $k_8 = 0.78;$
- $k_9 = 0.93;$
- $k_{10} = 0.91;$
- $k_{11} = 0.84;$
- $k_{12} = 0.9;$
- $k_{13} = 0.64;$
- $k_{14} = 1;$
- $k_{15} = 1;$
- $k_{16} = 1;$

These scores are show evaluation result of each student. In total 16 students are taking part of this evaluation process. 5 of 16 students evaluated of video content with maximum score 1. Two students evaluated with most worst scores: 0.64 and 0.67.

4. APPLICATION EXAMPLE

The offering guidance was applied to teaching process. Students (in future teachers) prepared video contents following giving guidance. The 3rd semester students of the Informatics class were divided into 5 groups and made video content on a specific topic.

Table 3. Video content topics

Group number and video content topics
Group 1: Data, information and knowledge
Group2: Multimedia
Group 3: Database
Group 4: Turtle Graph
Group 5: Types of information systems

Teams created video content on selected topics. The purpose of the topic was defined according to the instructions and a storyboard was created (Figure 13).

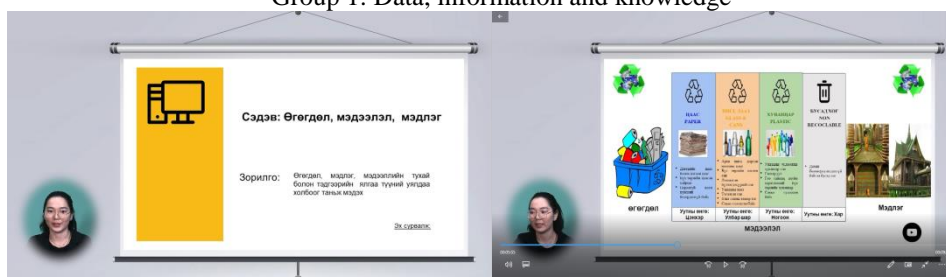


Figure 13. Introducing the storyboard

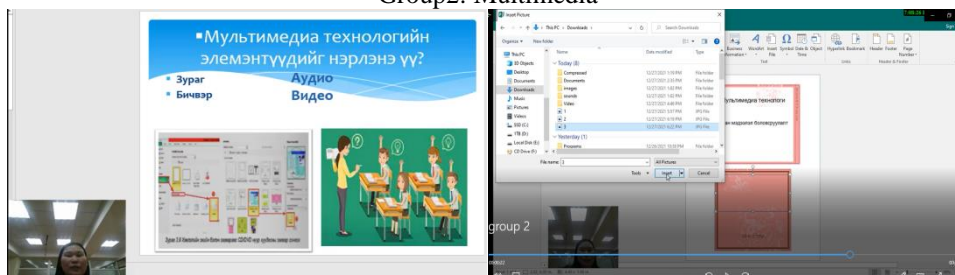
The video content was selected from the Typology of Video Production Styles (Figure 5). The developed video content was presented to other teams during the lesson and evaluated on its own and independently (Figure 13).

The video content of the teams is shown below.

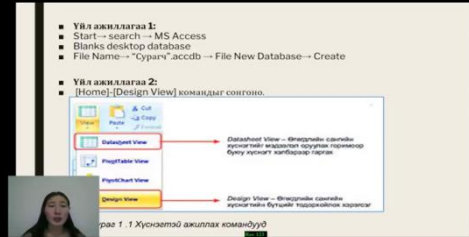
Group 1: Data, information and knowledge



Group2: Multimedia



Group 3: Database



УЙЛАЖИЛАГАА 1:
 ■ Start → search → MS Access
 ■ Blanks desktop database
 ■ File Name → "Company" .accdb → File New Database → Create

УЙЛАЖИЛАГАА 2:
 ■ [Home] [Design View] команды сонгоно.

Database View – Өгөгдлийн сангийн үзвэлтэй холбоотой оруулалт, гаргалт, бүтэц, үзвэлтэй холбоотой сонголт.

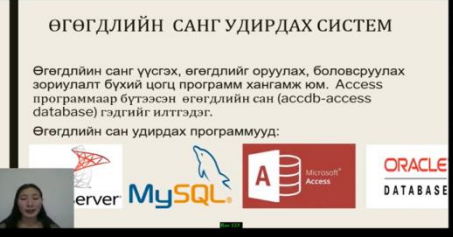
Design View – Өгөгдлийн сангийн үзвэлтэй холбоотой оруулалт, гаргалт, бүтэц, үзвэлтэй холбоотой сонголт.

Хүснэгт 1.1 Хүснэгтэй ажиллах командыуд

ӨГӨГДЛИЙН САНГ УДИРДАХ СИСТЕМ

Өгөгдлийн санг үүсгэх, өгөгдлийг оруулах, боловсруулах зориулалт бүхий цогц программ хангамж юм. Access программаар бүтээсэн өгөгдлийн сан (accdbe-access database) гэдгийг илтгэдэг.

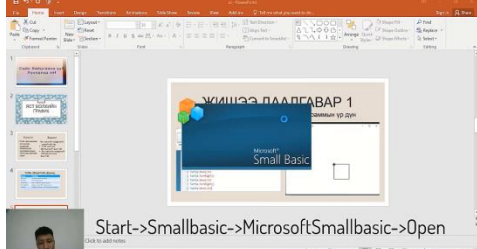
Өгөгдлийн сан удирдах программууд:



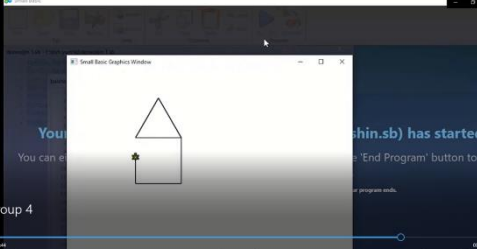
Access программ руу яаж орох вэ

Аравдугаар ангийн хүрээнд MS Access Программыг судалж

Group 4: Turtle Graph



Start → Smallbasic → Microsoft Smallbasic → Open



Group 5: Types of information systems

ХИЧЭЭЛИЙН ДЭГ

Багшийн яриаг анхааралтай сонсох

Ойлгохгүй үлдсэн зүйлээ тэмдэглэн Багшаас лавлан асууна.

Бие даан ажиллах



ЯАГААД БҮРТГЭЛ БОЛОВСРУУЛАЛТЫН СИСТЕМ ХЭРЭГЛЭХ ШААРДЛАГАТАЙ ВЭ?



5. CONCLUSIONS

During a pandemic, there is a need for a blended learning activities. Therefore, the study of video content development methodology is considered to be timely. This paper suggests that video content can be developed according to guidelines in the preparation, development, and evaluation phases, and that the video content can be evaluated using the SURE assessment model.

Students work with small groups to develop video content, introduce it to each other, and share experiences. The video content was evaluated according to a checklist developed by the teacher, and the results were analyzed using the SURE assessment model. The assessment is done twice. It can improve the quality of video content and create good video content.

Using this guidance, teachers can gain sufficient experience and skills to work with video content.

REFERENCES

- [1] UNESCO, I. (2013) *Glossary of Curriculum Terminology*. Geneva: (UNESCO-IBE).
- [2] Seel, N. M. (2012) *Encyclopedia of the sciences of learning*. London-New York: Springer.
- [3] Selviandro, N., & Hasibuan, Z. A. (2013) *Cloud-Based E-Learning: A Proposed Model and Benefits by Using E-Learning Based on Cloud Computing for Educational Institution*. In: Mustofa K., Neuhold E.J., Tjoa A.M., Weippl E., You I. (eds) *Information and Communication Technology. ICT-EurAsia 2013. Lecture Notes in Computer Science*, vol 7804. Springer. Berlin, Heidelberg. doi:https://doi.org/10.1007/978-3-642-36818-9_20

- [4] The economic times. (2022) Retrieved from <https://economictimes.indiatimes.com/definition/e-learning>
- [5] Lizcano, D., Lara, J. A., & White, B. e. (2020) Blockchain-based approach to create a model of trust in open and ubiquitous higher education. *Journal of Computing in Higher Education*, 32, pp109–134. doi:<https://doi.org/10.1007/s12528-019-09209-y>
- [6] Adekola, J., & Dale, V. (2019) *Development of an institutional framework to guide transitions into enhanced blended learning in higher education*. Research in Learning Technology, 25, 16. doi:<https://doi.org/10.25304/rlt.v25.1973>
- [7] Kong, S. C. (2021) *Delivery and evaluation of an e-Learning framework through computer-aided analysis of learners' reflection text in a teacher development course*. RPTEL 16, 28. doi:<https://doi.org/10.1186/s41039-021-00172-w>
- [8] Clements, M. D., & Cord, B. A. (2013) *Assessment guiding learning: developing graduate qualities in an experiential learning programme*. Assessment and Evaluation in Higher Education, 38(1), pp114–124.
- [9] NC. (2010) *Transitioning to Online Assessment in North Carolina*. The North Carolina State Board of Education, NC.
- [10] Ripley, M. (2009) *Transformational Computer-based Testing*. In: Scheuermann, F., Björnsson, J. (eds.), Reykjavik, Iceland.
- [11] Conole, G., & Warburton, B. (2011) *A review of computer-assisted assessment*. ALT-J 13(1), pp17–31.
- [12] Mora, M. C., Sancho-Bru, J. L., Iserte, J. L., & Sánchez, F. T. (2012) *An e-assessment approach for evaluation in engineering overcrowded groups*. Computers and Education, 59, pp732–740.
- [13] Mueller, J. (2014) *Authentic assessment toolbox*. North Central College, Naperville. Retrieved from <http://jfmuellet.faculty.noctrl.edu/toolbox/whydoit.htm>
- [14] Gaytan, J., & McEwen, B. C. (2007) “Effective online instructional and assessment strategies”, *American Journal of Distance Education*(21(3)), pp117–132. doi:<https://doi.org/10.1080/08923640701341653>.
- [15] Jones, D. R. (2011) Academic dishonesty: Are more students cheating? *Business Communication Quarterly*, 74(2), pp141–150. doi:<https://doi.org/10.1177/1080569911404059>.
- [16] McCann, A. L. (2010) Factors affecting the adoption of an e-assessment system. *Assessment & Evaluation in Higher Education*, 35(7), pp799–818. doi:<https://doi.org/10.1080/02602930902981139>
- [17] Noguera, I., Guerrero-Roldan, A. E., & Rodríguez, M. E. (2017) Assuring authorship and authentication across the e-assessment process. In *Proceedings of the Technology Enhanced Assessment, TEA 2016* (pp. 86–92). doi:<https://doi.org/10.1007/978-3-3>
- [18] Underwood, J., & Szabo, A. (2003) Academic offences and e-learning: Individual propensities in cheating. *British Journal of Educational Technology*, 34(4), 467–478. doi:<https://doi.org/10.1111/1467-8535.00343>.
- [19] Jisc. (1993) *Joint information systems committee*. Retrieved from <https://www.jisc.ac.uk>
- [20] Muravyeva, E., Janssen, J., Dirkx, K., & Specht, M. (2019) Students' attitudes towards personal data sharing in the context of e-assessment: Informed consent or privacy paradox? In *Proceedings of the 2018 International Technology Enhanced Assessment Conference, TEA 2018*, (pp. 16–26). Amsterdam, The Netherlands.
- [21] Means B, T. Y. (2010) *Evaluation of Evidence-Based Practices in Online Learning: Meta-Analysis and Review of Online Learning Studies*. Washington, DC: US Department of Education.
- [22] Schmid RF, B. R. (2014) *The effects of technology use in postsecondary education: a meta-analysis of classroom applications*. Comput Educ.
- [23] Allen W, A. S. (2012) *Effects of video podcasting on psychomotor and cognitive performance, attitudes and study behavior of student physical therapists*. Innov Educ Teach Int.
- [24] Kay R, H. (2012) *Exploring the use of video podcasts in education: a comprehensive review of the literature*. Comput Human Behav.
- [25] Lloyd S, A. R. (2012) *Screencast tutorials enhance student learning of statistics*. Teach Psychol.
- [26] Rackaway C. (2012) *Video killed the textbook star? Use of multimedia supplements to enhance student learning*. J Pol Sci Educ.
- [27] Hsin W, J. C. (2013) *Short videos improve student learning in online education*. J Comput Sci Coll.
- [28] Stockwell B, R. S. (2015) *Blended learning improves science education*. Cell.
- [29] Vyond Team, (2021), “How to Make an Instructional Video: 25 Essential Tips”, Online available: <https://www.vyond.com/resources/25-tips-create-engaging-instructional-videos/>

- [30] Gretchen Vierstra, (2022) "Teacher videos: 5 reasons why making your own videos can help with distance learning", Online available: <https://www.understood.org/articles/en/teacher-videos-5-reasons-why-making-your-own-videos-can-help-with-distance>
- [31] Patrick Lowenthal, Richard West, Leanna Archambault and Jered Borup, (2020) "Engaging Students Through Asynchronous Video-Based Discussions in Online Courses", Online available: <https://er.educause.edu/articles/2020/8/engaging-students-through-asynchronous-video-based-discussions-in-online-courses>
- [32] Peter Fadde and Phu Vu, (2014) "Blended Online Learning: Benefits, Challenges, and Misconceptions," in *Online Learning: Common Misconceptions, Benefits and Challenges*, eds. Patrick R. Lowenthal, Cindy York, and Jennifer Richardson (Hauppauge, NY: Nova Science Publishers, 2014), pp33–48.
- [33] Rahmatika, Munawir Yusuf, Leo Agung, (2021) "The Effectiveness of Youtube as an Online Learning Media", *Journal of Education Technology*. Vol. 5(1) pp152-158.
- [34] B.D. Collier, M.J. Scott, (2009) "Effectiveness of using a video game to teach a course in mechanical engineering", *Computers & Education*, Volume 53, Issue 3, 2009, Pages 900-912, ISSN 0360-1315, <https://doi.org/10.1016/j.compedu.2009.05.012>.
- [35] Peter D. Wiens, Kevin Hessberg, Jennifer LoCasale-Crouch, Jamie DeCoster, (2013) "Using a standardized video-based assessment in a university teacher education program to examine preservice teacher's knowledge related to effective teaching", *Teaching and Teacher Education*, Volume 33, 2013, Pages 24-33, ISSN 0742-051X, <https://doi.org/10.1016/j.tate.2013.01.010>.
- [36] Zanelidin E, Ahmed W, El-Ariss B. (2019) Video-based e-learning for an undergraduate engineering course. *E-Learning and Digital Media*.16(6):475-496. doi:10.1177/2042753019870938
- [37] Sarah McKibben (2014) Showing Videos in the Classroom: What's the Purpose? Education Update newsletter. ASCD.
- [38] Hansch, A., Newman, C., Hillers, L., Shildhauer, T., McConachie, K., & Schmidt, P. (2015) Video and online learning: Critical reflections and findings from the field. Retrieved from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2577882
- [39] UranchimegTudevtagva: Structure Oriented Evaluation Model for E-Learning. Wissenschaftliche Schriftenreihe Eingebettete Selbstorganisierende Systeme, Universitätsverlag Chemnitz, Chemnitz, Germany, July 2014. 123 p., ISBN: 978-3-944640-20-4, ISSN: 2196-3932.
- [40] UranchimegTudevtagva, Wolfram Hardt, A new evaluation model for eLearning programs, Technical Report CSR-11-03. Chemnitz, 2011.
- [41] UranchimegTudevtagva, W. Hardt, E. Tsoy, and M. Grif, New Approach for E-Learning Evaluation. In *Proceedings of the 7th International Forum on Strategic Technology 2012*, September 17-21, 2012, Tomsk, Russia, pp712-715.
- [42] UranchimegTudevtagva, and W. Hardt, A measure theoretical evaluation model for e-learning programs. In *Proceedings of the IADIS on e-Society 2012*, March 10-13, 2012, Berlin, Germany, pp.44-52.
- [43] Phillips, P. P. et al., (2010) *ASTD Handbook of Measuring and evaluating training*. Alexandria, VA: ASTD.
- [44] UranchimegTudevtagva, 2020 "Structure-Oriented Evaluation an Evaluation Approach for Complex Processes and Systems", Gewerbestrasse 11, 6330 Cham, Switzerland, Springer, pp92, ISBN 978-3-030-44805-9 ISBN 978-3-030-44806-6 (eBook), <https://doi.org/10.1007/978-3-030-44806-6>.

AUTHORS**Uranchimeg Tudevdagva**

Prof. Dr. Dr. h. c. Uranchimeg Tudevdagva, Guest Professor of Faculty for Computer Science at Chemnitz University of Technology. Professor of Mongolian University of Science and Technology. Prof. Tudevdagva is an expert on evaluation model for complex systems and e-learning.



Zolzaya Badamjav Lecturer of Department of Didactics, Mongolian National University of Education.



THE OPPORTUNITIES AND CHALLENGES OF LEARNING ONLINE DURING THE PANDEMIC: THAI HIGH SCHOOL STUDENTS' PERSPECTIVE

Miss Pichsinee Oimpitiwong

Triam Udom Suksa School, Bangkok, Thailand

ABSTRACT

This paper investigates students' online learning experience during COVID-19, specifically aiming to identify points of improvement within the current distance-learning infrastructure in Thailand. The research consolidates students' opinions toward online learning, their ease in adapting to the new learning environment, which depends not only on each student's learning style but also on their teachers as well as social and economic factors. Identifying the advantages and disadvantages of learning from home, the research presents students' needs and suggestions for improvement. As such, this work may guide future adjustments to online learning.

KEYWORDS

Learning Online, Students, Factors, Advantages, Disadvantages.

1. INTRODUCTION

Nobody anticipated that the beginning of a new decade would put humanity and the world through one of the most significant tests in recent decades. In 2019, a global pandemic originating from Wuhan, China, called the Coronavirus (COVID-19), spread worldwide. In 2020, the Ministry of Public Health (MoPH) in Thailand reported the first infected patient found in the country.

According to various medical journals, this infectious disease spreads from the patient's respiratory tract. Consequently, Thailand has since been on multiple lockdowns, as the government promotes social distancing at all times. Even so, cases have continued to rise negatively impacting the health, economy as well as education of the community.

However, education is a crucial foundation of every nation, directly impacting the future of the younger generations. Many countries have contemplated using innovative technology as part of an effective educational system. It is at this perfect moment that we see the greatest purpose and function of the internet and various learning communications.

As a student myself, I can see the changes and continued adjustments in the teaching and learning process during this pandemic. Studying online from home is one of the main solutions deployed to reduce widespread infection. Using online platforms is not new in Thailand. In 1995, King Bhumibol Adulyadej started the Distance Learning Television program to provide fundamental

education specifically for students living in remote areas. Nevertheless, country-wide online learning still requires both teachers and students to quickly adapt in major ways. Many students still prefer face-to-face learning while others are more willing to accept and make adjustments due to the pandemic, finding ways to learn more effectively through the screen from electronic devices instead of using a blackboard or whiteboard. This study investigates and reports on high school students' perspectives regarding the benefits and drawbacks of online learning. The resulting report can be used as a reference on how to address the problems of online learning.

This research aims to understand students' perspectives toward online learning and use the information as a guide to improve knowledge online learning. The broader goal of this research is to advocate for the need of schools as well as the Thai Ministry of Education to seek and reflect on teachers and students' feedbacks when it comes to online learning, whether it be the need for teacher training in creating online curriculum and using online applications or helping students to learn more effectively online during the COVID-19 pandemic.

2. REVIEW OF RELATED LITERATURE

“Following the pandemic-induced, sudden shift to online learning in Thailand, Hiranrithikorn [1] cautioned that online learning, while beneficial in reducing health risks during the pandemic, may be inaccessible to many students.” Furthermore, interviewing managers of small businesses, academics and policymakers, the study concluded the major advantage to online learning to be the additional amount of time students have as commuting and various in-school activities have been eliminated. “On the other hand, drawbacks included a lack of social connection between students and teachers and insufficient learning time [1].”

Even prior to the COVID-19 pandemic, the rise of online learning in recent years has led to various studies on its advantages and disadvantages. The primary advantage of online learning over the traditional method lies in the use of technology to aid learning. “Alghizzawi et al. [2] suggested that the ability to conveniently share digital content among teachers and students in online education led to more effective learning.” Moreover, the use of online forums could also encourage more effective interaction and discussion. “However, Gilbert [3] stressed the importance of face-to-face social interactions in students' learning satisfaction.” The study showed that most students preferred to have more in-person interactions with their peers and teachers than their online curriculum allowed. Students reported their online classes to consist mainly of assigned independent work with limited help from their teachers and discussion with their peers. Emailing teachers for help and using discussion forums often involved longer response-time than ideal for most students, resulting in an interrupted flow of learning Gilbert [3].”

“Nevertheless, Dumford and Miller [4] proposed that online education had more potential to serve students with different learning styles, compared to traditional classrooms.” Online learning platforms with all their features may be more suited to interactive exercises, the use of animations, or videos. While the ability to conveniently present subject matter in a more diverse manner will benefit learners, certain online-learning limitations may disadvantage some students. Learning outcome evaluation is one area in which online classrooms face complications. “Yilmaz [5] reported two main methods of evaluation: online exams with multiple choice or short answer questions and project assignments. Both forms of evaluation may be problematic for different reasons.” “As online exams were more difficult to monitor for cheating and plagiarism, teachers had limited ability to accurately assess students' learning outcomes Arkorful and Abaidoo [6].” “Even with project assignments, Sarraf, “Al-Shihi & Rehman [7] found students who struggled with organizational skills to perform poorer, as they failed to allocate the appropriate amount of time for each task.” Such students may benefit more from the traditional classroom setting where

teachers are able to assess their progress more closely in person, as well as provide help with organization and motivation.” “This notion was confirmed by Sarkar [8], who found students with lower self-motivation to be less successful in their online learning.”

While many disadvantages to online learning such as reduced in-person interaction and increased reliance on students' self-motivation are inevitable, its advantages are more open to possibilities depending on the design of the online program. “Gautam and Tiwari [9] named five areas of consideration in designing an effective online program: audience, course structure, page design, content engagement, and usability.” When done correctly, the authors argued that online learning could allow students to study interactively, according to their individual needs and pacing, as well as give them the confidence to self-regulate and organize their learning. As the pandemic necessitates online education, evaluation of current online curriculums must be done to ensure that arising problems are being addressed and potential advantages are being pursued.

3. METHODOLOGY

To gain insight into the perspectives of Thai high-school students on the advantages and disadvantages of online education during the pandemic, an online survey followed by one-on-one interviews was conducted from May to June of 2021. Participants consist of 100 Thai high-school students from Bangkok and nearby provinces. In Thailand, the high school covers grades 7 to 12 and is divided into two levels: lower high school (grade 7-9) and upper high school (grade 10-12). Figure 1 shows the percentage of participants by grade. At 43%, the majority of the participants were in grade 9, followed by grade 10 (28%), 11 (10%), and 7 (9%). The remaining 10% was from grades 8 and 12.

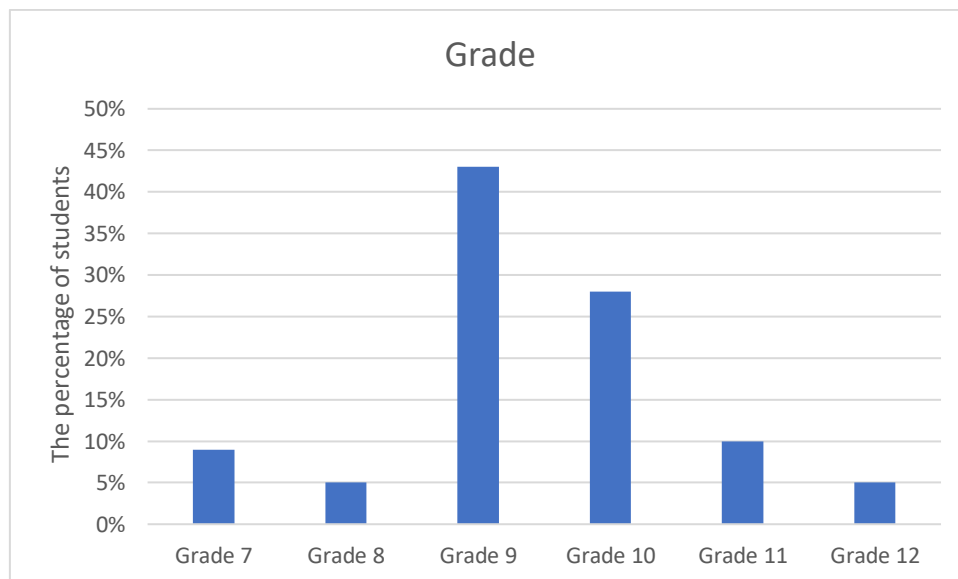


Figure 1. Percentage of participants by grade level

To investigate the impact of online learning on students from different academic concentrations, also known as "curriculums" in Thai high schools, participants were also asked to specify their curriculums. As shown in Figure 2, students in the Science-Mathematics curriculum (Sci-Math) make up 76.4% of the participants, followed by those in the Art-Language curriculum (Art-Lang) at 15.3%. The remaining 8.3% comprised those enrolled in the Art-Mathematics, Art-Science,

and Art-Computer curriculums, which are similar in course structure and will be collectively referred to as the Art-STEM curriculum in this study.

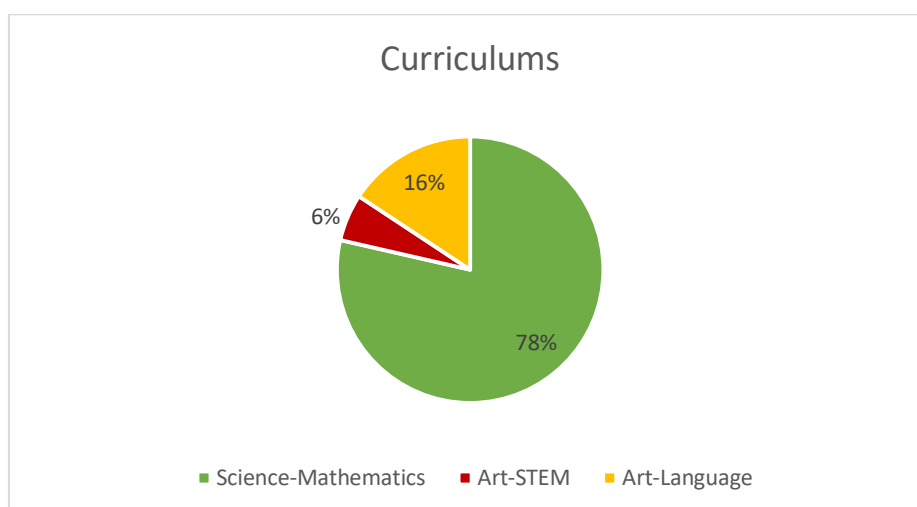


Figure 2. Percentage of participants by curriculum

The online survey consisted of 23 multiple-choice and seven short-answer questions. Multiple-choice questions were used to collect the following data; participants' grade level and curriculum, feelings towards various aspects of the online classroom, and after-class activities. Participants answered closed questions on whether or not they experienced the following five common problems in their online curriculums; assignment overload, schedule overload, attention, comprehension, and health problems, which they then elaborated on in the short-answer questions. Assignment and schedule overload refers to the excessive number of assignments given and the overly packed online class schedule respectively. Attention and comprehension problems refer respectively to the difficulty in focusing during class and in understanding the subject matter. Health problem encompasses the decline in mental and/or physical health as a result of online learning. Upon submission of online survey responses, one-on-one interviews were conducted to allow participants to discuss in greater depth the benefits and drawbacks they had experienced so far in online learning. Lastly, their recommendations on how to improve online education were noted.

Data collected from the online survey were computed into bar and pie charts, using Google Forms and Microsoft Excel graphing functions. Interview notes were compiled and analyzed for common themes among the reported problems and improvement recommendations.

4. RESULTS AND FINDINGS

4.1. Students' perspective on the advantages of online learning

During their interviews, most students named the convenience of studying from home without having to commute as one of the first advantages of online learning. Some students invested the time gained in hobbies that helped them destress from the packed schedule. A few students even reported turning their hobbies into small enterprises, some selling baked goods online while others picked up online work from home to help earn some income for their household during COVID-19 which has been financially difficult for many families.

In terms of academics, more time at home allowed some students to review and better prepare for each class. Certain time-consuming daily rituals such as morning assemblies had been replaced by each student prepping individually in front of their devices for their day of learning ahead. Some also benefited from the freedom of choosing their own time and place of study, if such options were available for their school's online learning program. Even though flexible class schedules are rare, many students still highly appreciate being able to learn in a more flexible manner, being free to move about, wearing their clothes of choice, or even doing something as inconsequential as snacking during class.

Some students proposed that online learning had resulted in improved student-teacher interactions. Pre-recorded lessons offered by some teachers allowed students the added flexibility of learning at their own pace. Students also preferred digital assignment submission to the less eco-friendly and inconvenient use of physical copies. Moreover, student-teacher communications became much more direct and efficient than pre-COVID where emailing teachers was uncommon in Thailand. In class, students were now encouraged to communicate with their teachers much more than before, making use of features like emoji buttons or virtual hand-raising to participate actively.

4.2. Common challenges faced by students learning online

The online survey revealed five main challenges students faced in their online learning, namely assignment overload, schedule overload, attention, comprehension, and health problems. Moreover, the prevalence of these problems varied across the three academic curriculums students is divided into starting in grade 10, namely Science-Mathematics (Sci-Math), Art-STEM, and Art-Language (Art-Lang). Figure 3 shows the percentage of grade 10-12 students in each curriculum who experienced the five common problems aforementioned. The blue bar represents all students while the green, red and yellow bars represent those enrolled in the Sci-Math, Art-STEM, and Art-Lang curricula respectively.

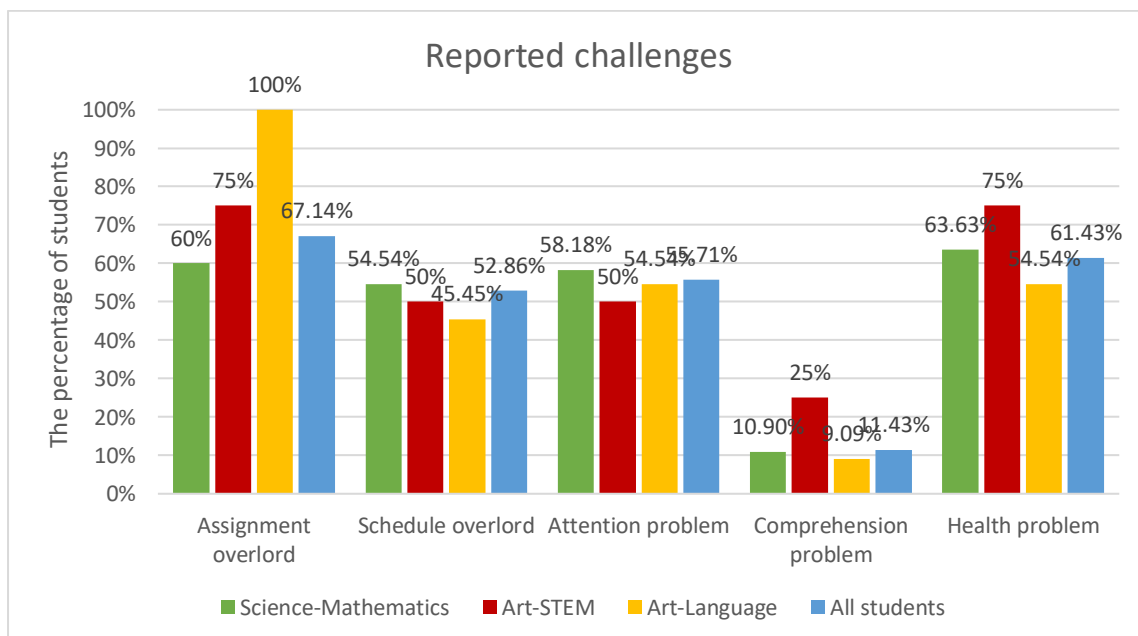


Figure 3. The percentage of grade 10-12 students experiencing five common online-learning problems

As shown by the blue bars in Figure 3, the most prevalent online-learning problem was assignment overload, experienced by 67.14% of students across all curriculums. This result was consistent with interview findings in which the majority of students complained of increased assignment load compared to learning in school. As the elimination of in-school activities and daily commute led to more free time, individual teachers were eager to fill it with assignments. However, the student argued that the lack of communication between teachers led to an uncoordinated increase in assignment load from all subjects, leaving students feeling overwhelmed without adequate daily rest. As a result, issues like backaches, eye problems, and stress began to deter students from their usual study habits. Seemingly simple tasks such as staring at their computer screens for eight hours daily had proven to be challenging for many. As such, it is unsurprising that mental and physical health issues would follow as the second most prevalent problem affecting 61.43% of the students studying online.

While some students shared that they were able to concentrate on their online lessons due to their passion for learning and high level of self-regulation, these were the minority. Figure 3 shows attention problems to affect more than half of the students at 55.71%. Many reported difficulties focusing while learning online since they were forced to do so from their homes. According to the online survey conducted, 51.5% of students deemed their home environments unsuitable for learning.

As seen in Figure 3, schedule overload was reported to be a problem for 52.86% of the students. During their interviews, students were asked to consider which subjects should be added or removed for online learning. First and foremost, students agreed that no subject must be added to the schedule because the existing curriculum was already too time and energy-intensive. Most agreed that subjects like physical education must be removed for the time being, as students cannot reap full benefits from PE lessons through learning theories online as opposed to actually exercising and practicing skill sets in different sports in traditional PE classes. This same argument was extended to other subjects requiring hands-on learning such as health education and music.

Figure 3 shows that comprehension problem was the least prevalent, affecting only 11.43% of students in grades 10-12. This surprisingly low percentage may reflect the fact that grade 10-12 students had gone through a selection process, evaluating their aptitude for their chosen curriculum. This selection process is particularly intensive for the most well-established and popular curriculums for Thai students, Science-Mathematics and Art-Language, both having fewer students with comprehension problems at 10.90% and 9.09% respectively.

In contrast, 53.3% of unspecialized students in grades 7-9 reported difficulties grasping the material being taught online, naming teachers' style of instruction as the main culprit. Students stated that some teachers went through material too quickly, perhaps as a result of not being able to gauge comprehension due to the lack of feedback in real-time from students. Unless the teacher has planned for an interactive portion of the lessons, most students will not feel comfortable interrupting the flow of the lectures to let the teacher know that they are not comprehending. Some complained that their teachers did not prepare sufficiently for class, resulting in disorganized lessons that were difficult to follow.

4.3. Challenges faced by students from different academic curriculums

Exploring the problems by the curriculum, Figure 3 revealed that more Art-STEM students suffered from comprehension problems compared to their peers. 25% of Art-STEM students reported having difficulties understanding the subject matter, a percentage twice as high as other

groups. As mentioned previously, the high level of selectivity for popular curriculums like Sci-Math and Art-Lang may contribute to their reported lower comprehension problems.

As for Art-Lang students, a striking 100% reported experiencing assignment overload. Art-Lang students reported spending a lot of time on project-based assignments, from extensive independent research and review of the literature to group playwriting. However, this group struggled the least in schedule overload, suggesting that online studies in arts and languages relied on more independent learning. Therefore, teachers must be more cautious of the learning burden placed on students when choosing this method of teaching.

Sci-Math students reported a moderately high prevalence (54.54-63.63%) for all problems except comprehension, where only 10.9% of the group had trouble as previously explained. Particularly of interest, Sci-Math students reported the highest level of schedule overload and attention problems. The former is unsurprising as this curriculum is made up of more subjects than others, as science is broken down into biology, chemistry, and physics. As for their shifting attention, Sci-Math students reported having difficulties staying engaged during online lessons especially when science was taught in lecture style without experiments or demonstrations. Passively learning about abstract concepts and going through numerous problem sets resulted in monotonous science lessons that failed to capture the attention of the 58.2% of Sci-Math students.

4.4. Students' behaviors and attitudes towards various aspects of online learning

Figure 4.1-4.3 shows students' perceptions towards compulsory use of camera, lesson content, and teachers' IT skills respectively, while Figure 4.4 shows activities carried out by students following online classes.

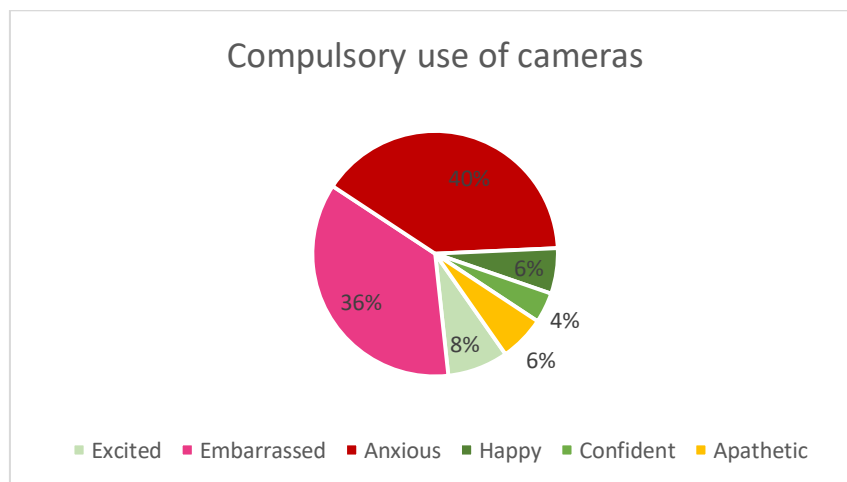


Figure 4.1. Compulsory use of cameras

Figure 4.1 illustrates that most students have negative feelings towards compulsory use of the cameras when learning online. 40% reported feeling anxious followed by 36% who felt embarrassed. Interviews revealed that students felt anxious knowing that teachers would penalize those who did not turn on their cameras as well as those who did but showed signs of inattention. Many reported feeling embarrassed as a result of their peers seeing their less-than-ideal living conditions or their private space as most students would study from their bedrooms. On the other hand, 8% and 6% of students were excited and happy to turn on their cameras, being able to see their friends' faces and expressions. Only 4% stated that they felt confident turning on their

cameras, showing their faces and living spaces. Those who performed well in the class also reported feeling confident on camera, being seen and addressed by their teachers. While some teachers permitted students to keep their cameras off so long as they stayed in class, most made the use of cameras compulsory in their online classrooms in order to monitor students' attendance and attention.

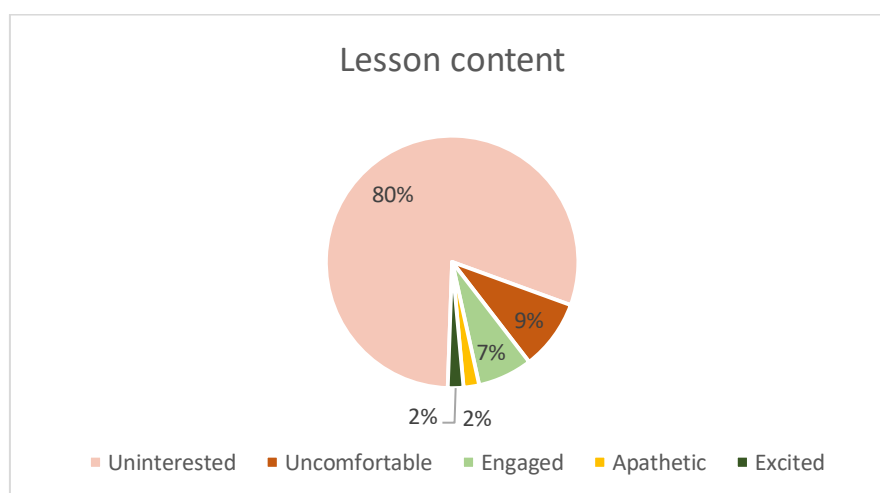


Figure 4.2. Lesson Content

Figure 4.2 shows that a vast majority of students, namely 81%, were uninterested in their online class contents. These students felt overwhelmed and desensitized by a large amount of information thrown at them daily, in the form of mundane slides. Most felt that teachers failed to put in the necessary efforts to present the subject matter in new and engaging ways. Still, 7% of students found their online classes to be engaging and 2% even found them to be exciting when teachers used well-made presentations incorporating pictures, videos, or live demonstrations. Additionally, incorporating interactive activities can also increase engagement. In their interviews, most students lamented the lost opportunity of learning alongside their peers. While a well-prepared presentation on the screen is appreciated, one student echoed the sentiment of many in saying that "it is not enough." Students expressed the need to learn in group settings, "with real people and real discussions". Online learning that did not incorporate these in-person elements made for lackluster learning. To make matters worse, demanding attention, teachers had resorted to disciplining the students through a point system or forcing them to keep their cameras on at all times. While this approach may successfully prevent students' attention from straying, it took away from the joy of learning, exacerbated stress, and led to online learning burnouts.

9% felt uncomfortable during class, citing ineffective equipment and a poor learning environment. This last finding also emphasized the importance of ensuring that online learning is accessible to all, as the lack of appropriate devices and means to learn from home meant some students' primary emotion to learning might be discomfort as opposed to any kind of engagement at all with what was being taught. According to the online survey, 63% of the respondents felt well supported by their families in terms of devices' internet connection.

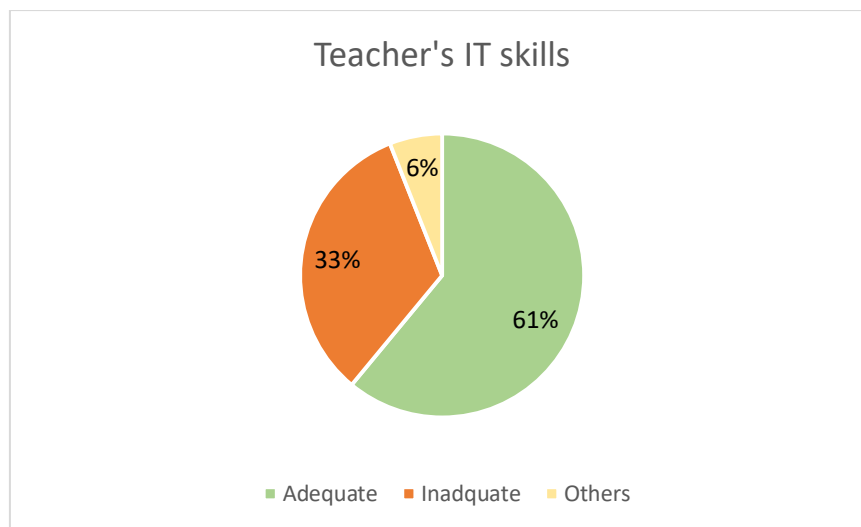


Figure 4.3. Teacher's IT skills

Figure 4.3 shows that at 61%, the majority of students believed most or all of their teachers to have adequate IT skills and/or readiness, while 33% believed most teachers did not. The 6% who chose 'other' further elaborated that they felt the numbers of skilled and unskilled teachers were similar. This result suggested that, even within the same school, there was a wide range of IT skill levels among the teachers. Students reported complicated class attendance rules as different teachers choose to use different meeting platforms. Some may not have the IT training necessary to manage an online classroom by themselves resulting in disjointed lessons or at the very least ones where the online learning platform's capabilities were not used to the fullest. Students also noticed that some teachers did not have stable internet connectivity and often disconnect from time to time during class.

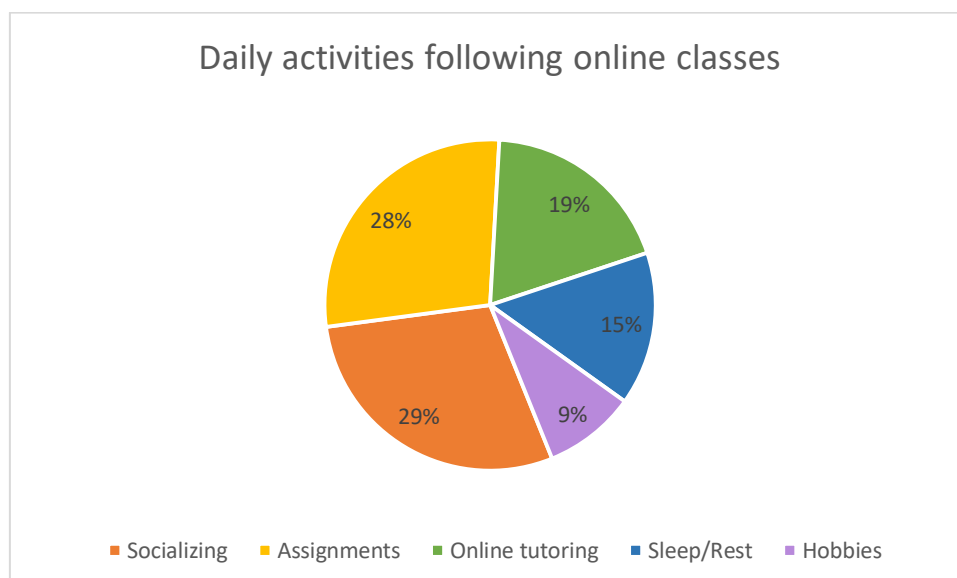


Figure 4.4. Daily activities following online classes

Figure 4.4 shows that following their online lessons, 29% of students socialized with their friends online or used social media. This is consistent with the majority of students sharing that they

missed in-person interaction with their friends, but could only substitute for it by doing so on social media. In addition, students complained of having a large number of assignments due each day. This was reflected in the 28% of students who spent the majority of their evenings on assignments. 15% chose to rest or sleep right after class, while 9% looked to their hobbies to recover from both physical and mental exhaustion. Still, the majority of students shared during interviews that they did not feel they had enough time to relax or unwind before the next day. They also did not have time to prepare for their university applications, which are particularly crucial for grade 11 and 12 students. As such, 19% of students chose to take online tutoring lessons after class in order to prepare for university entrance examinations. In any case, 76.8% of the students believed they were able to manage their time wisely with good planning strategies and organizational skills.

4.5. Students' reflection on the Thai educational system and its management of online education

Students were proactive in their effort to adapt to the sudden shift to online learning. However, they were also frustrated by the poor management and organization of the online programs being provided. One-on-one interviews revealed one theme less emphasized by the online survey; inequality meant online learning is not accessible to all students. Students were made acutely aware of the socio-economic disparity amongst themselves through online learning. Some families may not have the capacity to support their children's online education be it in providing electronic devices, internet, supervision, or even a suitable home environment to learn in. In families with many children, this burden is only multiplied. Economic instability during the pandemic means many parents are now unemployed, but information on financial aid is not readily available both from the government and schools.

In one interview, a student shares that "the Thai educational system doesn't support students, parents, and teachers as much as it should." During the pandemic, students and teachers alike must adapt to new ways of learning and teaching. Some struggled on the way to making the shift, but many students and teachers are still left behind with no access to devices, stable internet, technological skills, or training. Nevertheless, students still tried to turn the crisis of COVID-19 into an opportunity to grow. Those in grades 7-9 who wished to enter a new school had prepared so by looking outside their curriculums - purchasing textbooks to study by themselves, practicing admission exams, or taking online tutoring courses, in hope of transferring to a new school with better learning environments. Meanwhile, grade 10-12 students struggled to prepare for universities with limited guidance from teachers who were equally struggling to master online teaching. As admission requirements are changed almost annually, students must educate themselves on the application process and prepare as early as possible. From choosing the right course and university for themselves, preparing for all the necessary exams to putting together a strong application, the process had left many students feeling overwhelmed and unsupported by their teachers and schools during the pandemic. Nevertheless, students tried their best to prove to themselves that their learning had progressed despite the hindering effects of the pandemic.

5. CONCLUSIONS AND RECOMMENDATIONS

COVID-19 has thrust schools, teachers, parents, and students alike into a new makeshift world of online learning with little preparation. As such, it is no surprise that there remain many kinks in the system that need to be worked out. This study investigates high school students' experiences and evaluations of online learning, naming both advantages and disadvantages compared to in-school learning. On the one hand, students from resourceful schools, with sufficient equipment and access to high-quality teachers will stand to reap the full benefits of online learning, be it the

convenience and the more customizable nature of the platform. On the other hand, students with limited access to the same resources stand to lose some opportunities to learn effectively during this vital period of their education. For either group, COVID-19 requires students to make major changes in the way they learn and socialize. This burden on the physical and mental wellbeing of students cannot be overlooked and support systems need to be put in place both at home and in school.

The interviews conducted show that students are quickly learning from previous years' experiences and constantly evaluating the benefits and drawbacks of the online curriculum provided by their school. As a result, when it comes to improving online learning, students are a mine of knowledge. While some commend the extra free time, they gain from cutting down on commuting to and from schools daily, others complain of that time being taken by schedule and assignment overload. In the area of academics, some report learning benefits such as having more time to review their lessons at their own pace, while others complain of low-quality lectures provided by their teachers.

All in all, it is evident that online learning can either be a curse or a blessing, depending on its structure and management. Schools, teachers, and students must come together in sharing their struggles, evaluations, and recommendations to improve learning efficacy during the pandemic.

First and foremost, schools should be the official point of contact between students and the necessary governmental organizations, advocating for students with financial needs, ensuring that all students can afford online learning. For those who cannot, financial aids, devices, or any necessary teacher support should be provided. Additionally, schools must also advocate the government for their students' well-being, especially those who cannot access vaccines easily.

The majority of students suggest improving the teaching quality. Teachers should make more engaging presentations, fully utilizing the interactive capabilities of online learning platforms. The school must organize adequate training for teachers, allowing all of them to share knowledge and tips on creating a good online lesson, be it in regards to the use of technology or style of teaching.

Moreover, the school greatly help teachers communicate with each other as a whole in order to plan class, grading system, and assignment schedules that are practical and well-balanced for students, taking into consideration the physical and emotional burdens the pandemic has placed on students. Schools should further organize regular meetings between teachers and parents, in order to best understand how students are coping at home and in class and what measures should be in place to help support them. To help lessen the load of the overly packed schedule, students suggest that some subjects such as PE or music may be put on hold until they can return to school or perhaps reduced to weekly hours. Ideally, students would like the option of choosing to spend more or less time on different subjects in accordance with their interests and educational goals.

Basic rules of online classroom conduct should be set on a school basis, rather than left to individual teachers. Specifically, rules that students feel are insensitive to their privacy, such as those requiring their cameras to be on at all times or that they add their teachers on their personal messaging apps should be open to school-wide discussions and up for reconsideration of students' petition for it. Moreover, online exam-taking rules should be discussed between students and teachers in order to find an agreed-upon format that is practical as well as fair for all students.

As we inch closer to the two-year mark of life during COVID-19, students are eager to return to schools while trying their best to adapt to learning online. The evaluation of these advantages and

disadvantages can lead to new progress in technologies, education, and social media, resulting in a new normal for learning and teaching and also for a generation of students. As such, governmental organizations, schools, teachers, parents, and students must come together in an effort to help one another best adapt during this trying time.

ACKNOWLEDGEMENTS

Finally, I would like to thank all the participants of this study for their time and thoughtful recommendations. Moreover, I am grateful to all of my teachers who have given me their time and kind guidance through all the stages of my research. Without them, this work would not be impossible.

REFERENCES

- [1] Asst. Prof. Phatthanahiranrithikorn, "Advantages and Disadvantages of Online Learning," in The 2019 International Academic Multidisciplinary Research Conference in Berlin, 2019.
- [2] M. Alghizzawi, M. Habes, S. A. Salloum, M. Abd. Ghani, C. Mhamdi, and K. Shaalan, "The Effect of Social Media Usage on Students' E-learning Acceptance in Higher Education: a Case Study from the United Arab Emirates," in *International Journal of Information Technology*, 3(3), pp. 13-26, 2019.
- [3] B. Gilbert, "Online Learning Revealing the Benefits and Challenges," M.S. thesis, Dept. Education, St. John Fisher College, NY, USA, 2015. [Online]. Available: <https://core.ac.uk/download/pdf/48619313.pdf>.
- [4] A. D. Dumford and A. L. Mille, "Online Learning in Higher Education: Exploring Advantages and Disadvantages for Engagement," in *Journal of Computing in Higher Education*, 30(3), pp. 452-465, 2018.
- [5] R. Yilmaz, "Problems Experienced in Evaluating Success and Performance in Distance Education: A Case Study," in *Turkish Online Journal of Distance Education*, 18(1), pp. 39-51, 2017.
- [6] V. Arkorful and N. Abaidoo, "The Role of E-learning, Advantages and Disadvantages of Its Adoption in Higher Education," in *International Journal of Instructional Technology and Distance Learning*, 12(1), pp. 29-42, 2015.
- [7] M. Sarraf, H. Al-Shihi, and O. M. H. Rehman, "Exploring Major Challenges and Benefits of M-learning Adoption," in *Current Journal of Applied Science and Technology*, pp. 826-839, 2013.
- [8] S. Sarkar, "The Role of Information and Communication Technology (ICT) in Higher Education for the 21st Century," in *The Science Probe*, 1(1), pp. 30-41, 2012.
- [9] S. S. Gautam and M. K. Tiwari, "Components and Benefits of E-learning System," in *International Research Journal of Computer Science (IRJCS)*, 3(1), pp. 14-17, 2016.

AUTHORS

Pitchsinee Oimpitiwong has been a high school student since 2020 at Triam Udom Suksa School, Bangkok Thailand. She studies the Science-Mathematics curriculum. Furthermore, she is one of the members of the school's oracle club, and has experience in writing reports for publication in school. In 2020, she participated in the school writing essay competition.

MEDIA LEGITIMACY DETECTION: A DATA SCIENCE APPROACH TO LOCATE FALSEHOODS AND BIAS USING SUPERVISED MACHINE LEARNING AND NATURAL-LANGUAGE PROCESSING

Nathan Ji¹ and Yu Sun²

¹Portola High School, Irvine, CA, 92618

²California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

Media sources, primarily of the political variation, have a hastening grip on narratives that can easily be constructed using biased views and false information. Unfortunately, many people in modern society are unable to differentiate these false narratives from real events. Utilizing natural language processing, sentiment analysis, and various other computer science techniques, models can be generated to help users immediately detect bias and falsehoods in political media. The models created in this experiment were able to detect up to 70% accuracy on political bias and 73% accuracy on falsehoods by utilizing datasets from a variety of collections of both political media and other mediums of information. Overall, the models were successful as the standard for most natural language processing models achieved only about 75% accuracy.

KEYWORDS

Data Science, Political Bias, Fake News, Supervised Machine Learning, and natural-language processing.

1. INTRODUCTION

Political bias and fake news is often a challenging task to tackle, especially due to the volatile nature of modern media. The problem is often so engrained, especially in a polarized American society, that it greatly distorts many voters' grip on reality [1] and has been argued to create dubious practices that can often harm society and other people [2]. Although there have been some approaches to tackling the problem, like the Bipartisan press who uses AI to determine bias in their own news articles [3], research in this field is too scarce to offer significant help. Therefore, other scientific methods should be employed to effectively help combat these problems that have begun to encroach on the global order.

The primary goal of the paper is to explore whether data science can be adequately applied to a social science field of identifying political bias and falsehoods. Data science can be broadly understood to be the practice of filtering noisy and unstructured data through algorithms, models, and other scientific practices to achieve results that can be easily interpreted by human users. In the context of this paper, the data is primarily thousands of sentences that contain varying degrees of bias and truthfulness, which is then fed into a variety of models to attempt to achieve the goal.

This methodology often requires techniques to gather large swaths of data, which is incorporated within the procedures later on. The bulk of the data used in this paper has already been gleaned from political sources like senate debates and headlines in popular media.

The various scientific methods mentioned in the data science approach is where the largest variation of outcomes occurs in the paper. There are two primary differences in how the data can be interpreted using natural language processing (NLP) and supervised machine learning, which is the augmentation of a dataset and the kind of machine learning, using a variety of approaches often used in data science for different kinds of data.

Augmenting the dataset is often the first step in this procedure. Most of the time, the models only required a simple augmentation in which special characters and punctuation were removed if needed and the data was changed into a binary numerical format in order for our regression models to have any effectiveness [4]. Since logistic regression and linear regression both require binary outputs of non correlated, individual outcomes, it was important to not have a gradient of falsehoods or bias in the data to ensure this data could work. However, for our initial bigram model, the augmentation was far more complicated since the convote dataset came from senate speeches, which very often contained highly menial sentences including sentences like “Mr. Chairman, I thank the gentlewoman for yielding me this time” or “I yield to the gentleman from illinois” [5]. This meant that there needed to be a way to distinguish what sentences reflected Republican and Democratic senators’ ideology with those that contained only procedural and rather irrelevant information. A method offered by the University of Maryland was to split the dataset into bigrams, a string of two words, and rank them in usage to feed into the model [6]. This augmentation would allow the isolation of popular political phrases like “illegal alien,” which would help identify republican and democratic bias. Further augmentation was used in the actual model, in which the sentences went through pipelines to numericalize the data [7].

Afterwards, choosing the most accurate models and algorithms for interpreting the data was often highly tricky given that there was no knowledge of whether the data was linearly separable or if they were clustered into various groups. The approach was simply to try all the models available to get a better understanding of what the data looked like, which would then allow more specified tweaking of the algorithms. The initial tests only produced very low accuracy scores of 50-60%, meaning that further augmentation and testing was needed. After adding more robust methodology, the final result was that logistic regression and random forest classifiers were both effective and returned relatively similar accuracy of around 80%, which meant that the data was in some way separable in terms of frequency of different word vectors [7]. These types of models are all supervised machine learning, meaning that they were split in train and test sets to develop the model using incorrect and correct answers. Techniques like regression use a loss function to determine the validity of the current model and make necessary changes in order to increase the amount of data points correctly assessed [8]. This paper utilizes both types of supervised machine learning, classification and regression, to determine which best suits the data. In the end, both proved to be nearly equally useful with varying augmentation, meaning that a multitude of future techniques are still open to be able to be used. The final prototype was considered successful in that it proved that data science could be a viable method in identifying political bias and falsehoods purely from analysis of linguistic patterns.

The rest of the paper is organized as follows: challenges of creating an NLP model for fake news, the solution, including the methodology in reference to datasets and algorithms used, the ending results from the algorithms, and then a discussion of the results as well as the implications in the realm of natural language processing.

2. CHALLENGES

In order to create a model that could accurately identify political bias and fake news, a few challenges have been identified as follows.

2.1. Challenge 1

Initial approaches utilized sentiment analysis from various libraries including NLTK using the vader lexicon. However, the primary problem in this usage is that political media can often have highly varying sentiment without actually being false. For instance, sentences like “President Biden signs a new bill” and “a freeway collision has left 6 people dead and 15 injured in a large explosion” are both factually correct, but contain very different sentiments. Furthermore, this meant that the data was not going to be linearly separable nor was it going to be reliable, meaning that as previously stated, regression models would not work on the data and even decision trees could not give a consistent answer due to the vastly varying sentiments.

2.2. Challenge 2

Datasets play a critical factor within the legitimacy and reliability of a data science model, and those for political bias are rather scarce, making it difficult to come to accurate conclusions since identifying political bias needs to mimic human behavior and therefore needs to be generated manually. This presented a challenge as for each dataset found, testing had to be done to each one to determine if the data could be used reliably, and only until the 5th try was a viable dataset found. Additionally, many of the datasets found themselves using non numeric values, which could not be categorized by one hot encoding the data since many times the non numeric values were a gradient that was rather arbitrary. Ultimately, the usage of multiple datasets together would cushion the effect of many inaccuracies the data would have, as well as the usage of converting the non numeric values to a gradient through trial and error.

2.3. Challenge 3

Another challenge that was posed was the effectiveness of the models as well as the metrics generated by Sklearn. It was found that many times the data was extremely overfit or underfit to the dataset, and generated results that do not mimic human behavior. When testing some models, sklearn returned metrics that were rather high, but on manual testing of the model, it was found that the model would determine that sentences like “The sun is 93 million miles away from the Earth” or “The quick brown fox jumped over the lazy dog” were biased or false. Many times, objective sentences like these were flagged incorrectly by a rather high probability, which resulted in either the problems with the pipelines used as well as the datasets. The solution was a multi-faceted approach in which factual and objective sentences would be gleaned from wikipedia to better train the model in identifying these models, as well as changing the parameters that were fed into the model.

3. SOLUTION

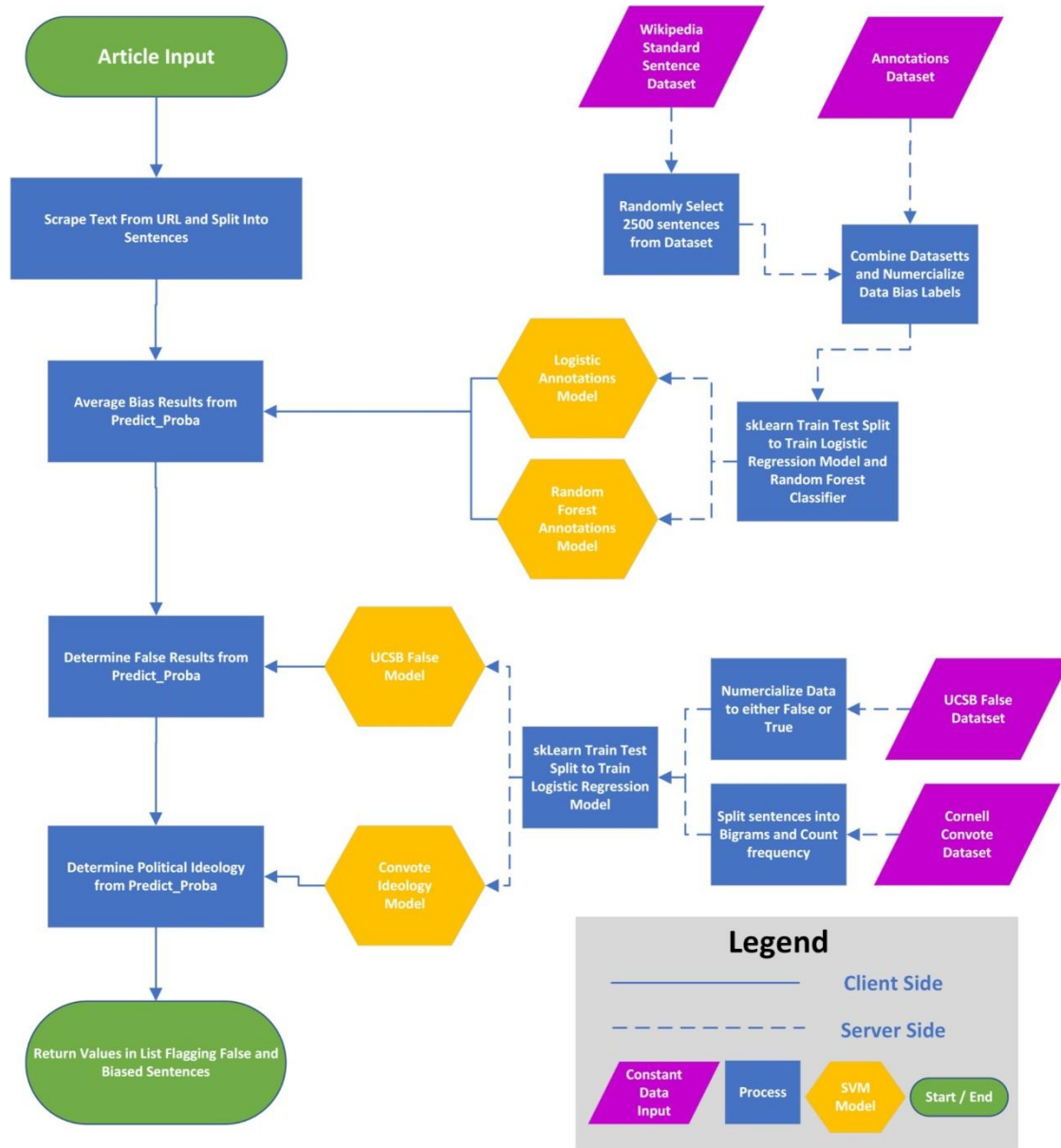


Figure 3.1. The Finalized Prototype

The final solution utilized a variety of techniques that were found to have the highest success rate. The inputs require 4 datasets that will be processed into various models using sklearn's pipelines as well as splitting the data into test and training for the models. Although these models do not need to be regenerated per article using the pickle library, the flowchart displays the first iteration of the prototype.

3.1. Datasets

The Annotations Dataset and Wikipedia Standard Sentence Dataset were both used jointly in determining bias. The Annotations Dataset, developed by MBIC, utilized 10 annotators who were given a set of 17700 sentences and asked to describe various information, including bias, author

information, publication information, and many other characteristics of the data. However, using the dataset alone resulted in data that was overfit since the dataset was

unbalanced as the ratio of biased to unbiased was 10651:7124, which made the model predict bias for factual scientific sentences. The solution was to concatenate the dataset with a split version of the Wikipedia Standard Sentence Dataset, which was used with the assumption that all sentences in the dataset are factual. By using a random number generator, the dataset was reduced from 7 million sentences to only 2500, which would balance the dataset and also include linguistically unique sentences that the annotation dataset might not be including. Afterwards, the data was numerated where biased was replaced by 1 and unbiased was replaced by 0.

The Cornell Convote Dataset gathered the dialogue from Senate speeches in 2006 where senators marked as either democratic or republican had individual files containing all of their dialogue in senate debates [5]. An important consideration is that these files included all dialogue in senate debates including sentences like yielding to other senators, thanking the chairman, and other rather nonsensical sentences that weren't relevant to the task of understanding ideological bias in linguistic patterns. The solution for augmenting this dataset was much different than the other two and did not use full sentences as inputs to a supervised machine learning model. Rather, the sentences were processed by splitting them into a list of bigrams, evaluating the frequency of the bigrams and using those as the weight, which also utilized the count vectorizer and the TFIDF transformer to evaluate sentences based on those bigrams. The other major difference is that this model dataset does not evaluate false or biased media but if they are right or left leaning sentences.

The UCSB False Dataset labeled political sentences as one of six labels on a gradient of truthfulness. Obtained from multiple mediums like social media, the data points can be from both conservatives and liberals and was cross validated manually through human analysis [9]. The dataset also was balanced beforehand, only containing a slight excess of "pants-fire" labels. However, the solution requires a binary output, which is not really compatible with a six label dataset, so rather than balancing the dataset by removing pants-fire label, the solution simply groups the first four true labels into a true output and the next 2 false labels into a false label, thereby balancing the dataset for Supervised machine learning models.

The novel dataset used in this paper utilizes web scraping and models created previously to identify bias to create a larger corpus to train a neural network. The dataset is created by scrapping multiple political news sites that have been evaluated by other papers using human annotators to determine the ideological bias as well as the presence of fake and misleading information. Each label is generated by the

3.2. Models

All of the models utilized in the experiment were Supervised machine learning models that provided a binary output that varied based on the dataset. These models were all provided by sklearn and focused on linguistic analysis as the sole benchmark for finding falsehoods and bias [7]. Logistic regression was used for all three datasets and a random forest classifier was also used in addition to logistic regression in the UCSB False Dataset. Using train test split to evaluate the model, the data was split 80:20 respectively and fed into a pipeline to numericalize the data, which in its raw form is a string.

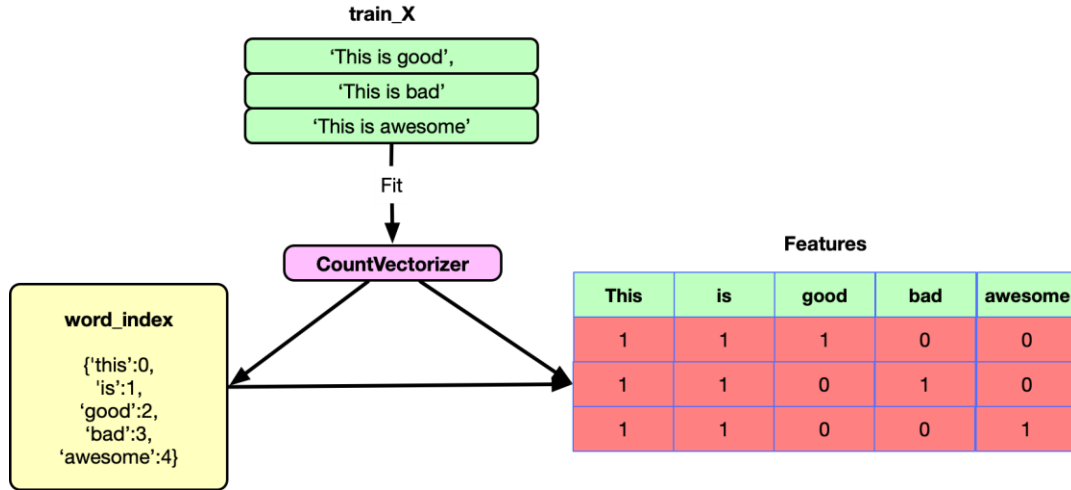


Figure 1. A flowchart of the Counter Vectorizer pipeline [7].

The first step of the pipeline is to pass the sentences as documents into the CountVectorizer pipeline provided by sklearn, which counts the frequency of the occurrences within each sentence and uses that as the word's weight. This is an extension of the bag of words linguistic technique that evaluates linguistic patterns through the usage of frequency. Then, after evaluating the frequency of each word, the solution evaluates its TF-IDF score utilizing two mathematical terms. Given N number of documents, d is the given document, D is the collection of all documents and w is the given word.

$$tf(w, d) = \log(1 + f(w, d))$$

Equation 1: The term frequency (tf) calculation [7]

The solution finds term frequency by taking the natural logarithm of the frequency of the word in the document calculated in the CountVectorizer pipeline added to one. In order to isolate key words within individual documents, the solution rewards a greater frequency for words that occur often in an individual document to emphasize the importance of the word.

$$idf(w, D) = \log\left(\frac{N}{f(w, D)}\right)$$

Equation 2: The inverse document frequency (idf) calculation [7]

The solution finds inverse document frequency by once again taking the natural logarithm of the number of documents divided by the frequency of a word in all the documents combined. The intention of this formula is to isolate words that may not be helpful in linguistic analysis if it is consistently present in all documents like articles “the” or “is” and help emphasize words that appear prevalently in either one document or a small group of documents.

$$tfidf(w, d, D) = tf(w, d) * idf(w, D)$$

Equation 3: TF-IDF calculation multiplying term frequency and inverse document frequency [7]

To fully transform the weights from the counter vectorizer into TF-IDF scores we multiply both

the term frequency and the inverse document frequency to find keywords that prove to be useful for further linguistic analysis. Afterwards, the list of binary labels and TF-IDF scores can be fed into supervised machine learning models that can be separable using logistic regression or classified from Random Forest Classifiers. The solution then calls `predict_proba` on the model that has been fit to the new data to find both the output of the Supervised machine learning models as well as the probability of the output to identify if there is a gradient.

4. EXPERIMENT

The final solution uses multiple supervised machine learning models that are able to discern frequencies within the linguistics of media to reach a conclusion about media bias and falsehoods. The solution's data can be seen below.

4.1. Results and Calculations

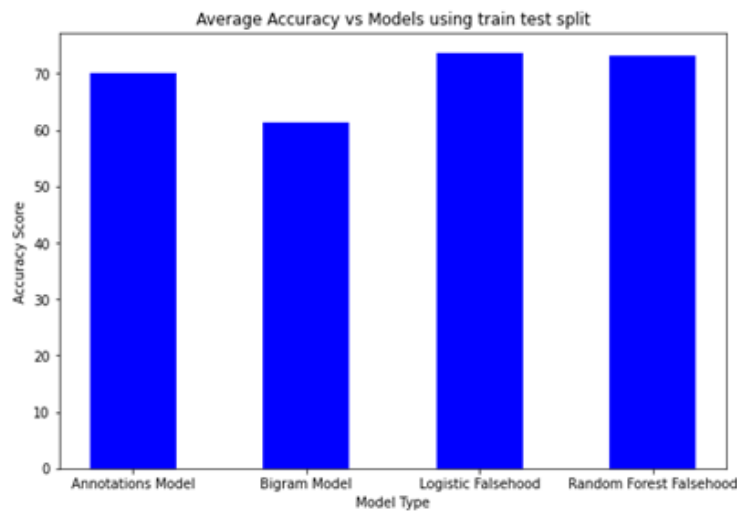


Figure 2. Provides results of Accuracy of all four models at identifying either falsehoods or bias over 10 fold repetitions where the model is regenerated and retrained each time.

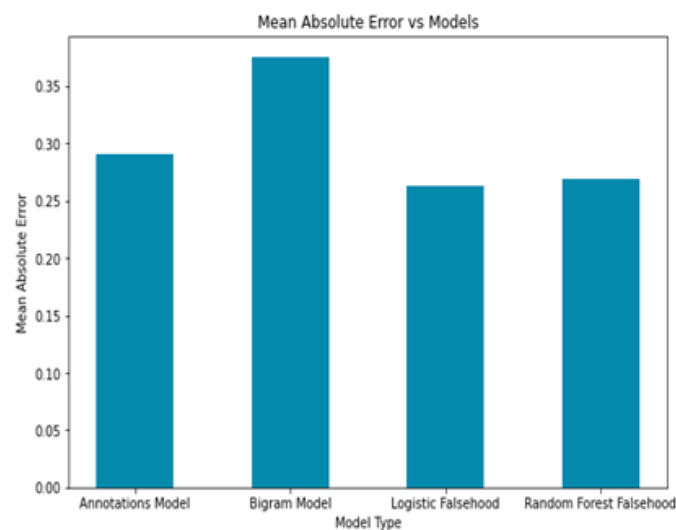


Figure 3. Provides the Mean Absolute Error, evaluated by equation 4 of all four models at identifying either falsehoods or bias over 10 fold repetitions where the model is regenerated and retrained each time.

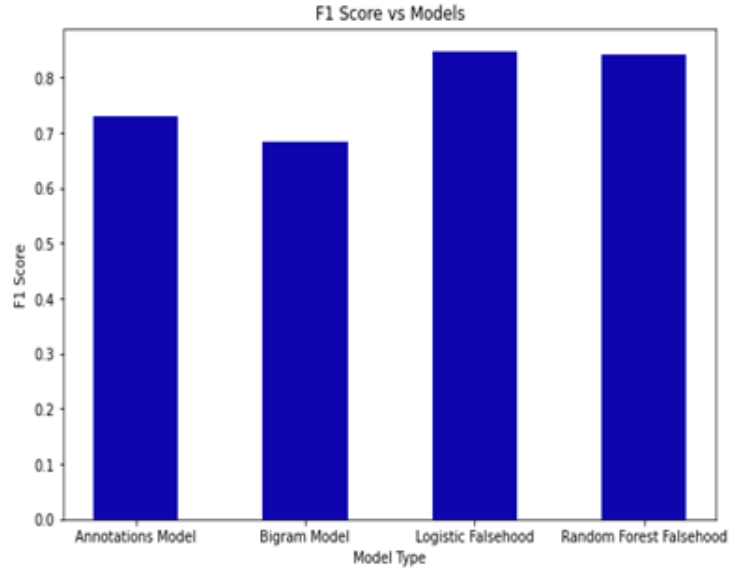


Figure 4. Provides the F1 scores, evaluated by equation 5, of all four models at identifying either falsehoods or bias over 10 fold repetitions where the model is regenerated and retrained each time.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Equation 4: Mean Absolute Error Calculation where n is the number of terms and y_i is the expected value and \hat{y}_i is the observed value.

$$F_1 = 2 * \frac{tp}{tp + \frac{1}{2}(fp + fn)}$$

Equation 5: Derived version of the F1 score calculation obtained from substituting the definition of recall and precision where tp , fp , and fn stand for true positives, false positives, and false negatives respectively.

4.2. Discussion

The results of the model's outcomes for the Annotations and two falsehoods are far more accurate than the Bigram Model and are more supportable for real world application. Understanding that for all the models, the two binary outputs are 1 and 0, any Mean Absolute Error (MAE) that is under 0.5 is essentially all relatively the same in terms of accuracy. This is since one can imagine a cluster around both these endpoints, noting that a MAE that is exactly 0.5 has no overlap between these two clusters. Given that the model's MAE is around 0.2-0.3 aside from the Bigram model, the clusters of results are fairly accurate. The other primarily significant data point is the F1 score of each model, where 1 is absolutely perfect and 0 is entirely incorrect. Understanding the formulas below, the f1 score can be primarily thought of as the harmonic mean between precision and recall. These two levy the correct answers with the two undesired results, effectively creating a numerical representation of how many false positives or false negatives there are in relation to our true positives. Therefore, the higher F1 score displays that the models

are accurate and do not exhibit a worrying amount of incorrect answers. Overall, the statistical interpretation of the model's data proves that the three out of the

four models are applicable for other datasets and their accuracies are justified to be able to be used for real world scenarios.

5. RELATED WORK

This research paper standardizes and compares various sentiment analysis lexicons, being Textblob, W-WSD, and SentiWordNet, resulting in the conclusion that using Support vector models and a Naive Bayes classifier. Using an API, researchers were able to grab tweets and preprocess them to remove URLs and special characters in order to assess their sentiments using multiple models. In relation to the present paper, which originally also used a twitter dataset, the usage of sentiment analysis for twitter proved to be inconsistent and therefore disregarded in the research due to the problems mentioned in this study. Additionally, for political bias, datasets that use tweets are often too inaccurate due to the potential that the tweets do not actually carry political information, which would make scraping them much more difficult and tedious. This study proves that the decision not to utilize sentiment analysis in the present paper was ultimately a useful way to prevent inaccuracies that sentiment lexicons contain to further test the legitimacy of other data science models [10].

This research paper explores media bias on gas drilling the Netherlands, using machine learning models as well as analyzing other factors than linguistic analysis such as media attention by referring to four different newspapers over a long course of time, resulting in findings that not only is media bias present, but is also that risk and the dramatization of the media bias are disproportionately related. Similar to the present paper, the flowchart of preprocessing data and then feeding into a model was utilized. However, this paper used manual validation of the final model in the active learning phase, as well as found different types of media bias over the course of multiple years rather than just analyzing media bias in the present day [11].

This research paper uses a recurrent neural network (RNN) to identify conservative and liberal political bias using a similar dataset as the bigram model in the present paper. The key differentiating aspect is that rather than using bigrams to get over the challenges of gleaning through a dataset containing nonsensical sentences that don't have any inherent bias, the authors handpicked over 4000 data points to use in their neural network. Additionally, by using a long short term memory variant of a recurrent neural network, the researchers were capable of generating a F1 score of around 0.7, which is fairly standard for linguistic models. This variation was also a key difference as it was mentioned that the researchers also were unable to accurately distinguish liberal and conservative bias without the model remembering long term dependencies [12].

This research paper uses a multilayer perceptron model (MLP) to tackle the problem of identifying the political ideology that aligns with possible bias in the articles. They found that the MLP produced about a 9% higher F1 score than RNN and claimed that MLP does in fact perform linguistic analysis better than an RNN model. Rather than using vectorizer pipelines like in the current paper, this paper uses word vectors to numericalized the data and the training also utilized stochastic gradient descent to train the method. Additionally, this model was adapted to a google chrome extension, but the results were fairly inaccurate resulting in almost all articles returning 100% neutrality as well as sometimes misidentifying the bias ideology showing that although the models work in practice with the dataset, live usage of the model is fairly inaccurate [13].

This research paper analyzes whether tweets are opinionated and which political ideology they

fall under if they are. However, this research paper does not physically make use of supervised machine learning models and only uses sentiment analysis corpuses and a usage of trigrams to find results of specific types of opinions. The research paper, however, does mention the future work of using a multilayer perception but notes the hardships the researchers may face since there is no easily accessible corpus for analyzing tweets, especially since so many of them are completely irrelevant to training data science models. Much like the current paper, the usage of tweets and other personal information was very quickly discarded in earlier prototypes because the usage of them for data-driven techniques is rather ineffective [14].

This research paper provides a full faceted approach to identifying political bias using a large majority of datasets online as well as using TF-IDF pipelines to preprocess the data into supervised machine learning models as well as using neural networks. Additionally, the researchers also utilize linguistic theories and use word vectors to find similarities across political bias. An interesting application of the results is the standardization of datasets within the field of work, where the LIAR dataset used in the current paper achieved one of these highest accuracy scores along with another dataset from Fake News Net. This paper finds that across all the recent studies, the best approach is to use a model that is capable of long term learning if one is using a neural network and using hand crafted, manual methods for datasets actually does drastically improve the accuracy of models [15].

6. CONCLUSION AND FUTURE WORK

The scientific findings coupled with the final product does present convincing evidence that linguistic analysis of political news can be used in conjunction with supervised machine learning to identify political bias and falsehoods, but can not identify what ideology those bias and falsehoods may represent. Given that the two falsehood models and the annotations model both did significantly better in statistically supported accuracy than the bigram model, it can be drawn that in order to identify ideological bias, there needs to be other factors incorporated like the media outlet a sentence is from. These findings also standardize supervised machine learning models across each other, showing that both logistic regression and random forest classifiers are well fit for linguistic analysis on possibly biased data. Overall, the data from the experiment does prove the possibility that data science can be adequately applied to fake and biased news, a very prevalent problem in an era where media has become an integral part of people's lives.

6.1. Limitations and Future Work

The current models proposed in the paper suffer from some level of inaccuracy due to the absence of a more robust dataset. Datasets for falsehoods and bias, although present, still are not only too minimal in size to fully overcome the problems of overfitting, but also present only a qualitative estimate of falsehood and bias and not a quantitative estimate of both metrics. All datasets used in this paper use labels such as “false” or “barely true,” which is ultimately subjected to arbitrary brightlines and can be interpreted differently in different experiences. A fully robust dataset that could be produced through continuous web scraping through news aggregators using a mathematical formula to determine bias is a possible solution to the lack of Datasets.

Another limitation faced was that accuracy plateaued at around 70-80%. Therefore, more robust models like the usage of a long term learning for a multilayer perceptron could compensate for weaknesses found in supervised machine learning models. However, the usage of a TF-IDF pipeline still proves to be a very simple but effective method for numericalizing data to find linguistic patterns, but the major difference in the future work is the model doesn't respond to punishment as a way of learning mistakes but remembers patterns useful later on. The more

robust model and a newer dataset may be the most effective way of analyzing linguistic patterns in bias, but in order to fully replicate human behavior, models would need to be trained also with other data independent from the text and may need to consider the publication, author, date, political climate, and ideologies during the current time.

Additionally, it may prove beneficial to focus on specific events that multiple news agencies of different political alignment report on, such as entire presidency administrations, COVID-19, or other major events that could provide a more standardized view of reporting as the base facts would be consistent and therefore do not need to be evaluated. This observation would therefore open doors to pure semantic analysis as a means of determining political leanings especially when events are correlated with actions done by a specific US administration.

REFERENCES

- [1] A. Alesina, A. Miano, and S. Stantcheva, "The polarization of Reality," NBER Working Paper Series No. 26675, 2020
- [2] Levy, Neil. "The Bad News About Fake News." *Social Epistemology Review and Reply Collective* 6, no. 8, pp. 20-36, 2017
- [3] S. Chandler, "This website is using AI to combat political bias," *Forbes*, 17-Mar-2020. [Online]. Available: <https://www.forbes.com> [Accessed: 04-May-2022].
- [4] Zach, "The 6 assumptions of logistic regression," *Statology*, 13-Oct-2020. [Online]. Available: <https://www.statology.org/assumptions-of-logistic-regression/>. [Accessed: 04-May-2022].
- [5] M. Thomas, B. Pang, and L. Lee, "Get out the vote: Determining support or opposition from Congressional floor-debate transcripts." *Proceedings of EMNLP*, pp. 327--335, 2006
- [6] M. Iyyer, P. Enns, J. Boyd-Graber, and P. Resnik, "Political ideology detection using recursive neural networks," *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014.
- [7] Scikit-learn: Machine Learning in Python, Pedregosa et al., *JMLR* 12, pp. 2825-2830, 2011.
- [8] IBM, "What is supervised learning?," IBM, 19-Aug-2020. [Online]. Available: <https://www.ibm.com/cloud/learn/supervised-learning>. [Accessed: 06-Jun-2022].
- [9] W. Y. Wang, "'Liar, Liar Pants On fire': A new benchmark dataset for fake news detection," *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017.
- [10] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, "Machine Learning-Based Sentiment Analysis for Twitter Accounts," *Mathematical and Computational Applications*, vol. 23, no. 1, p. 11, Feb. 2018, doi: 10.3390/mca23010011.
- [11] L. Guo, C. Su, Sejin Paik, V. Bhatia, V. P. Akavoor, G. Gao, M. Betke, D. Wijaya. "Proposing an Open-Sourced Tool for Computational Framing Analysis of Multilingual Data." *Digital Journalism* 0:0, pp 1-22, 2022.
- [12] M. Arkajyoti. "Political Bias Analysis." *Stanford University Computer Science*, 2016.
- [13] M. Vu. "Political News Bias Detection using Machine Learning." *Department of Computer Science at Earlham College*, 2016.
- [14] D. Maynard and A. Funk, "Automatic detection of political opinions in Tweets," *Lecture Notes in Computer Science*, pp. 88–99, 2012.
- [15] R. Oshikawa, J. Qian, W. Y. Wang, "A Survey on Natural Language Processing for Fake News Detection," *Proceedings of the 12th Conference on Language Resources and Evaluation*, pages, pp 6086–6093, 2020.

ENHANCING NETWORKING CIPHER ALGORITHMS WITH NATURAL LANGUAGE

John E. Ortega

Courant Institute of Mathematical Sciences,
New York University, New York, New York, USA

ABSTRACT

This work provides a survey of several networking cipher algorithms and proposes a method for integrating natural language processing (NLP) as a protective agent for them. Two main proposals are covered for the use of NLP in networking. First, NLP is considered as the weakest link in a networking encryption model; and, second, as a hefty deterrent when combined as an extra layer over what could be considered a strong type of encryption -- the stream cipher. This paper summarizes how languages can be integrated into symmetric encryption as a way to assist in the encryption of vulnerable streams that may be found under attack due to the natural frequency distribution of letters or words in a local language stream.

KEYWORDS

Networking, Natural Language Processing, Security, Stream Ciphers.

1. INTRODUCTION

A stream cipher can be illustrated in many ways. In its purest algorithmic form, a stream cipher is a type of symmetric encryption algorithm [1]. A symmetric algorithm achieves encryption by using the same cryptographic keys in order to encrypt or decrypt a message where a shared secret is shared by the sender and the receiver. The sharing of a secret, as most of us know from typical childhood “keep a secret” games, is not secure. And, as a result of their lack of security, stream ciphers must be considered attackable and in need of a stronger defence against attacks and greater security.

Algorithms are the key to privacy but by their nature are public and easy to read. The public availability of algorithms along with the simple frequency of a local language can prove to be devastating for a stream cipher algorithmic modeller. At times, safe stream ciphering can be considered almost as an n-complete problem due to the numerous attacks that have occurred in the past towards them.

The idea that the algorithms should all be public and the secrecy should reside exclusively in the keys is called Kerckhoffs' principle, all serious cryptographers subscribe to this idea. [2] Knowing the cryptography relies on the keys as its secrecy, an attacker will often times focus on breaking the key that is generated by a key generation algorithm. Key generation algorithms are directly used in the majority of stream ciphers and can be considered the weakest link for transferring data due to the aforementioned details where secrecy lies within a key.

One way to prevent attackers from using publicized symmetric algorithm knowledge and key decryption techniques that break stream ciphers is to provide an extra layer of security on top of

the currently available layers. The layer of security should be simple to understand while at the same time robust enough to be applied to any cipher stream available. Several methods [3] have been proposed and are used for strengthening security such as randomness, bit shifting, and the use of digits. Contrastingly, a common framework, while seemingly easy-to-decrypt and insecure, could be the use of a language for encryption. Most networking stream attacks, such as the commonly implemented replay attack [4], use the knowledge of the local language at hand to stage attacks; they normally do not consider the idea of another language being used as an extra layer of encryption.

Stream ciphers, as opposed to block ciphers, hide the pre-known fact that a message will be sent using the local language and; thus, are less prone to simple attacks. This paper explains several stream ciphers and how the addition of a foreign language as an extra layer on top of the current stream cipher capabilities can serve as an extra deterrent for attacks. First, a clear survey of traditional networking algorithms and their vulnerabilities is presented in Section 2. Then, Section 3 gives details on how natural language can be used for encryption in the networking algorithms. Finally, Section 4 describes the reliability and concludes on why it would be better to use natural language for network security.

2. STREAM CIPHERS

A stream cipher (or pseudo-random generator) is an algorithm that takes a short random string, and expands it into a much longer string, that still looks random to adversaries with limited resources. [5] Stream cipher algorithms are typically used as a mechanism for encryption on devices such as wireless routers where encryption is required in order to not expose the data packets that are being passed as messages. Since the data that is being passed back and forth is passed randomly and in real time, data transfer can be considered a stream of packets from one endpoint to another. A stream cipher specifies a device with internal memory that enciphers the j digit M_j of the message stream into the j digit of C_j of the cipher text stream by means of a function which depends on the secret key and the internal state of the stream cipher at time j . The sequence $Z_0, Z_1, Z_2, \dots, Z_n$ which controls the enciphering is called the *key stream* or *running key*. The deterministic automation which produces the key stream from the actual key k and the internal state is called the *running-key generator*, or *key-stream generator*. [6]

The key-stream generator generates the running key sequence described above as the key stream. The key-stream generator combines digit by digit the key sequence, or the running key, on top of the plain text sequence in order to obtain the ciphered text that can be considered somewhat easy to attack due to the fact that the text, although in a ciphered format, is normally produced using letters and/or words from the local language where the data stream occurs.

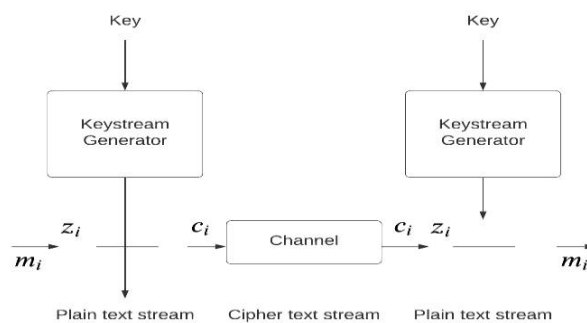


Figure 1. Key generation in a stream cipher. [5]

The "channel" from Figure 1 above shows the typical stream flow as it passes through the stream ciphering process. The ciphered text, meant to be secure, has been found to be vulnerable to attacks due to the frequency of letters in a common alphabet such as English. [7] If the ciphered text's secret key is used more than once, while appearing random to its adversaries, it can be easily decrypted by a skilled cryptographer even though stream ciphers operate with a time-varying transformation on the individual plain text digits.

Stream ciphers depend on a pre-agreed secret for their key encryption. That idea in itself could be considered a security breach since both parties have to maintain the same secret. In this paper, the focus is on the encrypted stream and how to avoid attacks that use cryptic algorithms to decrypt the streams with prior knowledge of a particular language, especially when the same key is used more than once.

There are two major components of a stream cipher algorithm: 1) a short input string (referred to as the **key** in Figure 1 and 2) a long output string called the *key stream*. Stream ciphers can be used for shared-key encryption, by using the output stream as a one-time-pad. [1] The stream cipher can deploy random digits or letters for its encryption and decryption process, this is known as a synchronous stream ciphering process [8]. Additionally, there is another model called *self-synchronous stream ciphering* [9] that calculates ciphered digits using the previous cipher text's digits which automatically synchronize the key generator when receiving the digits.

Both stream ciphering approaches can be considered part of the stream cipher paradigm. In this paper, an additional ciphering mechanism is described to further encrypt the cipher for heightened security that uses natural language as an extra layer to the key stream.

2.1. Stream cipher word frequency

The random digits (numerical or alphabetical) that are formed as part of the encrypted stream in a stream cipher are usually in a local language known to the cryptographer. For that reason, a cryptographer's attempt to decrypt a stream cipher that has been created using an alphabet known by both parties and used multiple times can be considered vulnerable. An attack could be formed that uses the easy-to-discern frequency of digits that focuses on the higher occurring digits, or letters, in the local language [10].

A good example of the weakness of a stream cipher that would typically use a local alphabet for its digits is the RS4 encryption algorithm described in Section 2.2.1. Here, one can assume that the algorithm is easier to attack due to the knowledge of the language at hand. A more concrete example of local language vulnerabilities could be found in a city such as Frankfurt, Germany where an attacker would probably attack a wireless network using the German language due to the fact that more than ninety percent of Frankfurt's inhabitants use German (or Dutch) as their language of choice. On the contrary, the same principle may not be applicable for cities of higher immigration such as Miami, Florida, USA where the spoken language (Spanish) is not the official language of the country (English).

Figure 2 displays the typical frequency distribution of letters in a word of the English language and gives conclusive notions that, by using the knowledge of the digit, or letter, distribution in a language, an attacker may be able to establish an attack model paradigm with ease.

It is clear that a stream cipher whose ciphered output is generated using the English language would, judging from Figure 2 above, probably contain the letter "e" within its context. Therefore, by using the fact that certain letters are more likely to be included in a stream, attacks are normally crafted using higher occurring letters from the local language's alphabet.

A cipher that is streamed, specifically a streaming cipher that uses the same key and input data will produce identical key streams if used with the same key and input data over successive operations. Since the key stream is frequently combined with plain text using an invertible operation, this means that successive cipher texts can be combined to produce a combination of the plain text. [1] That makes an attack, such as a replay attack [4], a good candidate for attack because one could identify the commonality of a repeating stream using easy-to-obtain tools such

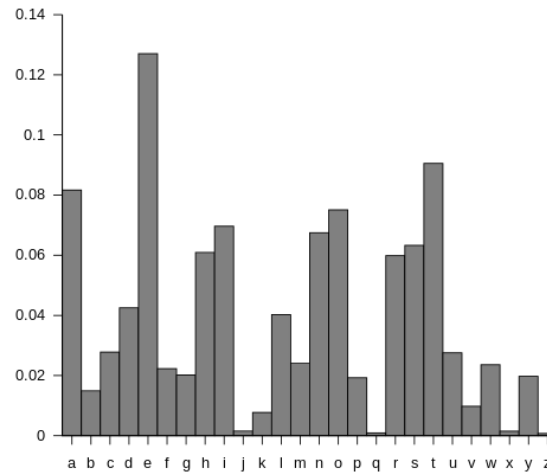


Figure 2. English language letter frequency. [11]

As Wireshark [12] and Aircrack-Ng [13]. A cryptographer could use those tools and a Wi-Fi stream to apply a special type of algorithm implementing the replay functionality that combines the knowledge of local language digit occurrences and possible input phrases to stage an attack.

In addition to simple repetition detection, local language frequency gives way to a high amount of redundancy. In the case of messages with a high amount of redundancy (like in natural language or other data formats), error propagation may be sufficient to detect modifications to a streamed message, but in general an additional cryptographic operation is needed to guarantee the integrity of a message. [14] Stream ciphers are normally processed in real time and the size and quantity of data this is passed via the two endpoints of a stream are normally unknown. Ideally, the algorithm that produces ciphered text in a stream would be random enough such that simple word frequency tactics and reasoning would not be enough for a cryptographer to decrypt. However, due to the easily attainable algorithms that are highly publicized and other general factors that apply to most stream cipher algorithms, stream ciphers are still vulnerable to attack and require an extra layer of security.

2.2. Stream cipher vulnerability

Stream ciphers and their counterparts, block ciphers, are vulnerable due to word frequency probability, local language use, and repetitiveness. The stream cipher key is dependent on the key generator which may produce output of a particular stream cipher that could be considered less weak due to its key. If a key has been generated using a weak algorithm, attacks can be executed with ease. Since many of the key generation algorithms are already published, certain algorithms have been proven to be more vulnerable.

2.2.1. RC4 algorithms in stream ciphers

In a strong key stream generator, each bit of the output will depend on the entire key for its value, and the relationship between the key and a given bit (or set of bits) should be extremely complicated. According to [15], the most widely used stream cipher is the RC4 stream cipher. RC4 is currently found in various applications. In stream cipher context, RC4 can be commonly found in a wireless protocol called wired equivalent privacy (WEP). WEP has already been considered a vulnerable protocol due to its stream cipher key vulnerability; newer protocols such as Wi-Fi protected access (WPA) have already been introduced to replace WEP. WEP is especially vulnerable when the beginning of the output key stream is not discarded, or non-random or related keys are used; some ways of using RC4 can lead to very insecure cryptosystems. [16]

RC4 generates a key stream using an internal state algorithm that has a permutation of 256 possible bytes with two 8-bit pointers. The pointers randomly swap bytes pointed to in order to XOR message bytes. RC4 can be considered a simple and quite elegant algorithm. Nonetheless, its simplicity makes it vulnerable to attacks such as the bit-flipping attack that uses the knowledge of the algorithm to decipher streamed text. It can be understood from Figure 3 that an RC4 application can be deciphered by knowledge of the algorithm easily found on the internet or other publications. A denial of service attack (DOS) [17] could be used to insert plain text that would produce a predictable output exposing the stream cipher's algorithm and, thus, makes it easier for an attacker to attack stream ciphered text. For example, previous work [18] presented an analysis of an RC4 stream cipher showing more correlations between the RC4 key stream and the key and was able to crack an RC4 encrypted algorithm for WEP in under a minute.

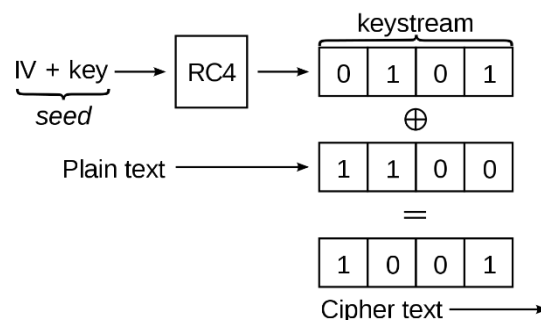


Figure 3. RC4 Stream Ciphers for WEP. [16]

RC4 attacks are now commonplace and almost any primitive hacker can use the knowledge of algorithms such as the RC4 algorithm to attack a stream such as the streams found in wireless WEP technologies for many commonly sold routers on the market. Many variations of the RC4 algorithm have been implemented and, unfortunately, successfully decrypted without the knowledge of the key. The RC4 creates a one-time key of about 24 bits for its security. 24 bit length really cannot be considered safe. The fact that the RC4 algorithm is readily available combined with its key shortness and use of local language digits make it highly vulnerable.

2.2.2. LFSR algorithms in stream ciphers

The LFSR (Linear Feedback Shift Register) algorithm [19] is yet another, considerably insecure, algorithm that can be used in stream ciphers to generate a key. LFSR depends on a previous state by applying a linear function to it. The most common linear function is to take the previous state's bit pattern and XOR it with some bits to modify the overall state. LFSR eventually repeats

because its registers have a finite number of states and, due to the states finiteness, could be considered less secure when states are cycled repeatedly. Nonetheless, if a LFSR algorithm is chosen with a strategic security plan in mind, it could appear randomly acyclic when under attack. LFSRs have long been used as pseudo-random number generators for use in stream ciphers (especially in military cryptography), due to the ease of construction from simple electromechanical or electronic circuits, long periods, and very long periods, and very uniformly distributed output streams. [19] A skilled attacker could decrypt an LSFR quite easily using output text combined with the simulation of a receiver to gain access to encrypted information. One such attack is known as the correlation attack [20].

A correlation attack can be devised to understand the Boolean, cyclic nature of the LFSR algorithm. Predictive possibility tables can be drawn that take the possible input and output in order for the hijacker to be able to decrypt the stream cipher using Boolean logic like Figure 4. So, the decryption would intercept the stream cipher, apply the key stream generation algorithm table using statistical probability, and gain access to the stream. In order to statistically decrypt a stream cipher algorithm, the cryptographer would only have to apply the correlative technique in a key generation algorithm with an algorithm such as the Geffe generator. [21]

Boolean function output table

x_1	x_2	x_3	$F(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Figure 4. A Boolean table for a correlation attack. [21]

If the stream cipher algorithm is implemented using LFSR, the key stream may be too vulnerable and easy to attack, even with another layer of non-local language applied. While a natural language layer could be applied to a stream cipher with LFSR, the simple fact of the repetitiveness in LFSRs cycle make it easier to attain the correct keys. Nonetheless, if one could find a correlation between the output of one of the shift registers and the key stream, then one can try to find the initial state of this LFSR independently of the other LFSRs. [22] Correlation attacks are the most common way to attack LFSR key generations and serve as an example of the weakness of stream ciphers. Correlation attacks can be considered extremely dangerous and stream ciphers extremely susceptible; extreme care must be taken when designing stream ciphers in order to protect against correlation attacks.

3. NATURAL LANGUAGE ENCRYPTION

Natural languages are based on the day-to-day conversations that we experience and can be considered as pieces of information that help us as humans to communicate more effectively

within our domain. Natural languages are governed by implied rules for which natural selection inherently defines. [23] While we can attempt to define those rules using techniques such as finite state transducers (FST) [24], it can be assumed that natural language rules are nearly impossible to approximate via mathematics or grammatical structure, at least with one-hundred percent accuracy. This motivates the study of their use in cryptography as a stronger cipher because they are complex and difficult to solve even by those highly trained in statistical digit, or letter, probability. Overall, it makes practical sense that a key generated with an extra layer of natural language may be more secure due to its grammatical and mathematical incorrectness that make prediction of the key more complex.

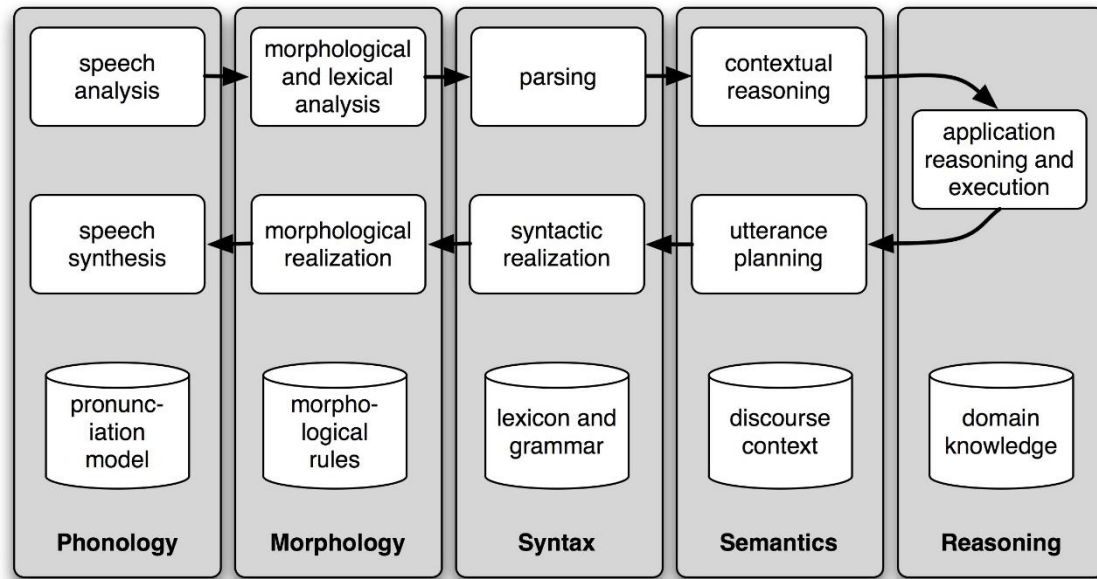


Figure 5. NLP in software. [26]

NLP is the process of a computer extracting meaningful information from natural language input and/or producing natural language output. NLP, as the complexity of Figure 5 shows, is often considered the problem and not the solution due to the difficulty of the task of accepting natural input and producing natural output that are governed by implicit language grammatical models that may not be traceable to any group of persons. Notwithstanding, if a stream cipher is created by the implementation of a natural language that is typically spoken where the stream is being transmitted, the likelihood that the stream can be decrypted using a local natural language is higher than if it were to use a non-local language. In the following section, an introduction to the idea of encryption by using an atypical natural language is proposed.

3.1. Plain text language encoding

A stream cipher that is used for encoding performs its encryption at the level of individual letters or bits. Typically, a cipher, whether a stream or block cipher, uses *plain text* letters to encrypt a message. A cryptographer is considered an expert at decoding *plain text* letters. It is not a surprise, then, that plain texts are often used as targets for decryption algorithms that a cryptographer may routinely use. [10] Plain text taken from everyday sources such as newspapers, recorded telephone conversations, and wireless traffic can be considered a prime target for an attack. The knowledge that a specific target may be written in plain text combined with the fact that a target's implemented language is probably the most common language used within the target's geographic location allows cryptographers to devise plain-text algorithms

using bits or letters from the local language. For example, the following scheme could be used as a way of encrypting letters in English:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Q W E R T Y U I O P A S D F G H J K L Z X C V B N M

This general system, according to Stallings [2], is called a **mono-alphabetic substitution**, with the key being the 26-letter string corresponding to the full alphabet. The encryption key in this example is “QWERTYUIOPASDFGHJKLZXCVBNM”. For the key above, the plain text word: “ATTACK” would be transformed into the ciphered text word: “QZZQEA”. Plain text can be described as the typical writings that we see written in the media that surrounds us and is often near grammatically correct. In order to understand plain text, the plain text's reader would have to have a basic knowledge of grammatical rules that govern the language that the plain text is written in. Stream cipher encoding which uses plain text is insecure when using a locally spoken language of a cryptographer. If the plain text is encoded using a highly redundant language -- such as English or any other natural language -- it can be extracted without knowledge of the key. [1] Ideally, if a sender and a receiver would like to communicate using ciphers and plain text, an encrypted layer must be applied to the plain text in order to make the cipher less vulnerable to attack. One such case where plain text was found to be undecipherable is a study on the Al-Qaeda group in the United Kingdom that used a combination of known natural languages from countries where the group exists such as Pakistan, Yemen, and Sudan. It was almost impossible for the local cryptographers who were accustomed to decrypting messages sent in the native local language, English, to decrypt messages encrypted with natural languages from other countries. [25] The encoded messages were finally decrypted by employing cryptographers from the aforementioned countries. Between them, the code-breakers spoke all the dialects that form the basis for the code. Several of them have high-value skills in computer technology. The local language, native to the Al-Qaeda group, was used as a way of encrypting plain-text messages that could not be understood by the local, mostly English native speakers, inhabitants. Plain text can seem somewhat simple to decrypt. But, if the plain text is written in a language that is not known to the reader and if that language is written in a natural (unstructured) form, it would be much more difficult to decrypt.

3.2. Natural language layer for ciphers

The presence of a natural language can be seen as the weak link of a stream or block cipher. While it may be difficult to determine the text of an encrypted message, given the natural language of a base encryption, a cryptographer can use word frequency algorithms, such as the Berlekamp-Massey Algorithm [27], to exploit one weakness in the decryption process. In that respect, the use of NLP can be seen as a weakness in stream ciphers; however, NLP can also be used for heightened security.

Alternative language constructs can be used as a way of obfuscating encrypted keys. In order to hide the keys, a layer of encryption for increased security can be applied in a language spoken by non-natives to enhance the quality of the algorithm. In NLP, the term “noise” can be defined as the extra phonetics or disturbance inherent to a language that makes the language hard to understand. Languages such as German could be considered “noisy” forms of the English language. [28] With sufficient distortion, or noise, a language can be undecipherable and nearly impossible to dissemble. One may consider this technique as a form of “scrambling” [29]. For example, in the United States, it is known that the Federal Bureau of Investigation has scrambled mobile phone signals when conducting investigations. [30] The noise that one hears when a mobile phone signal is scrambled makes conversations nearly impossible to understand.

This paper proposes the addition of natural language to block and stream ciphers by using a non-native “noisy” layer to scramble text in an encrypted message to the point where a local cryptographer would have a hard time decrypting the message, similar to the scrambled mobile phone message described in the previous paragraph. With additional use of a foreign language to scramble the text, an attacking cryptographer would first have to decrypt a message and then translate it, the translation would be in two or more languages make it very difficult for the most state-of-the-art machine translation systems like those from Google.

Since translation of two or more encrypted languages added as layers to stream ciphers would require that a parallel key is known by the sender and receiver. In this proposal, the parallel key is combined with a non-native language which is considered to be the “noise” of an already encrypted stream. If the noise caused by the encrypted natural language is sufficient enough to scramble encrypted messages, the type of security can be considered an addition to current standardized layers. One example of how this has been done in the past is the use of language mixing by terrorist. [31]

Consider the following example:

native language: bob is a joker

simple encryption algorithm: b=a, o=c, i=r, s=z, a=q, j=g, k=e, e=x, r=t

result: aca rz q gcext

For a cryptographer, the example above would take seconds to decrypt. But, if an additional language was added as an extra layer of encryption, and if the language was a mixture of two or more languages, the message would be tougher to decrypt. Below is the same example using *Spanglish*, the mixture of Spanish with English, a language without official rules spoken in several parts of the world. [32]

mixed language: bob es un joker

simple encryption algorithm: b=a, o=c, i=r, s=z, a=q, j=g, k=e, e=x, r=t, u=h, n=l

result: aca xz hl gcext

While the example above is simple in nature, the decryption technique is more difficult to decipher due to the language not only lacking grammatical sense but also having no meaning after decryption. An attack on the encrypted text above, for example, would be difficult for a person who deals only in English. Additionally, if we assume that the cryptographer was from a country where English and Spanish were not spoken (Russia for example), the decryption above would be even more difficult.

The idea of using natural language in stream ciphers will motivate cryptographers to break it, and if they break it, the stream will be hard to understand because of the ignorance of the natural language that is used in the cipher i.e. ARABIC, Chinese or Japanese, Italian or Greek languages[33]. In order to apply a natural language to a stream cipher, a dependency must be established and the encoding language set as a part of the encryption. The application of the language on top of the stream layer requires that a Unicode representation deemed as input for the second language is created. After the representation has been combined to the stream, an XOR operation is performed on the binary Unicode representation of the input in the second natural language and a binary key is then used to generate an encrypted output. Decryption is finalized by the receiving end using the reverse order.

While NLP can be used as a key deterrent against attacks, it is still not full proof. It is important, for a better level of security, that the generated keys not be repeated twice. Repetition avoidance

applies to stream ciphers specifically because of the encryption cycle that occurs. It would also be wise that the stream cipher's encryption algorithm and its language counterpart use languages that are not so typical to a specific region. For example, if a key generator algorithm created for a wireless router is made in Spain, it would not be wise to create the ciphered text using an algorithm that translates to a nearly similar alphabet such as Spain's neighbouring France.

It can be noted that, by frequency alone, stream ciphers are considered vulnerable. In some ciphers, such properties of the natural language plain text are preserved in the cipher text, and these patterns have the potential to be exploited in a cipher text-only attack. Language models typically written in published algorithms can be trained to learn ciphers. While research is still ongoing, some language algorithms can learn by repetition. Therefore, the pure repetitiveness of certain words such as the article "the" in English can serve as a weak point in a stream cipher text encryption. Cryptographers dedicate themselves to finding patterns in common texts that render symbolic patterns. By applying the NLP technique described here, decryption becomes more difficult due to the language barrier that a cryptographer would probably display. Contrastingly, multi-lingual cryptographers are more likely to find patterns in ciphered texts that have been encrypted with non-native languages due to the fact that they are probably more likely to have seen specific data points within language patterns that serve as key indicators that a stream may have been encrypted using another language.

The insertion of a distinct language in a stream is not difficult to perform. The most important role that language plays in the stream cipher is the protective role of defence. As is typical in stream ciphers, both the sender and receiver must be aware of the language applied and its rules should be made clear before a key is generated. When applying the XOR described above as a binary set, if one of the words does not match a set pattern, the decryption algorithm may be thrown off and more difficult to read. While this may sound simple to do, local languages, by their sheer use, are less likely to be bound by rules which make them less useful in general. Regardless, if a common language can be understood in a local area, rules can be applied to inject the proper encryption. The parallel key (along with key stream bits) for this type of ciphers can be the languages name itself or other world of common interest between two parties may be used[33].

The XOR operation can be considered the single most important part of applying a NLP technique to a stream. An XOR operation is also a key focus of attackers. When adding the language in as an extra layer of protection, the key generation algorithm must be careful that a replay attack can't reproduce through redundancy techniques a way of combining series of messages. For that reason, it is more secure to add a non-local language into the XOR operation. Randomness plays an important key in any key generation technique for stream ciphers. Hence, a naturally spoken language should be clearly known by both the sending and receiving algorithms in order to avoid simplistic yet meaningful collisions that can be translated using a key deciphering algorithm.

The principle vulnerability in a stream cipher, and the reason why the XOR operation is the most important, is the frequency at which letters or symbols occur within the encryption language. The final binary added on as a layer discussed in this paper should help to disqualify stream ciphering encryption detection algorithms. The likelihood of attack would highly decrease if a key is created with high security by using a key that is not repeated and random along with the extra layer of security that languages provide. An attacker would have to have great knowledge of languages and decryption in order to recognize patterns that may occur; especially, if the XOR operation implies a mixture of languages similar to those used by the military originally created by native American tribes [34].

4. STREAM RELIABILITY AND CONCLUSION

Application and protocol designers, even those with experience and training in cryptography, cannot be expected to always identify accurately the requirements that must be met for a mode to be used securely or the conditions that apply to the application at hand. As in [34], private enterprises such as Google and Microsoft receive millions of attacks a year. Whether an enterprise level user or a simple home user, network security, no matter at what level, can be attributed to a price with information containing a value. The protection of that information really depends on its value. Credit card numbers may be considered more important than a user id for an adventure gaming website. Higher valued items and messages are retrieved via network streams of data and are captured and decrypted by skilled cryptographers. The heightened sense of security towards streams must be considered important.

Attacks are direct and easy to accomplish with the current attacker tools available. Wireless WEP attacks have proven to be as simple as inserting a disc or usb into a laptop and pressing enters. Although the latest wireless networks seem to be more secure and robust, keys are retrieved through cryptology and it is inevitable that algorithms will be created to decrypt the most difficult encryption. But, if tactics such as the NLP layer described in this paper are employed, a cryptographer's job can be made considerably more difficult.

REFERENCES

- [1] S. Burnett and S. Paine, RSA Security's official guide to cryptography. McGraw-Hill, Inc., 2001.
- [2] W. Stallings, Network Security Essentials. Pearson Education Limited, 2017.
- [3] S. Malladi, J. Alves-Foss, and R. B. Heckendorn, "On preventing replay attacks on security protocols," IDAHO UNIV MOSCOWDEPT OF COMPUTER SCIENCE, Tech. Rep., 2002.
- [4] R. El Abbadi and H. Jamouli, "Takagi Sugeno fuzzy control for a non-linear networked system exposed to a replay attack," Mathematical Problems in Engineering, vol. 2021, 2021.
- [5] Special relativity. (2021). Retrieved January 5, 2022, from https://en.wikipedia.org/wiki/Stream_cipher.
- [6] T. W. Cusick, C. Ding, and A. R. Renvall, Stream ciphers and number theory. Elsevier, 2004.
- [7] L. Singh and R. Johari, "Clct: cross language cipher technique," in International Symposium on Security in Computing and Communication. Springer, 2015, pp. 217–227.
- [8] R. A. Rueppel, "Stream ciphers," in Analysis and Design of Stream Ciphers. Springer, 1986, pp. 5–16.
- [9] C. S. Lamba, "Design and analysis of stream cipher for network security," in 2010 Second International Conference on Communication Software and Networks. IEEE, 2010, pp. 562–567.
- [10] Li, H., Chen, M., Yan, S., Jia, C., & Li, Z. (2019, September). Password guessing via neural language modeling. In International Conference on Machine Learning for Cyber Security (pp. 78-93). Springer, Cham.
- [11] Frequency Analysis. (2021). Retrieved January 8, 2022, from https://en.wikipedia.org/wiki/Frequency_analysis.
- [12] Wireshark. (n.d.). Wireshark. Retrieved January 9, 2022, from <https://www.wireshark.org>.
- [13] Airshark-NG. (n.d.). Airshark-NG. Retrieved January 3, 2022, from <https://www.aircrack-ng.org/>.
- [14] H. C. Hudde, "Building stream ciphers from block ciphers and their security," Seminarar be it Ruhr-Universität Bochum, 2009.
- [15] R. Wash, "Lecture notes on stream ciphers and rc4," ReserveUniversity, pp. 1–19, 2001.
- [16] Wired Equivalent Privacy. (2022). Retrieved January 5, 2022, from https://en.wikipedia.org/wiki/Wired_Equivalent_Privacy.
- [17] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram, and D. Zamboni, "Analysis of a denial of service attack on tcp," in Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No.97CB36097). IEEE, 1997, pp. 208–223.
- [18] A. Klein, "Attacks on the rc4 stream cipher," Designs, codes and cryptography, vol. 48, no. 3, pp. 269–286, 2008.

- [19] L.-T. Wang and E. J. McCluskey, "Linear feedback shift register design using cyclic codes," *IEEE Transactions on Computers*, vol. 37, no. 10, pp. 1302–1306, 1988.
- [20] W. Meier and O. Staffelbach, "Fast correlation attacks on stream ciphers," in *Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1988, pp. 301–314.
- [21] Correlation Attack. (2021). Retrieved January 1, 2022, from https://en.wikipedia.org/wiki/Correlation_attack.
- [22] Lund University: Faculty of Engineering. (2009). Project 3: Correlation Attack [Slides]. <https://www.eit.lth.se>.
<http://www.eit.lth.se/fileadmin/eit/courses/edi051/projects/corattack/CorrAt.pdf>
- [23] S. Pinker and P. Bloom, "Natural language and natural selection," *Behavioral and brain sciences*, vol. 13, no. 4, pp. 707–727, 1990.
- [24] M. Mohri, "Finite-state transducers in language and speech processing," *Computational linguistics*, vol. 23, no. 2, pp. 269–311, 1997.
- [25] Staff, W. (2009, November 21). Al-Qaida "secret language" decoded. *World Net Daily*. Retrieved December 28, 2021, from <https://www.wnd.com/2009/11/116381>.
- [26] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. O'ReillyMedia, Inc., 2009.
- [27] N. B. Atti, G. M. Diaz-Toca, and H. Lombardi, "The berlekamp-massey algorithm revisited," *Applicable Algebra in Engineering, Communication and Computing*, vol. 17, no. 1, pp. 75–82, 2006.
- [28] C. Bakir, "Automatic voice and speech recognition system for the German language with deep learning methods," *International Journal of Applied Mathematics Electronics and Computers*, no. SpecialIssue-1, pp. 399–403, 2016.
- [29] V. Phillips, M. Lee, and J. Thomas, "Speech scrambling by there-ordering of amplitude samples," *Radio and Electronic Engineer*, vol. 41, no. 3, pp. 99–112, 1971.
- [30] Y. Zou, G. Zhang, and L. Liu, "Research on image steganography analysis based on deep learning," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 266–275, 2019.
- [31] M. Guidère, N. Howard, and S. Argamon, "Rich language analysis for counterterrorism," in *Computational Methods for Counterterrorism*. Springer, 2009, pp. 109–120.
- [32] A. Ardila, "Spanglish: an anglicized spanish dialect," *Hispanic Journal of Behavioral Sciences*, vol. 27, no. 1, pp. 60–81, 2005.
- [33] M. Mahmud, "Natural language (arabic) as a strengthening layer for stream ciphers in wireless networks," in *Proceedings of the 17th IASTED International Conference*, vol. 609, no. 090, 2008, p. 130.
- [34] S. Marshall, "A hidden story: American indian code talkers," *DttP*, vol. 40, p. 27, 2012. [25] S. M. Bellovin and M. Blaze, "Cryptographic modes of operation for the internet," 2001

AUTHORS

John E. Ortega is a PhD holder from the Universitat d'Alacant and a member of the European Association for Machine Translation. He has over 15 years of software engineering and development experience in the private sector. His main field of research is fuzzy-match repair, although he has also worked on low-resource machine translation and topic modelling. He has earned several patents for various bayesian techniques and is a professor and guest lecturer at New York and Columbia Universities.



AI_BIRDER: AN INTELLIGENT MOBILE APPLICATION TO AUTOMATE BIRD CLASSIFICATION USING ARTIFICIAL INTELLIGENCE AND DEEP LEARNING

Charles Tian¹ and Yu Sun²

¹University High School, 4771 Campus Dr. Irvine, CA 92612

²California State Polytechnic University, Pomona,
CA, 91768, Irvine, CA 92620

ABSTRACT

Birds are everywhere around us and are easy to spot. However, for many beginner birders, identifying the birds is a hard task [8]. There are many apps that help the birder to identify the birds, but they are often too complicated and require good internet to give a result. A better app is needed so that birders can identify birds while not depending on internet connection.

My app, AI_Bider, is mainly built in android studio using flutter and firebase, and the AI engine is coded with TensorFlow and trained with images from the internet [9]. To test my AI engine, I made six different prototypes, each having a different number of times that the code will train from the dataset of pictures. I then selected 5 birds that are in my dataset and found 5 pictures on the internet for each of them, which I then uploaded to the app. My app will then give me 3 bird species that most closely resemble the image, as well as the app's confidence in its choices, which are listed as percentages. I recorded down the percentages of accuracy for each picture. After taking the average percentage of all the models, I selected the most successful model, which had an average percent of accuracy of 79%.

KEYWORDS

Machine Learning, AI platform, Computer vision.

1. INTRODUCTION

There are countless birders in the world, and many of them struggle with identifying the bird that they saw, which is the most important part of bird-watching. Many birders resorted to using field guides or existing birding apps like Merlin or Audubon birding Apps. However, the former methods are highly inefficient. A field guide requires the birder to memorize the features of the bird while having to flip through all the pages to find the bird. This method is highly impractical because not every birder has a field guide and the experience needed to efficiently use it. The other methods, the apps, are highly unreliable as well for the apps rely on the users to memorize details of the bird - color, size, tail shape, the type of the bird, its activity, habitat, voice, and wing shape - to accurately identify a bird. To an inexperienced birder, this could be a grueling task because a new birder has not likely been exposed to the different behaviors of birds, not to mention being able to spot body shapes and color for a fast-moving target in the air. Learning from and improving off of these common flaws of the former methods, I am going to create an app that can accurately identify birds using Artificial Intelligence, while only needing a picture of

the bird from the user. The benefits of my app are that my app is going to be highly accessible to all birders for my app will be free to download from the google play store or the app store for IOS. And also my app will be easy to use and successful because of the implementation of AI identification. Overall, my app has the potential to be very popular among birders and can be used instead of previous identification methods.

Using Artificial Intelligence to identify birds has been attempted by scientists before by Scientists from Kimberley, South Africa (Ferreira et al, 2020), and the Mutah University of Karak, Jordan (Al-Showarah and Al-qbailat, 2021) [3]. They used deep learning to accurately identify small birds and general birds, respectively. Ferriera's group used convolutional neural networks(CNN), a type of deep learning that automatically analyzes data like color, shape, and sizes(Ferreira et al, 2020). Al-Showarah and Al-qbailat used a type of CNN called VGG-19(Al-Showarah and Al-qbailat, 2021). VGG-19 is a type of CNN that is pre-trained and can identify and distinguish different traits. This feature enables VGG-19 to identify objects like birds better than the conventional CNN models. Al-Showarah and Al-qbailat also used principal component analysis(PCA) to decrease the dimensional of the AI code while minimizing data loss [11]. The dimensional of a dataset is how many input variables or features that dataset has [10]. PCA, therefore, increases the efficiency of the code by significantly cutting down the time for the engine to train and produce an accurate model [12]. Both groups had a significant number of bird pictures that groups used as data for the AI engine. Al-Showarah and Al-qbailat used images from a database to train the model, whereas Ferriera's group captured bird pictures in the wild, setting up cameras near bird feeders. Although the latter method might produce more reliable data, the formal method is significantly more accessible and can be used to gather data for birds that are not native to the coders' regions. Using CNN, both groups obtained highly accurate AI models. However, their engines are not too accessible to general birders for the engine is trained and stored on the device. Therefore I designed an AI model and connected it to an app that users can access easily.

In order to make an APP, I used flutter, Firebase, and TensorFlow. All of my code was coded in Android Studio. To make the user interface and the different pages of my app, I used flutter. Flutter is a software development toolkit (SDK) developed by Google [13]. Inside of the Flutter framework, everything is coded with Dart, a coding language that is similar to java (Amadeo, 2018) [1]. For the back-end part of the APP - accessing the pictures that need to be displayed on the APP, getting the name of the birds that are displayed on the result page, logging-in information, etc. - I used Firebase. Firebase is a database service system that provides data storage and a server for hosting (Lardinois, 2014) [2]. The AI code is inspired by the CNN used by the previous scientists; I used TensorFlow, which is a library that contains materials needed for making my AI model. One major difference between my APP and the AI models of previous scientists is that I am able to display the results of my AI model on my APP, which is more appealing to general users. My app also allows the user to accurately identify birds while not worrying about internet connection. I have ensured this feature in my app by downloading a pre-trained AI model. This way the user can both use a functional model and not worry about training the model on their devices, for the model is ready from the moment the user downloads the app.

To prove my result and test my prototype, I mainly tested my AI model. This is because the main user interface and the "APP" part can not really be tested, rather the APP either works or doesn't. To test my AI Model, I tested my model's accuracy when identifying birds. I first selected five random birds that are in my dataset and found five random images from the internet for each of the five birds. Then I uploaded the images to my app and recorded the result in my notebook. To improve my model, I experimented with the number of times my model retrained itself. In my code, this amount of time is determined by a variable called epochs. And by changing the value of epochs, the model will train itself accordingly. I later experienced different numbers for the

value of epochs and selected the model with the best result. After selecting the best model, I cleaned the data in the training code. Cleaning the code means that I went through some of the training images in my dataset and deleted the ones that either have a disruptive background that hinders the machine from recognizing the bird. For example, a picture might have the bird covered by a leaf or a shadow, this alteration will decrease the effectiveness of the AI model for the picture has the wrong color(from the shadow) and the incorrect shape(from the leaf). Inaccurate pictures(ie. the wrong bird is displayed) are also crucial to be removed or else the model will be reading through all the images while training for the wrong bird.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that I met during the experiment and designing the sample; Section 3 focuses on the details of my solutions and prototype corresponding to the challenges that I mentioned in Section 2; Section 4 presents the relevant details about the experiment I did, following by presenting the related work in Section 5. Finally, Section 6 gives the concluding remarks, as well as points out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Creating the app using flutter and Firebase

The first challenge I faced was creating the app using flutter and Firebase. Making an app is challenging because the process requires the coder to be familiar with a lot of keywords. Flutter has a lot of specific keywords that are not often used in python or java, like widgets, scaffold, app bar, state, floating, etc. I coded with Java and python before; although dart, the coding language for Flutter, is similar to Java, switching to Dart and learning all the new keywords took a lot of practice and time. Connecting my app to Firebase is another issue, for I have never used Firebase before. Firebase is an important part of my project, serving as the back-end portion of the project. Oftentimes I had to search up online guides and resources to search for each keyword and what they do. I also had a problem accessing an emulator to test out my APP, for I have a Ryzen CPU and it at first did not support the android emulator provided by android studio. But that problem was solved by some BIOS updates.

2.2. Making an accurate AI model using TensorFlow

The second challenge I faced was making an accurate AI model using TensorFlow [14]. I have created AI engines with python, namely python PANDAS [15]. However, python PANDAS does not work as well with android studio and flutter compared to TensorFlow. But creating an AI Model was not easy. Just like making the app, there are a lot of keywords like labels, subsets, etc. The AI model consists of many different sections, there is an initializing part where the images are reshaped for preparation, a training part, and a validation part. And I had to refer to online guides to creating AI Engines with TensorFlow. Other than making an AI Model, I had trouble making it accurate, for my first prototype was only 53 percent accurate on average. I started to test out different methods to improve my model. I experimented with different numbers of times that the code retrains itself using the dataset. Another thing that I tried was cleaning the dataset by eliminating any inaccurate pictures and adding pictures that directly describe the bird.

2.3. Finding a large amount of accurate data

The last challenge I faced was finding a large amount of accurate data. The image in the dataset is a crucial part of making the AI model. The accuracy of the AI model significantly depends on how accurate the training images are. At first, I used a dataset online that contained around one hundred and fifty pictures per bird, and there is a total of three hundred and twenty-five birds. However, there are way more birds on earth than three hundred and twenty-five, so I must find a way to bulk copy images from the internet for each bird. To solve this problem, I used a bulk image downloader that will find and download however many pictures I told it to download from Bing. With the bulk downloader, I did not need to worry about finding images for my data. However, I still needed to go through the pictures to make sure they are accurate and are qualified to be used for my model. Unfortunately, there is not a more efficient way to check the quality of the images than manually going over them.

3. SOLUTION

The Overview of AI_Birder is presented in Figure 1. The user first creates an account and arrives at the main page. On the main page, the user can browse through the birds that the other users of AI_Birder have found. The birds are presented in a list with their name and a picture of the bird. And if the user clicks on the picture or the name, the APP will open up a new page showing all the pictures of the birds that have been uploaded by the users. If the user wants to identify a bird, they can return to the main page and either take a picture (pressing the camera icon button in the bottom right) or upload an image from the device (clicking on the image icon button in the top right). By clicking on either of the two buttons, the user can either select an image or take a picture. After that, the user will arrive at the AI identification page, where the AI engine will display the name of three bird species that are closest to the image uploaded. After the engine returns three bird species, the user can select the correct bird name. The image will then be added to the list of pictures for that bird species, and if the bird does not have any pictures yet, the APP will create a new section with a list of images for the new bird. And if the bird is not in the database, the user can enter the correct name of the bird, and there will be a new bird species added to the database. The main components of the APP are the login/logout and create account page, the main page, and the AI identification page with the AI engine.

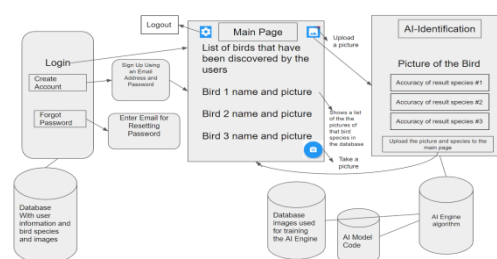


Figure 1. Overview of AI_Birder

To start off, Figure 2 shows the code that creates the Login page. First, I created a widget, or an object, called loginBody. This loginBody returns a container that contains text boxes, padding, buttons, etc. Padding are used to leave some space on the left and right sides of the app to make the page look aesthetically pleasing. CustomTextFields are also used for the user to input their email address and their password. I also implemented an elevated button at the end of the code. When the user clicks this button, the button takes the value of the email and the password to set the user. The code also checks the email and the password beforehand to make sure that the user entered a valid email.

Figures 3 and 4 show the main page and the code that gets all the images and names from Firebase, respectively. In order for the code to get all the images and names, the images and names have to be stored somewhere. For AI_Birder, the names and images are stored in Firebase. Figure 5 shows the dataset that contains the images and names of each discovered bird. The “cover” photo of the bird species and its name is stored in the bottom right field, where there is an URL to the image and the name as a string. The rest of the photos are stored in the collection called “data ” in the top right. Those photos are displayed when the user clicks on one of the bird species(a new page will open up with a list of the pictures of that species). The code is able to get the images from Firebase by returning a scaffold with a streambuilder [4]. The streambuilder builds an object based on the latest updated screenshot(state) of the stream, in this case, the stream of images. The streambuilder returns a listview of a column of images so that the user can see all the pictures of the bird. Another component of AI_Birder is the AI engine. The engine is made by implementing TensorFlow. The engine mainly trains itself by using images in a database. The engine uses the images in the training folder to train itself, then it uses the images in the validation folder to test its accuracy. The AI engine is modeled after MobileNetV2, which is a pre-trained CNN(see Figure 6). And from there, the model adds onto MobileNetV2 by recognizing patterns in bird pictures. This adding-on is called transfer learning, where, compared to traditional machine learning, the model creates a new neural network based on an existing one. This feature makes transfer learning more efficient and more effective than traditional machine learning types. After implementing MobileNetV2, multiple layers - Conv2D (which finds the patterns), Dropout(prevents overfitting), Global Average Pooling 2D (returning the average output of information learned from previous layers), and Dense (receives all the information from previous layers) - trains the AI engine. With all the training done, the code returns a label (all the birds’ names) and a model, which is the trained AI engine. Those two things can be connected to the APP by uploading them to the code. After uploading, the APP is now functional with an AI engine.

```
Widget loginBody() {
  var targetIndex = 1;
  var targetPadding = 2;
  return Container(
    padding: EdgeInsets.symmetric(horizontal: 50),
    child: Column(
      children: [
        customTextField(
          controller: emailController,
          hintText: 'Email',
          inputType: InputType.email,
        ),
        customTextField(
          controller: passwordController,
          obscureText: true,
          autocorrect: false,
          hintText: 'Password',
          inputType: InputType.password,
        ),
        SizedBox(height: 20),
        customElevatedButton(
          onPressed: () {
            widget._firebaseAuth.signInWithEmailAndPassword(email: emailController.text, password: passwordController.text).then((user) => widget.setUser(user: user));
          },
          text: 'Log In',
        ),
      ],
    ),
  );
}
```

Figure 2. Code for creating the Login page



Figure 3. The main page of AI_Birder

```
Widget Build(BuildContext context) {
  // getting a snapshot of the species collection from our firebase database
  // and passing it as a stream of the type query snapshot =/
  final Stream<QuerySnapshot> _birdStream = FirebaseFirestore.instance.collection('species').doc(widget.birdType).collection('data').snapshots();

  return Scaffold(
    appBar: AppBar(
      centerTitle: true,
      title: Text(widget.birdType),
    ),
    // create a screen from a stream of data - our specifically is the snapshot of the species collection
    body: StreamBuilder<QuerySnapshot> (
      stream: _birdStream,
      // snapshot - the current data within our collection
      builder: (context, snapshot) {
        if (snapshot.hasError) {
          return Text('Some error has occurred');
        }
        if (snapshot.connectionState == ConnectionState.waiting) {
          return Text('Loading');
        }

        return ListView(
          padding: EdgeInsets.only(top: 10),
          children: snapshot.data!.docs.map((document) {
            var docData = document.data() as Map<String, dynamic>;
            var downloadUrl = docData['image-url'];
            return Column(
              children: [
                Image.network(downloadUrl, width: 350,
                  SizedBox(height: 10,
                ), // Column
              ).toList(),
            ); // ListView
          }); // StreamBuilder
        ); // Scaffold
      }
    );
  }
}
```

Figure 4. Code for getting the names and the pictures of each bird species

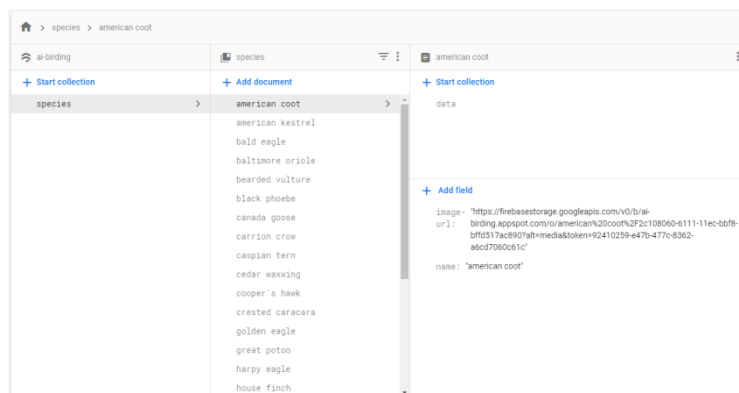


Figure 5. The Firebase database page

```

labels = '\n'.join(sorted(training_generator.class_indices.keys()))

# open the file 'labels.txt' and writing labels to it
with open('labels.txt', 'w') as f:
    f.write(labels)

# the size of our image (224, 224, 3)
img_shape = (img_dimension, img_dimension, 3)

# create a base model using MobileNetV2
base_model = tf.keras.applications.MobileNetV2(
    input_shape=img_shape,
    include_top=False # used to prevent training previous layers and only adding on new ones
)

base_model.trainable = False
model = tf.keras.Sequential([
    base_model,
    tf.keras.layers.Conv2D(32, 3, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.GlobalAveragePooling2D(),
    #change the number of folders
    tf.keras.layers.Dense(327, activation='softmax')
])

model.compile(
    optimizer=tf.keras.optimizers.Adam(),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

```

Figure 6. Code for training and creating the labels and model for the AI Engine

4. EXPERIMENT

4.1. Experiment 1

To test my prototype, I mainly tested the accuracy of my AI engine, and I experimented with my prototype to find the best model. I experimented with a different amount of times that my code retrains itself using the dataset images, and I also experimented with a different number of training pictures for a species of bird.

The first experiment I did was testing the accuracy of my AI Model and experimenting with different amounts of times that my model retrains itself using the pictures in the dataset. That amount of time is determined by a variable called “Epochs”, so I changed that variable throughout the experiment. To test my AI model, I randomly selected five birds that are in my dataset and found 5 random pictures from the internet. I then uploaded the images to my app and recorded the result and took the average for each bird and for everything. Because my final average is the result of 25 random pictures, the average in the result reliably reflects the accuracy of my AI model.

Table 1 and Figure 7 illustrate the resulting percent of accuracy of all the models with different epoch values. By experimenting with different numbers of epochs, I discovered the model with Epochs = 11 (the code will read through and train with all the pictures in the dataset 11 times) was the most successful. My data shows that the AI engine doesn’t become more accurate as it retrains itself more times. Because my AI engine was highly inaccurate - with a result of 52.2% average accuracy - when it retrained itself 19 times. This performance was due to the model overfitting, meaning that the model has trained itself too many times that it can only accurately identify the images in the training dataset. On the other hand, the model with only 9 epochs was also inaccurate because the engine did not train sufficiently. The model with epochs = 11 had an average percent of accuracy of 79 percent, proving itself to be the best model. I will be using the model with epochs = 11 for future experiments and tests since it yielded the best result.

Epochs vs accuracy(average)	Average Percent of Accuracy(%)
9	65.6
11	79
13	55.2
15	63.8
17	72.32
19	52.2

Table 1.Average Percent of Accuracy for Models with Different Epochs value

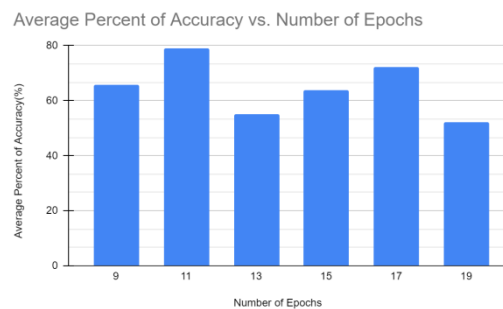


Figure 7. Average Percent of Accuracy for Models with Different Epochs value

Bird Species	Average Percent of accuracy
Red-tailed Hawk	83.8
Northern Shoveler	83.2
Mourning Dove	57
Black Phoebe	92
Bald Eagle	78.8
Average	79

Table 2. Data for Model with Epochs

4.2. Experiment 2

The next thing I tested was how my training images affected the accuracy of my models. Training and learning from the dataset's training pictures are the most important parts of making an AI engine, and having good training images is key to creating a successful AI engine. When testing my engine, I discovered that most of the inaccurate identifications resulted from pictures with birds flying, while the pictures with birds perching(staying still) yielded high accuracy. To improve my AI engine, I selected 5 random birds and found 5 random pictures of them flying. I then uploaded different numbers of pictures - 100, 200, 300- of the chosen birds flying to their respective database. Another set of pictures that my engine had trouble identifying was female and juvenile birds. Birds often have distinct differences in plumage between the male and female, and between adults and juveniles. To improve my AI engine, I uploaded different numbers of pictures - 50, 100, 200(total) - of female birds to my training dataset and retrained my AI engine using the new images. I then recorded the percent of accuracy using online images of female birds. I also selected 5 random birds and 5 random images for each bird as the first experiment.

By adding female bird pictures, I hoped to make my AI engine more well-rounded so that the user can identify female birds as well as male birds.

Table 3 and Figure 8 displays the increased accuracy of my AI engine after I added the images to the training dataset. The average accuracy increased from 60.7% to 86.6%. Therefore my experiment proves that I can improve my engine's effectiveness by adding more images of flying birds. Table 4 and Figure 9 display the resultant percentages of accuracy depending on different numbers of pictures of female birds. My data shows that by adding pictures of female birds to the training dataset, the AI engine can identify the birds more accurately, namely the female birds; almost doubling the accuracy from 32.4% to 64.5%. In conclusion, I can increase the effectiveness of my APP by adding more pictures of female and juvenile birds, because by doing that, the APP can not only identify the male birds but also their female and juvenile counterparts. Making sure the AI engine can identify the female and juvenile birds is crucial for there are countless female and juvenile birds in the world; not training the AI engine with appropriate images will cause the user to misidentify the bird, defeating the whole purpose of the APP.

	Barn Owl	California Condor	Blue Heron	Common Grackle	Common Loon	Average
50 pictures	0.00%	28.0%	75.8%	28.2%	30.2%	60.7%
100 pictures	47.6%	67.0%	58.4%	63.4%	67.2%	64.5%
200 pictures	16.8%	78.4%	77.8%	63.0%	88.8%	86.6%

Table 3. Average Percent of Accuracy vs. Number of Pictures of Flying Birds for Each Species

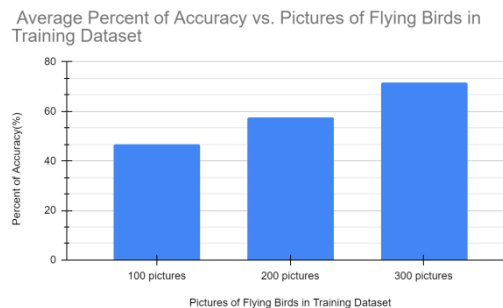


Figure 8. Average Percent of Accuracy vs. Pictures of Flying Birds in the Training Dataset

	Indigo Bunting	Mallard	Northern Cardinal	Northern Flicker	House Sparrow	Average
50 pictures	0.00%	28.0%	75.8%	28.2%	30.2%	32.4%
100 pictures	47.6%	67.0%	58.4%	63.4%	67.2%	60.2%
200 pictures	16.8%	78.4%	77.8%	63.0%	88.8%	64.5%

Table 4. Average Percent of Accuracy vs. Number of Pictures of Female Birds for Each Species

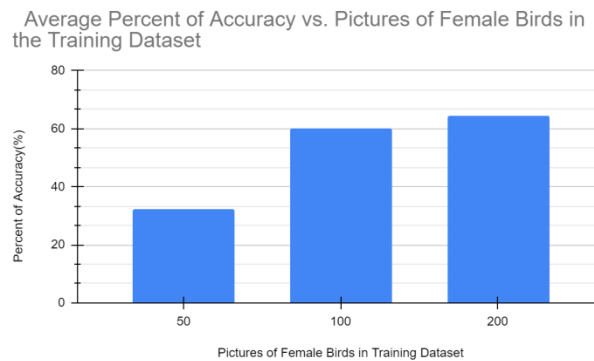


Figure 9. Average Percent of Accuracy vs. Pictures of Female Birds in the Training Dataset

After doing experiments and testing my AI engine. I found out that the model with epochs = 11 is the most accurate model with an average percent of accuracy of 79.0%. Which is highly accurate for identifying random images. Using flutter, I was able to create a working APP that is now published on google play for android users and the APP store for IOS users. The AI-identification part of the APP is also completely functional even without the internet; this feature is achieved by implementing a pre-trained AI Model instead of on-device training. This method means that the users do not have to train their model when they first download the APP since the APP already has a model trained previously by me. This method not only circumvents internet requirements but also saves the user time and phone storage(the training database images take up a lot of space) from training the model themselves. My AI engine can also be easily improved by adding pictures of flying birds and female/juvenile birds. Although it may seem like a lot of pictures, my bulk-image downloader allows me to find hundreds and thousands of quality images within seconds. Therefore expanding and improving my database is simple and doable.

5. RELATED WORK

Ferriera's group from Kimberley, South Africa used convolutional neural networks(CNN), a type of deep learning that automatically analyzes data like color, shape, and sizes(Ferreira et al, 2020) [5]. Similar to Ferriera's group, I also used a convolutional neural network when creating AI_birder. And both networks were able to achieve high accuracy when identifying birds. However, I also applied my AI engine to an APP so that the model is more accessible to other people. For even though everyone can copy the code, not everyone can collect a huge amount of pictures for the data and have a lot of time to train the engine. Using an app, on the other hand, is easier and more accessible to most birders, for they do not need to train the model or collect data; all they need to do is turn on AI_Birder and take a picture of the bird they saw.

Al-Showarah and Al-qbailat from the Mutah University of Karak used a type of CNN called VGG-19(Al-Showarah and Al-qbailat, 2021) [6]. VGG-19 is a type of CNN that is pre-trained and can identify and distinguish different traits. Similar to VGG-19, the transfer learning platform I used also trains the model through deep learning. But instead of VGG-19, I used MobileNetV2. The main difference between the two networks is the number of layers they have, which significantly influence the accuracy of the model. VGG-19 is a network with 19 layers, while MobileNetV2 has 53 layers, making MobileNetV2 more effective and enabling it to identify and differentiate more traits.

Mario Lasseck from the Museum fuer Naturkunde Berlin, Germany worked on identifying bird calls(Lasseck, n.d.) [7]. Their work is similar to mine in that we both used CNNs to train our

model: I used MobileNetV2 and they used a DCNN from PyTorch. But a major difference between my work and Lasseck's work is that mine focuses on identification by looks/picture, while Lasseck's focuses on identification by bird calls/audio. Identification by sound is a brilliant idea and is very useful, for many times the birder can hear the bird but can not see it to take a picture. This is because birds often hide in vegetation or have good camouflage. Therefore identification by sound has a lot of potential and is a good improvement and feature to add to my app.

6. CONCLUSIONS

In creating AI_Birder, I sought to create a more efficient and viable way for birders to identify birds. In order to create a working APP, I used Flutter for my main user interface codes and the main structure of my APP. I used Firebase for the backend portion of my code, which is mainly the dataset containing the images and names for the bird species and usernames. For the AI engine, I used TensorFlow as the main library that contains the resources needed to create an AI engine. To test my project, I mainly tested the accuracy of my AI project. To test my AI engine, I selected five random birds in my database and found five random pictures of each of those birds online. I then uploaded the pictures to my app and recorded the results. When my first prototype did not perform adequately, I experimented with the number of times that my AI engine retrains itself using the dataset pictures; that number of times is represented by a variable called "Epochs". After changing the value of epochs and hours of training my code, I found that the model with Epochs = 11 (meaning the code retrains itself 11 times using all the dataset images) is the most accurate. With the accurate model, I can accurately identify perching birds. However, the preliminary engine originally could not identify some birds that are flying and the female and juvenile versions of some birds. This issue was easily fixed by adding pictures of flying and female/juvenile birds in the training database. As the average accuracy increased from 60.7% to 86.6% for flying birds, and from 32.4% to 62.5% for female/juvenile birds. AI_Birder solves all the above-mentioned problems. Birders from all over the world can download AI_Birder from the google play store or the APP store. Identifying birds with Artificial Intelligence is the future of birding, and all the birders in the world can accurately identify birds thanks to AI_Birder.

Although the overall accuracy of my AI engine is high at 79.0%, some birds are still hard for my engine to identify for many reasons: the dataset does not have enough images for those species, there are multiple other birds that look extremely similar, and the image the user uploaded could be blurry and unclear. The first two issues can be easily fixed by adding more images to the training database. However, the last one is hard for any engine to identify a blurry and terrible picture. A bird call feature can also be added to the AI engine, for many times the birder can only hear the bird, not able to see it.

Identifying by voice can be attempted to be implemented in the AI engine. For many times the birds might be covered by tree branches or hiding in a bush, obscuring the birds from the birders' vision. A special library is needed for the bird call feature and there are other things to consider, like background noises, etc. Nonetheless, having identification by calls is an interesting addition to the app.

REFERENCES

- [1] Al-Showarah, S. A., & Al-qbailat, S. T. (2021). Birds Identification System using Deep Learning. Retrieved January 2, 2022, from https://thesai.org/Downloads/Volume12No4/Paper_34-Birds_Identification_System_using_Deep_Learning.pdf
- [2] Amadeo, Ron - Feb 27, 2018 2:00 pm U. T. C. (2018, February 27). Google starts a push for cross-platform app development with flutter SDK. Ars Technica. Retrieved December 31, 2021, from

- <https://arstechnica.com/gadgets/2018/02/google-starts-a-push-for-cross-platform-app-development-with-flutter-sdk/>
- [3] Audubon Bird Guide App. Audubon. (n.d.). Retrieved February 21, 2022, from <https://www.audubon.org/app>
 - [4] Ferreira, A. C., Silva, L. R., Renna, F., Brandl, H. B., Renoult, J. P., Farine, D. R., Covas, R., & Doutrelant, C. (2020, July 26). Deep learning-based methods for individual recognition in small birds. *besjournals*. Retrieved January 2, 2022, from <https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.13436>
 - [5] Lardinois, F. (2014, May 13). Firebase adds web hosting to its database platform. *TechCrunch*. Retrieved December 31, 2021, from <https://techcrunch.com/2014/05/13/firebase-adds-web-hosting-to-its-database-platform/>
 - [6] Lasseck, M. (n.d.). Audio-based bird species identification with deep ... Retrieved March 20, 2022, from http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-2125/paper_140.pdf
 - [7] TensorFlow. (2019, November 23). TensorFlow lite now faster with mobile gpus (developer preview). *Medium*. Retrieved December 31, 2021, from <https://medium.com/tensorflow/tensorflow-lite-now-faster-with-mobile-gpus-developer-preview-e15797e6dee7>
 - [8] Prokop, Pavol, and Rastislav Rodák. "Ability of Slovakian pupils to identify birds." *Eurasia Journal of Mathematics, Science and Technology Education* 5.2 (2009): 127-133.
 - [9] van Lent, Michael, and John Laird. "Developing an artificial intelligence engine." *Proceedings of the game developers Conference*. 1999.
 - [10] Van Der Maaten, Laurens, Eric Postma, and Jaap Van den Herik. "Dimensionality reduction: a comparative." *J Mach Learn Res* 10.66-71 (2009): 13.
 - [11] Abdi, Hervé, and Lynne J. Williams. "Principal component analysis." *Wiley interdisciplinary reviews: computational statistics* 2.4 (2010): 433-459.
 - [12] Wold, Svante, Kim Esbensen, and Paul Geladi. "Principal component analysis." *Chemometrics and intelligent laboratory systems* 2.1-3 (1987): 37-52.
 - [13] Vanfretti, Luigi, et al. "A software development toolkit for real-time synchrophasor applications." *2013 IEEE Grenoble Conference*. IEEE, 2013.
 - [14] Birnbaum, Lawrence, Margot Flowers, and Rod McGuire. "Towards an AI model of argumentation." *Proceedings of the First AAAI Conference on Artificial Intelligence*. 1980.
 - [15] McKinney, Wes. "pandas: a foundational Python library for data analysis and statistics." *Python for high performance and scientific computing* 14.9 (2011): 1-9.

AN INTELLIGENT MOBILE APPLICATION FOR DEPRESSION RELIEF USING ARTIFICIAL INTELLIGENCE AND NATURAL LANGUAGE PROCESSING

Zhishuo Zhang¹, Yu Sun² and Ryan Yan²

¹Arnold O. Beckman High School, 3588 Bryan Ave, Irvine, CA 92602

²California State Polytechnic University, Pomona,
CA, 91768, Irvine, CA 92620

ABSTRACT

“What is an simple yet effective method to improve the mental health of individuals?” is the question that we chose to tackle [7]. The solution that we came up with was having a deep conversation with another person. From personal experiences, having deep conversations with another person seemed to be one of the most effective ways to keep someone's mental health issues under control and maintain a more positive outlook on life. Sharing similar experiences with another person can demonstrate to people that they are not alone and there is always someone who can relate to them and lead them down the right path.

In order to provide people with an easier method to have deep conversations with one another, we decided to create an application called Affinity, which was developed using Flutter [8]. In this application, users with various mental health issues will be able to talk with other users who have shared similar experiences. Users can connect to each other based on similar mental health issues, and they can engage in deep conversations with one another through a chat messaging system. We tested the results by providing twelve participants with two surveys. One survey measured a self-given score regarding the participant's levels of stress and anxiety before using Affinity as well as after one week of using Affinity, and the other survey asks participants to tally the number of conversation partners that shared at least one mental health issue or experience with them compared to the total number of conversation partners. The results we have found are that daily usage of this application will generally reduce levels of stress and anxiety, and the majority of the individuals that the application will offer as conversation partners will be able to connect to a user through at least one additional similar shared experience or mental health issue.

KEYWORDS

Artificial Intelligence, NLP, Mobile Application.

1. INTRODUCTION

Our topic is mental health issues, which we address by creating an application dedicated to improving the mental health of its users [9]. Mental health is something that has been emphasized more in recent years, as therapy is becoming more commonplace. Unfortunately, mass shootings have appeared more frequently on the news as well, in which the perpetrators of these shootings were often mistreated and/or did not seek or receive proper mental care in time. The benefits to

having a healthy mind is clearer thinking, increased productivity, and finding more enjoyment and motivation in living and accomplishing daily tasks. Other benefits can include reductions in stress and anxiety, as well as a boost in self-esteem [10]. On the other hand, there are some major consequences to having poor mental health. A lack of attention and care to one's mental health could result in relationship difficulties, lowered productivity, and extreme mood changes. In severe cases, it could potentially lead to self-harm and suicidal thoughts.

This topic is essential in our current world situation, as people around the world lost their jobs, became unable to see their loved ones and friends, and suffered various other setbacks due to the pandemic. As a result, these people may become stressed and anxious, in which individuals with chronic forms of these symptoms are at increased risk to develop substance abuse, tendencies for self-harm, and medical conditions such as cardiovascular and gastrointestinal issues [2][3]. During a time when mental health should be emphasized more than ever, we hope that this application can inspire positive change in its users.

There are some mobile applications that tackle mental health issues and aim to improve the mental health of their users. Certain applications emphasize therapy to improve the mental health of its users, and many therapy applications involve dedicated therapists and coaches that connect with these users. Other applications rely less on human interaction and more on self-reflection in the form of mood journals and gratitude journals. These journals work by reaffirming one's values and internalizing one's emotions to inspire positive change. Breathing exercise applications focus on certain breathing patterns and techniques in order to manage anxiety. An example of a breathing technique to reduce anxiety is diaphragmatic breathing, which involves pulling your stomach in when exhaling and helps to regulate breathing when feeling panicky or unable to take deep breaths. Some applications, such as Moodfit, include all of the aforementioned features. Moodfit also incorporates other tools such as depression/anxiety assessments, medication logs, and summary reports [1]. Although most of these applications seem to be free of any major noticeable issues, a possible issue with those that don't involve therapy or any other human connection is that they may not be as effective for some users. While some will be perfectly content with the use of mood and gratitude journals, others may instead seek a conversation partner to share a deep conversation with. With applications that do rely on therapy, some of the users may find it hard to connect to the therapists and coaches due to differences in cultural background or a lack of shared similar experiences. If therapists and clients are unable to easily understand or relate with one another, the overall experience of the client could be reduced greatly. At worst, it could cause some therapy application users to view therapy in a negative light and actively avoid it.

Our tool to tackle mental health issues is a smart device application called Affinity, which requires users to complete their profile, select from three options to identify with (stress, anxiety, and family problems), and choose from a list of users to connect with that appear based on the selected options. Users can optionally write a description of their mental illness in their profile as well. After choosing a user from a list, the user will be able to initiate a conversation with the selected user. Some existing mental health mobile applications do not allow any interaction with other people at all, such as applications that are solely used as a gratitude journal. Others that do involve interaction with other people, such as therapy, may have certified therapists and life coaches that connect with the users to help them through their mental health struggles. What sets Affinity apart from most other mobile applications that use therapy is that users of the application can help other users by acting as conversation partners and talking through their issues. While therapy apps often need specifically assigned therapists or coaches to help users, Affinity can become completely self-sufficient without the help of these therapists, as long as there is an active and dedicated userbase that is willing to reach out to one another [11].

Our results were proven through two experiments. For each of these experiments, twelve participants were chosen. The first experiment that we used is a survey to test the effectiveness of the application at improving mental health. Each of the aforementioned participants would give themselves a score of 1 to 10 regarding their levels of stress and anxiety, with 1 meaning no stress/anxiety and 10 meaning unbearably high levels of stress/anxiety. The participants would register their account, complete their profile, and initiate conversations with anyone that shared the same mental health issues with them. The issues that participants were able to check off and identify with were stress, anxiety and family problems. Then, the participants would use the Affinity mobile application for a week, in which they would check on the application daily and respond to any messages they had received. They would all use the application during the same time period, ensuring that they would have active users to communicate with. After a week of using this application, each participant will do the survey again and record another self-given score of their stress and anxiety levels from 1 to 10. The self-given scores of stress and anxiety from both before and after using Affinity will be compared.

The second experiment was designed for testing the features of the application, and it takes place after the first experiment is completed. The main feature that was tested was the matching of the users through the checkbox options and how effective it actually was at matching conversation partners that share similar experiences. With all the conversation partners that the participants had formed in the previous experiment, the participants would list at least five mental health-related issues that they have had. If the participant has felt like they could empathize or relate with any of the issues that their conversation partner listed, they would record it and tally up the number of conversation partners that they could relate with out of their total number of conversation partners.

The rest of the paper is structured in sections, labeled 2 through 6. Section 2 explores the challenges that had to be overcome during the process of coding the application and coming up with new features. Section 3 analyzes our solution to tackle mental health issues, a mobile application that allows sending messages to others with similar experiences, and it goes over both the general system and the specific components. In Section 4, we explain the experiments that were performed on the application to prove its effectiveness in daily usage. Section 5 brings up a brief summary of three related works and how each one of them compares and contrasts with our project. Finally, Section 6 offers a summary of the paper and possible next steps for the application.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Finding the best method

The first challenge we faced during the creation of our mobile application was finding the best method to get two users involved in a deep conversation with each other. An application that incorporated voice calls and video calls seemed like a possible solution, but voice and video calling would be a fairly difficult feature to implement. Furthermore, talking face-to-face with a complete stranger and immediately engaging in deep conversation could be off-putting and uncomfortable for some users. Therefore, we opted with the final implementation to be a chat messaging system [12]. The chat system allows users to still have deep conversations with each other, and this system comes with the added convenience of users being able to respond in their own free time. This eliminates the need to list out schedules and plan out a time period that works for both conversation partners, which simplifies the process of using the application.

2.2. Finding a system

The second challenge that we came across was finding a system that could specifically connect users with similar shared experiences together. At first, we wanted to include an AI in our application. Using the descriptions/bios that users type in their profiles to describe themselves, the AI would extract keywords and compare it with the extracted keywords of other users' descriptions in order to find the most compatible or similar conversation partners. However, with fairly little experience in using Flutter, implementing such a feature seemed far too ambitious. The connection of users as conversation partners was a vital part of our application, so we needed a simple yet reliable method to implement this. Eventually, we settled on the checkbox method, in which the user would be asked to choose multiple options out of a selection of mental health issues that they identify with when first completing their profile. Depending on which issues were checked off, the application would provide the users with a list of other users who have checked off at least one similar mental health issue to them.

2.3. Deciding which mental health issues to include

The final challenge we faced was deciding which mental health issues to include in the application. Because of the way that the similarity between users was compared, users would be limited in what mental health issues we could identify with. We wanted to have a number of possible mental health issues that was large enough so that users would be able to have distinctly different profiles based on what they selected, yet small enough that the user would be able to complete that section of their profile within just a few seconds. We originally considered having numerous options in the checklist to choose from, but we soon realized that many of the options that we came up with overlapped with each other, and a new user might be overwhelmed by such a large number of options. For our application, we came to the conclusion that having three options would be the best choice. Using the original ideas for checklisted options of mental health issues that we had, we categorized them into the main mental health issues of stress, anxiety, and family problems [13].

3. SOLUTION

The name for this app is Affinity. It is an application that was developed through Android Studio [14]. The main purpose is to create a platform for patients with various mental health issues to find people that have similar experiences with them to talk to. The user will first create a profile in which they enter their information. Then they will select from the list of mental illnesses that they suffer from. The user will also provide a broad description of their illness to the best of their ability.

The app will match different users through the options they selected. I'm also developing an AI that will identify the descriptions of the users to better match them. Once users are matched they can begin to chat privately about what they are currently suffering from and support one another. I will be seeking advice from mental health professionals to further improve the app. The impact and success can be measured through user feedback and also through interviews with patients to discuss their improvements in their mental health. This project will be able to bring a positive impact to the community as it aims to reduce the problem of depression and anxiety.

The idea for this project came to me during the beginning of the pandemic. Last year alone, the number of people that reported having symptoms of depression increased threefold, and this trend is still continuing to this day. I was heavily affected as I suffered anxiety throughout last year due to a personal event. I never felt so lonely in my life, and it got me into the habit of overthinking.

Even if it was the most insignificant setback, I would think of all the worst possible scenarios, and anxiety took away all my ability to feel joy to the point where I thought there was no purpose to my life. I don't know where I would've been if I continued down this path, but thankfully, someone noticed my abnormalities and had a deep conversation with me. I never knew how much it helps to talk with someone that shares similar experiences with you, and this person reignited my passion and love for life.

This gave me the idea to create an app that will help others that shared my past experiences but have not yet found anyone that they can talk to. Learning from my journey battling with anxiety, I know that the best medicine for someone suffering from mental health issues is to talk to a person that truly understands what they are going through. Hopefully, with this project, I will be able to provide comfort and further help those that are in need during these difficult times.



Figure 1. Overview of the project

The entire system of the Affinity application works through a combination of multiple Dart files. As Dart is the sole programming language of this application, Dart controls both the front-end and back-end. A major component of the application is a database called Firebase, which brings the users the feature of logging in and registering for an account. Firebase allows the storing of usernames and passwords, as well as all the information in user profiles and the messages inside conversations. Therefore, users are able to log in to their accounts and easily view the history of their conversations.

The first step that a user would have to complete upon downloading the application and opening it for the first time is registering an account. By moving to the register screen and filling out all the necessary information in the blanks, the user can create their account. The next step that users would take is completing their profile. The user profile asks for the user's birthdate, a bio/description about their mental health issues, and gender. Most importantly, the profile asks the user to select from a checklist of three issues (stress, anxiety and family problems). After users confirm their information, they would be able to update their bio and issues they are facing anytime in the settings. Finally, users will be able to interact with other users by having a list of users with at least one shared issue recommended to them as conversation partners. These conversation partners can be selected, which the users will be able to start conversations with through sending messages. Users will be able to see the usernames of their conversation partners in their home menu, view past messages in each of these conversations, and respond to these messages.

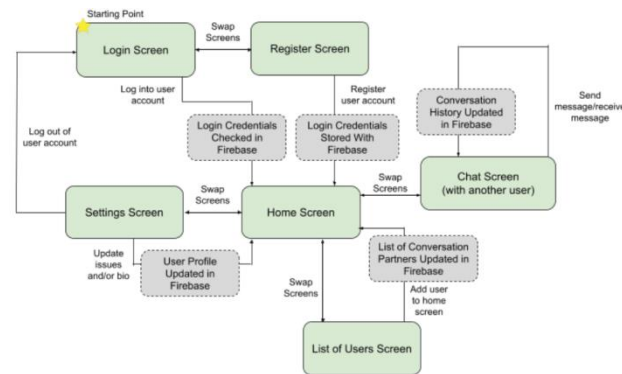


Figure 2. Overview of the progress

This application involves a login screen, a register screen, a home screen, a settings screen, a screen showing the recommended list of users as conversation partners, and a chat screen. In total, there are six main screens that are incorporated into this application. The home screen shows the list of conversation partners that the user has selected or has interacted with, and the settings page allows users to update their profile information. All of these screens make use of a database, which helps to make all the features run as intended. The database is one of the most crucial parts of the Affinity application. Without an implemented database, the application would not be able to store any information, and users will be unable to message one another. Firebase was chosen as the database to use for the application, as Firebase is very compatible with Flutter. Furthermore, the cost of using Firebase is free as long as the amount of stored data remains below a certain limit. Many Dart files in this application, such as the ones responsible for implementing the home screen and creating a new chat with a user, involve creating instances of Firebase Auth and Firebase Database.

```
Future<String> login() async {
  String outcome = 'Unknown Error';
  try{
    UserCredential userCredential = await FirebaseAuth.instance.signInWithEmailAndPassword(email: _email, password: _password);
    outcome = 'Successfully signed in ' + _email;
  } on FirebaseAuthException catch(e){
    if(e.code == 'user-not-found'){
      outcome = 'Error: No user found for that email.';
    }
    else if(e.code == 'wrong-password') {
      outcome = 'Error: Wrong password provided for that user.';
    }
    else{
      outcome = 'Unknown Error.';
    }
  }
  return outcome;
}
```

Figure 3. Screenshot of log in code

```
outcome = 'Successfully registered ' + email;
Navigator.pushReplacement(context, new MaterialPageRoute(builder: (BuildContext context) => HomePage()));
} on FirebaseAuthException catch(e) {
  if (e.code == 'weak-password') {
    outcome = 'Error: The password provided is too weak.';
  }
  else if (e.code == 'email-already-in-use') {
    outcome = 'Error: The account already exists for that email';
  }
  else if (e.code == 'invalidEmail') {
    outcome = 'Error: The email provided is too invalid.';
  }
  else{
    outcome = 'Unknown Error.';
  }
}
print(outcome);
Scaffold.of(context).showSnackBar(SnackBar(content: Text(outcome)));
```

Figure 4. Screenshot of successfully registered code

In the login screen, the user can fill out the email and password in the corresponding blanks. If the email has not been linked to an existing account or the password is incorrect, the application will send a corresponding message. In the register screen, the user would fill out the email, password, and username blanks. There are two password blanks, in which the second blank is to confirm the password. If the passwords do not match, the application will detect that the two variables holding the passwords do not hold equal values and will prompt the user to input matching passwords. If the email has already been used for a different account, the email is invalid, or the password is considered too weak (contains less than 6 characters), the Firebase authentication system will detect these cases and the attempt to register an account would be rejected. All of these checks for invalid input are accounted for in the Dart file responsible for handling the register page.

```
// Submit button function
void submitForm() async{
  print(submitForm());
  print(postPhoto());

  // This is to make sure it only is only happening once at a time.
  if(uploadLock == false){
    uploadLock = true;

    List<String> selected = checklist.getSelectedOptions();
    String selectedString = "";
    for (int i = 0; i < selected.length; i++){
      selectedString += selected[i];
    }
    if(selected.length != 0){
      selectedString += ",";
    }
    print(selectedString);

    String username = FirebaseAuth.instance.currentUser.displayName;

    var database = FirebaseDatabase.instance;
    final databaseRef = database.reference().child('profiles').child(FirebaseAuth.instance.currentUser.uid);
    databaseRef.set({'username': username, 'bio': bioDescription, 'birthday': prettyDate(), 'tags': selectedString, 'username': FirebaseAuth.instance.currentUser.displayName});
    // // Navigator.of(context).pushUntil((route) => route.isFirst);
    Navigator.pushReplacement(context, MaterialPageRoute(builder: (BuildContext context) => HomePage()));
    uploadLock = false;
  }
}
```

Figure 5. Screenshot of submit button code

```
List<String> getSelectedOptions(){
  List<String> selectedOptions = [];
  for(var option in checklistOption){
    if(option.isChecked){
      selectedOptions.add(option.value);
    }
  }
  return selectedOptions;
}
```

Figure 6. Screenshot of selected options code

Once the user either successfully registers or successfully logs into the application, the application will move to the home screen. If the application detects that the user has not yet completed their profile, it will display a button labeled “Complete Profile” that brings the user to the settings screen to complete their user profile. In the settings page, the user can fill in their birth date, bio, gender, and issues. All of this user profile information will be saved using Firebase, so the user will not need to complete the user profile again if they logout and log back in. In particular, the selected issues are stored as a list, and a method loops through all the selectable options and only adds the ones that have been checked off. When the user hits the “Submit” button, the user will be sent back to the home screen. After the application detects that the user profile has been completed, it will consider the user as a returning user and unlock two buttons in the home screen. This works by having one Dart file that is specifically for the home screen of a new user, and the regular home screen of a returning user is implemented through a separate Dart file. One button that is represented as a gear icon leads back to the settings screen to update the bio or issues of the user, in which the updated information is also stored in Firebase after pressing the “Update” button from the settings screen. The second button that is labeled with the “+” symbol moves the application to another screen that features the list of recommended users when pressed. The users are recommended based on whether they share at

least one similar issue, and their issues are listed right under their usernames. For example, if a user chooses stress and anxiety as the issues, the user would be recommended other users that have also marked either stress or anxiety as one of their issues.

```
DatabaseReference databaseRef = FirebaseDatabase.instance.reference().child("chatIDs/$chatID");
databaseRef.set({ 'user1': FirebaseAuth.instance.currentUser.uid, 'user2': otherUserID });

databaseRef = FirebaseDatabase.instance.reference().child("userChats/${FirebaseAuth.instance.currentUser.uid}/$chatID");
databaseRef.set({ 'otherUser': otherUserID });

databaseRef = FirebaseDatabase.instance.reference().child("userChats/$otherUserID/$chatID");
databaseRef.set({ 'otherUser': FirebaseAuth.instance.currentUser.uid });

Navigator.pushReplacement(context, new MaterialPageRoute(builder: (BuildContext context) => ChatRoomPage(chatID) ));

}

Future<Map<String, String>> GetUserProfiles() async{
  Map<String, String> userProfilesList = {};

  var userProfiles = await FirebaseDatabase.instance.reference().child("profiles").orderByKey().once();

  if(userProfiles.value != null) {
    var datapoints = Map<String, dynamic>.from(userProfiles.value);
    List<String> userTags = datapoints[FirebaseAuth.instance.currentUser.uid]['tags'].toString().split(',');
    print('UserSEDFSJEWJDL');
    print(userTags);

    for (var entry in datapoints.entries) {
      print(entry.key);
      print(entry.value);
      if(entry.key == FirebaseAuth.instance.currentUser.uid){
        print('pass');
      }
    }
  }
}
```

Figure 7. Screenshot of Database reference code

Once the user is moved to this screen of recommended users, the user has the option to choose any of these users to start a conversation with by tapping on a user's username. This works within one of the Dart files responsible for handling new chat pages, where a method adds a data item to a snapshot in Firebase that corresponds to a user. After tapping on a username, the user is brought back to the home screen as well. The home screen shifts between having a widget that accounts for no available chats and a widget that creates a functional chat list, and the widget for no available chats is only used when the application detects from Firebase that there are zero data items in the Firebase snapshot. The home screen allows the user to enter a chat screen with a specified user by tapping on the corresponding user's username. In the chat screen, the user can send a message by typing in the text box and pressing the button with a triangle pointing to the right. A Dart file is responsible for building the user interface and the chat bubbles of the chat screen through widgets. When the user either sends or receives a message in a conversation, the updated information is stored in Firebase and the widgets allow for the user to view the conversation history in an easily readable manner.

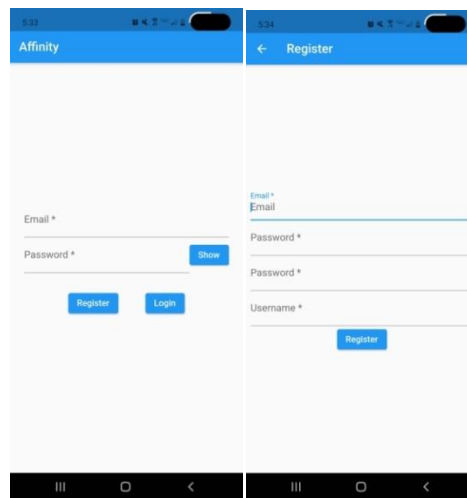


Figure 8. Screenshots of the Login and Register screens

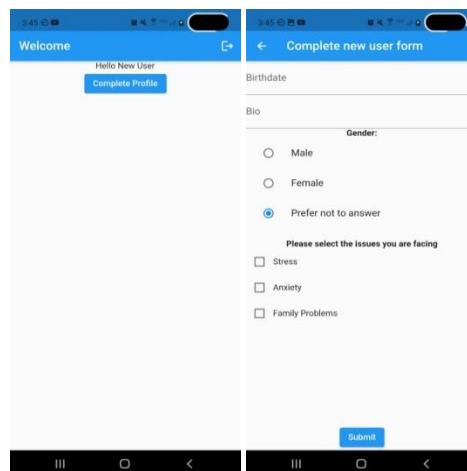


Figure 9. Screenshots of the Home and Settings screens (for a new user)

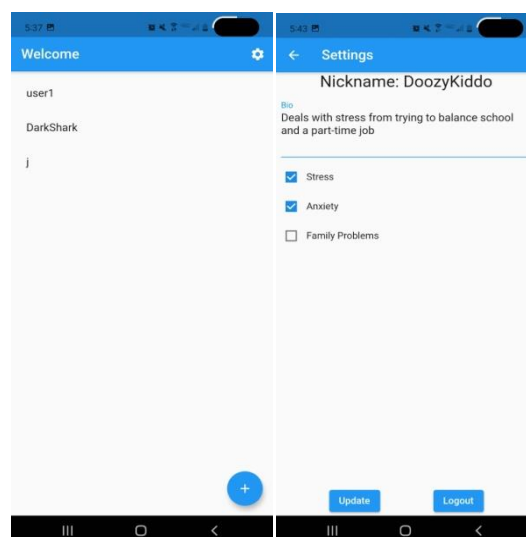


Figure 10. Screenshots of the Home and Settings screens (for a returning user)

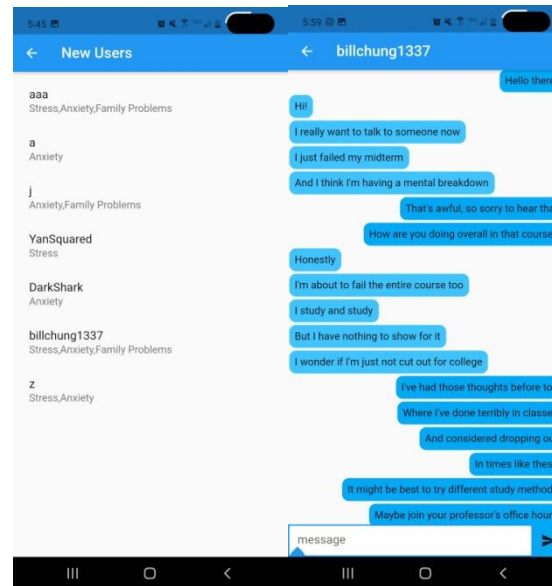


Figure 11. Screenshots of the list of users and chat screens

4. EXPERIMENT

4.1. Experiment 1

We aim to solve the problem of untreated mental health issues among people globally, especially those that have been greatly negatively affected by the pandemic, by having users of the Affinity application engage in deep conversations with one another. In doing so, these users can improve their mental health and lead happier lives. The main mental health issues that were focused on in this experiment were stress and anxiety. Our sample size of twelve participants allows us to account for any variability. In this experiment, the participants would first give their stress and anxiety levels a score from 1 to 10, with a lower score meaning less stress/anxiety and a higher score meaning more stress/anxiety. The levels of stress and anxiety are measured as separate statistics. The participants would then download the Affinity application, register their accounts, and complete their profile. During profile completion, they would choose from a checklist of mental health issues (Stress, Anxiety, and Family Problems) that they identified with, and they had to choose at least one. The participants would then initiate conversations with all users that the application recommended as conversation partners. The application recommends any user that shares at least one common mental health issue from the checklist of options in the user profile. For one week, the participants will check the application daily and respond to any messages they receive. After this week, they will give their stress and anxiety levels another score from 1 to 10.

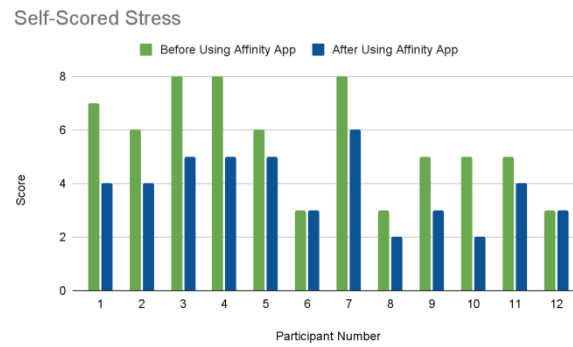


Figure 12. Result of self-scored stress

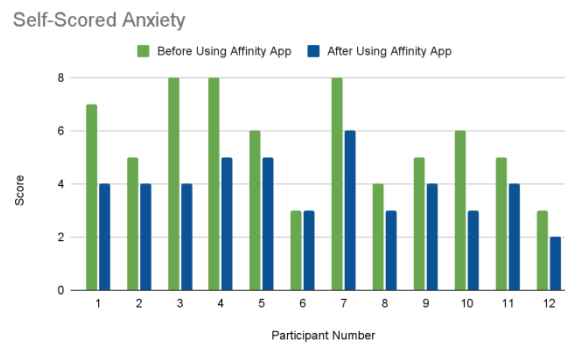


Figure 13. Result of self-scored anxiety

The results from this experiment revealed that all the participants' stress and anxiety levels were very similar. In the statistics from both before and after using the Affinity Application, the self-scored stress and anxiety levels were at most 1 apart for each participant. Although the results would have been very similar if the statistics of stress and anxiety levels had been collected together rather than separately, the participants were provided with more freedom to express themselves. A noticeable pattern in the results were that those with higher initial stress and anxiety levels tended to have greater reductions in stress and anxiety after a week of using Affinity. While participants with initial stress and anxiety levels of 7 and 8 were met with score reductions of 2 or greater, the majority of those with initial levels 5 and below only faced reductions of at most 1. Every participant's stress and anxiety levels either stayed the same or was reduced after one week of using the application.

4.2. Experiment 2

Through having people relate and empathize with one another mental health issues, our application tackles the problem of increased stress and anxiety during a difficult global situation. The Affinity application is capable of achieving this through its profile system, in which new users are asked to select from a checklist of three possible mental health issues when they first register their accounts. The application uses the selected options in the checklist to recommend other users as conversation partners. This experiment involves twelve participants in order to account for variability. Before this experiment, the participants will have already spent a week in the application engaging in conversations with all other active users who shared at least one common mental health issue as selected in their profile. Each of these participants would send a

list of at least five mental health issues/experiences that they most strongly identified with to all of their conversation partners. This list would exclude any of the selectable options of mental health issues in the user profiles, which tests if the participants can empathize with one another beyond the base application options. The participants would then view all the lists they have received, tally up the number of conversation partners that had at least one mental health issue or experience they could empathize with, and report this number along with the number of total conversation partners. Each participant also listed which options they chose in the user profile's checklist of mental health issues.

Participant #	Number of conversation partners that could be related with	Number of total conversation partners	Percentage of conversation partners that could be related with
1	9	11	81.8%
2	8	8	100%
3	10	10	100%
4	11	11	100%
5	7	8	87.5%
6	3	3	100%
7	10	10	100%
8	6	7	85.7%
9	9	10	90%
10	10	11	90.9%
11	7	7	100%
12	8	8	100%

Figure 14. Result of experiment 2

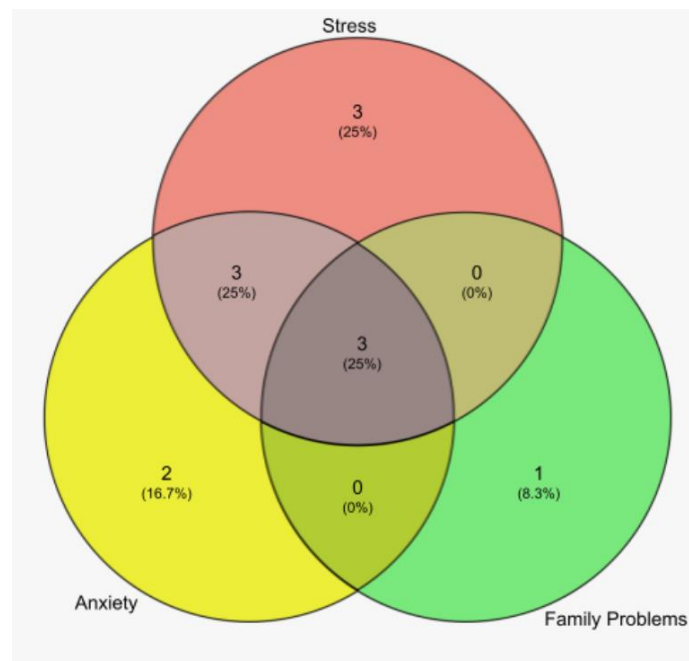


Figure 15. Distribution of mental health issues among participants

The results table of this experiment indicates that the majority of participants could relate to all of their participants in at least two mental health issues or experiences. One mental health issue comes from the list of at least five mental health issues that each user provided their conversation partners with, and the other mental health issue comes from the checklist options in the user profile that allowed the users to connect with each other in the first place. The lowest recorded percentage of conversation partners that could be related among all participants was 81.8%. Besides one participant that only had 3 conversation partners, all the other participants had at least 7 conversation partners. Therefore, each participant had a decent number of conversation partners they could relate with. According to the three-way venn diagram, the most common mental health issue was stress, followed closely behind by anxiety. Family problems were the least common mental health issue.

In the first experiment, there is an overall significant reduction in self-scored stress and anxiety levels among all participants after using the Affinity application for one week. Since having deep conversations with other people was believed to improve the mood and mental health of people, these were our expected results. These results of greatly lowered stress and anxiety levels prove the effectiveness of the application at improving the mental health of its users. In the second experiment, over half of the participants reported that every conversation partner had at least one shared mental health issue or experience outside of the options in the user profile. All participants shared a common mental health issue or experience, excluding the user profile options, with at least 80% of their conversation partners. This experiment proved the effectiveness of the application at bringing people with shared similar experiences together, as having one mental health issue in common seems to generally indicate having another mental health issue or experience in common as well. When combining the results of the two experiments, a significant correlation between the percentage of conversation partners that could be related with and change in stress and anxiety levels after one week could not be concluded.

5. RELATED WORK

This work focuses on a mobile application that acts as a mental health assessment, which includes the monitoring of stress, substance usage, sleeping, exercise, and diet. Information from the assessment is delivered to general practitioners for medical review [4]. This paper also chose to place a focus on a mobile application dedicated to the mental health of its users, and the experiments of this paper also resulted in the participants having overall improved mental health. The strengths of this related work lies in the ability of the application to record many different mental health symptoms, along with the expertise of general practitioners, to improve the mental health of the application's users. On the other hand, our work emphasized on our application users having deep conversations with other people that share the same issues or experiences in order to improve their mental health.

The primary focus of this work was to analyze the downloads and activity of mental health mobile applications during the pandemic and to view how they have changed from before the pandemic. The results of these analyses are that mental health mobile application downloads had a significant growth during the pandemic and applications were found to have no clear correlation between app quality and app popularity [5]. While the related work provided a greater emphasis on already existing applications, our work involved the creation of a new mobile application. The related work also spread its focal points across the quality, popularity, and clinical effectiveness of the tested applications, whereas our work focused solely on the effectiveness of the created application.

This related work explores the effectiveness of mental health services in mobile applications in preadolescents and adolescents. The results concluded that there was not enough evidence to

prove that these mental health services had a significant impact on mental health. The related work, just like our work, chooses to place its primary focus on the effectiveness of the mental health mobile applications. The related work, however, observes multiple preexisting mental health applications and how these kinds of applications will continue to grow in number over time. On the other hand, our work focuses solely on the effectiveness of our newly created application at improving its users' mental health and what can be done in the future for the application to become even more effective.

6. CONCLUSIONS

We propose a mobile device application called Affinity. The purpose of this application is to help those who may be suffering from mental health issues. Users would use Affinity by registering an account, selecting their mental health issues from a checklist of options, and engaging in conversations with other users that deal with similar mental health issues over a chat messaging system. By having deep conversations with other individuals, we believe that people can greatly improve their mental health and live happier lives.

In order to test our application, two experiments were conducted, in which both experiments involved the same twelve participants. One experiment involved a survey that required the participants to give their stress and anxiety levels a numerical score, initiate conversations and use the application daily for a week, then give another numerical score for their stress and anxiety levels after a week of using the application. A second experiment involved the participants sending all their conversation partners a list of at least five mental health issues or experiences they have suffered, then tallying how many conversation partners had at least one mental health issue or experience that they could relate with out of the total number of conversation partners. The results from the first experiment indicate that, in the majority of users, using the Affinity application would reduce the levels of stress and anxiety. The results from the second experiment prove that the current system that recommends other users as conversation partners is effective at bringing those with shared similar experiences together.

Currently, the most limiting factor of our application is the accuracy in which users are matched with other users as conversation partners. At the moment, we only have three possible mental health issues to match users with and would like to have more options to match users with. However, the current implementation of check boxes is far too inefficient to use as the sole long-term solution. There are a myriad of mental health-related issues to take into account, and even after adding as many options as possible, users may still not find issues that they can identify with. Because of this, we are seeking a more flexible way to implement the matching of users.

In the future, we hope to create an AI to add to Affinity that will be able to better match a user to other users [15]. We currently have users choose the checkbox options of stress, anxiety, and family problems as a simple yet effective implementation to match users who have checked similar options. However, we want to match the keywords in users' bios/descriptions using an AI so that we can offer users the best possible conversation partners.

REFERENCES

- [1] Prince, Martin, et al. "No health without mental health." *The lancet* 370.9590 (2007): 859-877.
- [2] Cocozza, Joseph J., and Kathleen R. Skowrya. "Youth with mental health disorders: Issues and emerging responses." *Juv. Just.* 7 (2000): 3.
- [3] Jones, Peter B. "Adult mental health disorders and their age at onset." *The British Journal of Psychiatry* 202.s54 (2013): s5-s10.
- [4] Reid, S.C., Kauer, S.D., Hearps, S.J. et al. A mobile phone application for the assessment and management of youth mental health problems in primary care: a randomised controlled trial. *BMC Fam Pract* 12, 131 (2011). <https://doi.org/10.1186/1471-2296-12-131>
- [5] Wang, Xiaomei, et al. "Investigating Popular Mental Health Mobile Application Downloads and Activity During the COVID-19 Pandemic." *SAGE Journals*, 7 Mar. 2021, journals.sagepub.com/doi/full/10.1177/0018720821998110.
- [6] Grist, Rebecca, et al. "Mental Health Mobile Apps for Preadolescents and Adolescents: A Systematic Review." *Journal of Medical Internet Research*, JMIR Publications Inc., Toronto, Canada, 25 May 2017, www.jmir.org/2017/5/e176/.
- [7] Boorse, Christopher. "What a theory of mental health should be." *Journal for the Theory of Social Behaviour* (1976).
- [8] Waage, Peter, and Cato Maximilian Gulberg. "Studies concerning affinity." *Journal of chemical education* 63.12 (1986): 1044.
- [9] Harnois, Gaston, and Phyllis Gabriel. "Mental health and work: Impact, issues and good practices." (2000).
- [10] Blascovich, Jim, et al. "Measures of self-esteem." *Measures of personality and social psychological attitudes* 1 (1991): 115-160.
- [11] Wampold, Bruce E., Scott A. Baldwin, and Zac E. Imel. "What characterizes effective therapists?." (2017).
- [12] Jennings, Raymond B., et al. "A study of internet instant messaging and chat protocols." *IEEE Network* 20.4 (2006): 16-21.
- [13] Bystritsky, Alexander, and David Kronemyer. "Stress and anxiety: counterpart elements of the stress/anxiety complex." *Psychiatric Clinics* 37.4 (2014): 489-518.
- [14] Esmaeel, Hana R. "Apply android studio (SDK) tools." *International Journal of Advanced Research in Computer Science and Software Engineering* 5.5 (2015).
- [15] Smith, Reid G., and Joshua Eckroth. "Building AI applications: Yesterday, today, and tomorrow." *AI Magazine* 38.1 (2017): 6-22.

A CONTEXT-AWARE VOCABULARY MANAGEMENT AND READING ASSISTANCE SYSTEM USING MACHINE LEARNING AND NATURAL LANGUAGE PROCESSING

Zhanhao Cao¹ and Yu Sun²

¹Troy High School, 2200 Dorothy Ln, Fullerton, CA 92831

²California State Polytechnic University, Pomona,
CA, 91768, Irvine, CA 92620

ABSTRACT

Through the increase in the popularity of online reading, many people rely on online dictionaries to further understand the text [1]. However, looking up a word manually is a great inconvenience as well as a form of distraction [2]. This paper develops a chrome extension to automatically detect the difficult words for each user, and provide the words' associated definition with a mouse hover. The chrome extension can be customized by adding and removing personal difficult words and personal easy words [3]. Also, the chrome extension offers a deeper level of analytic, including the system analyzing part of speech of the word, to further understand the definition of a selected word or sentence. The chrome extension is applied to a school/work setting in order to improve the working efficiency by providing a simple model to analyze the word definition; it is also useful for casual reading, especially to those that aren't fluent in English. Following the strict SDLC model, the end of the testing stage reflects that most of the users gave positive feedback to the chrome extension with most of the comments centered around convenience and accuracy [4]. Through alpha testing and a small sample of beta testing, the Chrome extension presents productivity improvement on difficult texts.

KEYWORDS

Chrome Extension, NLP, Cloud Computing.

1. INTRODUCTION

The topic is mostly centered around two issues: time management issues and language difficulty [5]. Since the return from the pandemic, I noticed the increase of reliance on the internet, and thus a large number of time people spend on the internet. I have also noticed the increase of distance between different cultures and groups, not just physically, but also the understanding of each other. Through this observation, I see the opportunity to make a useful chrome extension to address both topics. The chrome extension is extremely beneficial to students and workers who often read material from a webpage, as well as the English learners who wish to understand the material or study the language conveniently while reading an online text [6]. High school and college philosophy classes are greatly benefited from the chrome extension because the reading material in those classes is generally moderately difficult as well as covers a variety of areas. In a context where the understanding of the text is more important than the connotation and literary devices, computer software can greatly improve efficiency and understanding. In this case, this

chrome extension is extremely useful among philosophy students. Moreover, time management is becoming a huge issue for high school students, and searching up definitions manually or manually analyzing the context of a word is extremely time-consuming, distracting, and inconvenient. There is no benefit to manually searching up anything when it can be done efficiently and accurately through a computer program. Also, through the distance created by the pandemic, the inclusiveness of every community has become a great issue. This application improves the accessibility of non-fluent English speakers, making them feel more inclusive [7].

Some of the techniques that were previously used without the reading chrome extension are physical dictionaries, googling definition, or Quizlet made by other users [8]. Although all of those techniques usually provide a thorough and accurate definition, they are usually time-consuming and make the reader lose the flow of thinking while looking for the definition through a different resource. These proposals assume readers' willingness to invest their time to go through the inconvenient process to understand the material, which is rarely the case in practice [9]. These proposals are accurate in the areas that they are supporting, but they are not designed for online reading, yet they are still the primary resources for the readers. Similarly, tools such as POS tagging can be used to provide deeper analysis to the accurate definition in a given context, but the actions needed to go through a POS tagging and then a definition is extremely inconvenient and time-consuming. They are the working methods that are impractical. When referring to the resource outside of the web page, it is difficult to draw a connection to the text: they do not address the context of the word and they do not provide a repeated accessible route to the analytic such as hovering over the highlighted word. Also, it is a struggle for the users to manually find the words that they do not understand, which requires a deep understanding, but it is also a premise to the tools available. Another reading assistance such as dark mode reading is addressing a similar issue—but they do not support the understanding of the text nor assist the non-fluent English speaker to achieve a more inclusive community. There are many useful resources available on the internet to address similar issues, but none of them are useful specifically for reading an English article online.

To solve the problems that other tools do not address very well, a chrome extension that can analyze the difficult words automatically and provide easily accessible definitions, as well as an analytic of the definition, would solve the issues. The tool that I am creating addresses the issue starting from a simple interface, where the entire text will be scanned to find the predicted uncommon word and its definition [10]. The uncommon word is also customized with a simple right-click to select whether to add or remove a word from the common or uncommon list, where no words from the common word list will be highlighted and defined, and the rest of the uncommon words will be highlighted and defined. This feature creates accessibility that the chrome extension is useful to almost everyone. Another feature in the right-click drop down menu is its analytic ability, which is an essential ability when working with the definition. Words often have many different definitions across parts of speech and they would often have different meanings, and it's often difficult to know which definition to use when the word is not understood. Through some natural language processing and machine learning, the computer will calculate the part of speech of a word in a certain context, which provides an accurate definition of an uncommon word in a sentence, making reading easier and more convenient for the users.

First, I tried to implement the chrome extension into my daily life. Over the span of two weeks, I used the chrome extension in my philosophy reading assignments, and I saw a significant increase in productivity, and I was able to focus and understand the text much better. In my own experiment, I purposely applied all of the features that the chrome extension provides such as adding and removing common and uncommon words into my account or using deep analysis of a word in context. I have demonstrated an experiment that is applicable in a real-life scenario and it has proven to be useful. I also used public opinion by presenting the chrome extension in a game

competition, where this powerful program can help the gamers further understand a game's explanation or patch notes. The public's positive feedback on the chrome extension is reflected as the chrome extension ended up being top 3 out of over 500 competitors. Lastly, I gave the chrome extension file to some friends as a small beta testing sample, which provides valuable feedback on users' opinions. Through a survey that I gave, I asked specifically about convenience, increase in productivity, and its contribution to a more inclusive environment. Although minor issues were reported such as the speed of the program on a large website, I have received an average rating of 9.8/10 inconvenience, 8.9 in increasing productivity, and 8.6 to a more inclusive environment (all English learners presented 10/10 in this study. Through my own testing and some general feedback, the chrome extensions addresses all of the topics that it was meant to contribute positively to.

The above concludes the introduction of the chrome extension from its purpose to its application. Section 2 of the paper will detail the challenges I went through during the experimenting and designing stage. Correspondingly, section 3 will detail the process of conquering the challenges along with the solutions. To further understand the experiment, section 4 will discuss the design of my experiment and followed by section 5 to draw a comparison to some related work. Finally, section 6 will conclude the research as well as present some future improvements or expansion of the work in the future.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Observation in a real-life context

I have always wanted to write a computer program to assist human productivity and make the environment a more inclusive place, and the positivity of this topic has become more apparent ever since the pandemic. However, it is a challenge to come up with a solution that is generally applicable that has not been done. I did a lot of research regarding the issues of productivity and inclusiveness, and then it follows with the observation of my surroundings to come up with a program that I want to make. I also had to visualize the application of the program to determine the specific features that I want to include in order to achieve my goal, as well as find out the best form of program for the convenience aspect of the topic. Every aspect of the program design is supported by my observation of my environment's daily challenges, my own challenges, and my understanding of computer software.

2.2. Code/Server/Chrome Extension organization

The clarity of the organization becomes an issue when there are many aspects of the program that need to work together to make the final product work. Although there isn't any complicated class structure, the communication between each aspect of the program causes confusion when many similar features share a similar name. Also, errors would often occur under manual input. Encapsulation is still needed even when the methods are under the same class, but it becomes a challenge to separate them out based on each method's purpose for the final product. It's also necessary to understand the information that needs to be communicated through the chrome extension and the server. It is important for the connection of client-server communication because the chrome extension serves as an input to the server's action, yet it also presents the output that comes from the server. It's crucial to understand and organize the features of each class or method and how it is strictly communicated with the others.

2.3. Chrome boundaries and external library' s documentation

The program uses many pre-existing sources, but not all of them are generally applied and have clear documentation. The implementation of the popular libraries such as flask and firebase was not as challenging, but libraries such as beautiful soup for web-scraping and spacy for word Analytics have methods that I was previously unfamiliar with. Since some of the functions that I am looking for are niche, the less well-known libraries have to be used, and it was a large effort looking for an external library with the feature that I am looking for. However, the biggest challenge of resources comes from chrome plugin implementation. The limitation of the programming language that can be used for the Chrome plug-in forces me to use javascript as its main language. Also, chrome extension requires a strict format that it presents unclearly in its documentation. Moreover, the program fails if any part of the implementation does not fit the requirement of chrome plugins.

3. SOLUTION

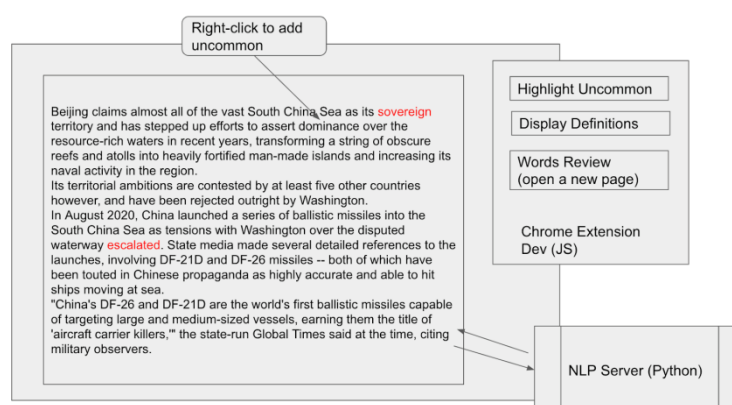


Figure 1. Screenshot of Chrome plugin

When using the Chrome browser, the chrome plugin will highlight the uncommon words for a specific user along with the words' definition when the mouse hovers over it with one click of a button. The button is located on the top right under the pop-up of the Chrome extension, and it is the interface that the user can interact with. In the webpage, the users can highlight the word and right-click to perform actions such as adding/removing a word to their specific common/uncommon list or they can run a deep analysis of the context they highlighted to understand the texts better. The function is achieved by creating a natural language programming server that includes all of the methods of the function, including web-scraping a website's HTML given a link, which can extract the raw form of the unfiltered text and structure that can be filtered and analyzed. Other functions such as adding/removing words or defining words are all defined in the server, and all of them can be accessed through a link associated with each method. The information dedicated to each user is stored in Firebase, where each user has a unique account and lists that are unique to them. The chrome extension part, it's written in javascript, and its function is to read the information in the webpage, mainly accessing and changing the information from the content section. It also defines the appearance of the program, from the way the words are highlighted to the pop-up that the user can interact with. Also, the drop-down from the right-click is also defined under the chrome extension code.

```

91 # define all unknown words in a website
92 @app.route('/scrape_website')
93 def define_all():
94     url = request.args['url']
95     definitions = {}
96     print(url)
97     a = scrape_words(url)
98     un_common_word = get_uncommon_word("", a)
99     count = 0
100     print(len(un_common_word))
101     #print(un_common_word)
102     for word in un_common_word:
103         print("#####")
104         print(word)
105         print("This is " + str(count) + " loops")
106         count += 1
107         definition = word_definition(word)
108         if(definition != None):
109             definitions[word] = definition
110     return definitions

```

Figure 2. Screenshot of code 1

The main function is completed by running many of the sub-functions after giving it some necessary information. The route /scrape_website is used for communication with the chrome extension. Concise documentation is used to distinguish the difference in each method, and printing lines are used for debugging purposes and to see the progress of the word definition. Similarly, other functions each as adding/removing common/uncommon words are written in methods given its necessary parameter; concise documentation is used everywhere throughout the code to manage the system properly. All of the methods are used either as a part of a bigger method, or it can be accessed by providing a route to it and connecting it to the Chrome extension through a listener.

```

182 # performing test
183 @app.route("/")
184 def homepage_test():
185     return("test")
186
187
188 # def build_database():
189 #     f = open("common10k", "r")
190 #     common_words = f.readlines()
191 #     f.close()
192
193 #     print(common_words)
194 #     for i in range (len(common_words)):
195 #         common_words[i] = common_words[i].rstrip()
196
197 #     words = db.collection(u'project_data').document(u'words')
198
199 #     words.update({'commonWords': firestore.ArrayUnion(common_words)})
200
201 # build_database()

```

Figure 3. Screenshot of code 2

In order for the program to work, the machine has to understand what words are considered to be basic words, which is done by storing the most basic 10k words in the database as the default common words. However, adding 10k words manually is not realistic, so the commented code above is used to build the database given a file of words, which is a tool that can be used by developers for changes in the database, supporting the communication between the server and the database.

```

1 from flask import Flask, request
2 from flask_cors import CORS
3 from PyDictionary import PyDictionary
4 import firebase_admin
5 from firebase_admin import firestore
6 from firebase_admin import credentials
7 from bs4 import BeautifulSoup
8 import urllib.request as urllib
9 import spacy
10 app = Flask(__name__)
11 cors = CORS(app)
12
13 cred = credentials.Certificate("ReadingExtensionDatabaseServiceKey.json")
14 firebase_app = firebase_admin.initialize_app(cred)
15 db = firestore.client()

```

Figure 4. Screenshot of code 3

Implementation of the structure of the server and functions such as dictionary and web scraping is properly implemented. These resources are crucial for the functioning of the methods as well as communication between two different components such as the server and the database. Spacy—a word analytic tool that supports the deep analysis function of the program—is implemented and applied properly to meet the function of the application, which is to help the user better understand the word given a context. Py-Dictionary—the default dictionary that provides the detailed raw definition of a word—is used to identify the definition and whether the input is a word. BeautifulSoup is used as a web scraping tool, which can extract most of the texts accurately given HTML, which can then be filtered to clean words in the body section and communicated with the content of the Chrome extension.

```

10 chrome.runtime.onMessage.addListener(
11   function(request, sender, sendResponse) {
12     if (request.type === "get_url") {
13       // alert(request.data);
14       // call
15       $.ajax({
16         type: "GET",
17         url: 'https://readingextension.laziestcactus.repl.co/scrape_website?url=' + request.data,
18         data: JSON.stringify({url: request.data}),
19         encoding: 'UTF-8',
20         success: function (resp) {
21           // console.log(resp);
22           // var res = JSON.parse(resp);
23           var myMap = resp;
24           var words = [];
25           for (var m in myMap) {
26             // console.log(m);
27             // console.log(myMap[m]);
28             words.push(m);
29           }
30
31           // var all = document.getElementsByTagName("p");
32           var all = document.getElementsByTagName("p");
33           for (var i=0, max=all.length; i < max; i++) {
34             highlight(all[i], words, myMap);
35           }
36           alert("Uncommon words identified!");
37         },
38         error: function(er){
39           JSON.stringify(er)
40           // alert(er.responseText);
41         }
42       });
43     }
44   }
45 );

```

Figure 5. Screenshot of code 4

The Chrome extension aspect of the program is implemented following strict formatting provided by Chrome's format documentation. Alert and console.log are used and commented on throughout the code segment for debugging purposes. The purpose of the code segment is to add a listener from Chrome and use it to control the functions in the server. Similarly, other aspects such as manifest, background, and popup are all implemented by following Chrome's format documentation and focuses on how it can be used to interact with the server and the webpage's content. The style of the popup and interface is also designed as a part of the plug-in.

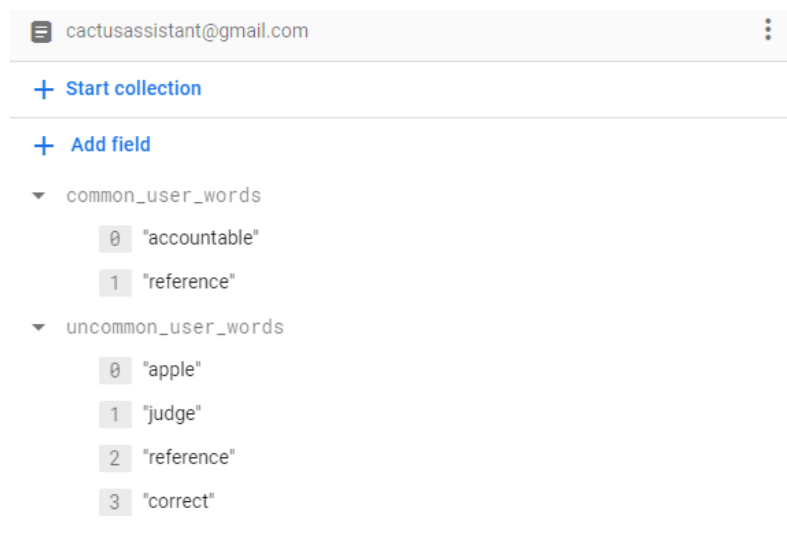


Figure 6. Screenshot of common word

Firestore is used as the database that stores the default information in the database, the users, and the user's information. Above is an example of a user's common word and user's uncommon word, and these can be added through a listener created in Chrome extension, which calls a function in the server to add/remove the word into the database.

same categories. Keep in mind that for team projects, the **judging** panel will have a higher expectations **commensurate** with the number of students on t

Projects must be submitted by the participants only. If a project submitted under an individual's name is discovered to have been prepared by a team of more than one student, the participant will be **disqualified**.

{"Adjective":["corresponding in size or degree or extent"]}

Vocabulary Assistant

Uncommon Words

Scan

Figure 7. Content style and the interface

Above is the content style and the interface that the user will interact with. There is also another interaction with a right-click drop-down menu and the context menu of the Chrome Extension. The buttons that the users can interact with are linked to listeners through Chrome extension, which will invoke functions in the server, and maybe change the user's personal content in the database.

4. EXPERIMENT

4.1. Experiment 1

An experiment is run on myself, where I use the app to monitor the changes in my productivity while doing philosophy homework. I time myself how long I spend on my philosophy homework for two weeks using the program and two weeks that I do not. Each week, I will be summarizing the main benefits and drawbacks of using or not using the program. The purpose is to find the areas that the app should try to assist by finding the weakness of reading from a webpage and see whether the problems can be addressed in the program. The criteria of success are measured through my feedback and the timetables.

With this chrome extension, I notice significant positive changes in my productivity and understanding as a result of being more focused and having convenient access to definition, which reduces my anxiety in an otherwise tedious assignment. Though the time saved is not a result of the time saved from looking up the word, but from the focus that it brings to me that I complete the work about 21.6% quicker as data collected in a span of two weeks. I have also noticed that I was able to understand the text better especially when the philosophy assignment is talking about an area that I was previously unfamiliar with, and I was able to learn the word much quicker with easy access to its definition. In comparison to reading on a paper, reading on a web page brings a greater distraction and frustration to look up for the definition of the word, which I have noticed that I often get carried over to do other things or lose the flow of thinking that is necessary for a philosophy assignment.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	avg.
35	38	31	34	42	n/a	n/a	36	36	31	32	37	n/a	n/a	35.2
29	28	31	24	25	n/a	n/a	26	23	31	29	30	n/a	n/a	27.6

Figure 8. Result of experiment 1

4.2. Experiment 2

The next test involves the other participants in my philosophy class, including two non-fluent English speakers. Also in a span of two weeks, I ask them to send feedback on what can be improved, what is the most helpful aspect of the Chrome extension, and a rating of the Chrome extension based on 3 categories: convenience, productivity, and inclusive environment. The rating will be out of 10, and the score of 5 is neither improving nor making it worse. The data will be analyzed by taking the average, and the data analysis will have a spotlight for the two non-fluent English speakers.

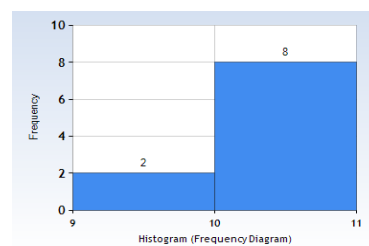


Figure 9. Result of experiment 2 (1)

This is a frequency graph of the rating received from the experimenters, where there are 8 scores of 10 and 2 scores of 9 for the convenience of the program. Based on the experimenter's feedback, there is no negative response regarding the convenience. Conversely, there is feedback complimenting how easy the program is to use for everyone. The two non-English speakers presented a score of 10, which is expected because the time saved for non-English speakers will be significant with this program.

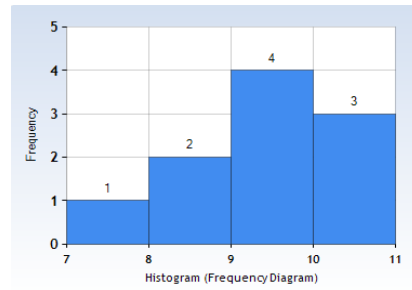


Figure 10. Result of experiment 2 (2)

Corresponding to the convenience aspect, all of the users responded with positive feedback for productivity. However, some users suggested that the program does not increase productivity, with further investigation with people that give a score of 8 or lower, I find that they usually do not look up the unknown words to understand the articles. Once again, the non-English speakers both presented a score of 10 for productivity.

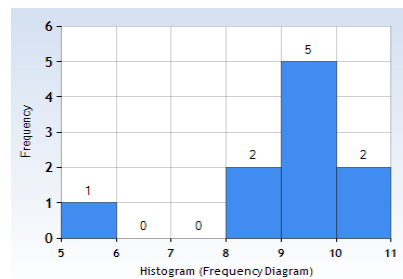


Figure 11. Result of experiment 2 (3)

For the category for an inclusive environment, none of the fluent English speakers presented a score of 10, and even one experimenter gave a score of 5 (no improvement from the program). However, most non-English speakers said that they do not feel the inclusive environment for themselves, but they do believe the program will help those who aren't fluent in English. Their suspicion turns out to be correct, where both of the non-English speakers once again present a score of 10.

Experiments from myself have presented its uses for convenience and productivity, where I can finish my assignments significantly quicker with better focus and less stress. Similar to the other fluent English speakers, I do not see the improvement for an inclusive environment, but I do think the perspective will be different from the non-English speaker. Through the experiment on myself, I can better understand the feedback from the other experimenters. Expectedly, almost every experimenter presented the benefit for convenience and productivity. However, I did not expect people who are extremely good at English to not see an improvement in productivity even though it's logical. As shown in the data, the Chrome extension is useful for everyone, but it is extremely useful for non-English speakers, who gave the extension a 10/10 in every category.

Unrelated to the topics presumed in the experiment, some feedback requested improvement in the interface and expanded the program into more areas. The deep analysis feature is rarely used but it has received positive feedback regardless.

5. RELATED WORK

The work analyzes students' habit of reading the digital text through student's ability to understand grammar visually [11]. The work stresses the difference in each student's reading ability. My work is a complement to this research, where my goal is to reduce the gap in students' reading ability on a webpage. Compared to the current research, they offered solutions through knowledge, where my solution for those that cannot understand texts very well is through technological assistance. Also, the research presents a deeper and more scientific understanding of students' understanding of the digital text, where I focused more on the application of the program and the creation of a program that is easy to understand and use.

This related work focuses on the natural language processing of a sentence that uses computer language to identify sentence type, annotation, and other broad aspects of language processing [12]. It presents a deep analysis of any texts pasted into its application. Compared to my work, this related work focuses more on the analysis of the sentence rather than the application to improve productivity and make a more inclusive community. This related work is more on representing the power of computers to understand the natural language. The strength of the related work is on the variety of ways to analyze the text and the off of website limitation by pasting the text into an application. The strength of my program is in the real-life application to understand the text, where the deep analysis feature in my program is similar to this related work.

The related work focuses on the Chrome extensions that help the students succeed academically in a variety of areas [13]. It focuses on the technological tool that can be translated to real-life applications through the Chrome browser. Compared to my work, this related work is more about discussing the technology's impact in our current world; on the other hand, my work presents a tool to help the readers on Chrome browser, which is also under the umbrella of this related article. This related work has the strength to help the readers understand the usage of Chrome Extension through reasoning. My work has the strength of solving a distinguished problem through technology rather than focusing on the general topic.

6. CONCLUSIONS

Through observation, I see the usage of making a Chrome extension that can help readers read articles from a webpage. The purpose of the Chrome extension is to make the users' life more convenient, increase the users' productivity, and create a more inclusive environment, especially for non-English speakers. The design of the Chrome extension is to automatically scan the uncommon words in an article through web scraping, and then highlight the words and provide the definition with a mouse hovering over the word. To further understand a word in a context, the program also provides a natural language processing aspect with the deep analysis feature, which can often lead to finding the correct definition of a word in a given context. Philosophy class is an example of the Chrome extension's application since it often reads articles from a webpage across a variety of topics [14]. I conducted an experiment on myself, seeing the Chrome extension's impact on convenience, productivity, and an inclusive environment. Through tracking the time I spent on my philosophy homework for 4 weeks, I can see an increase of productivity of about 21%, which is a result of the distraction that I can prevent from looking up the word and the convenient accessibility of the definition. Similarly, through an experiment conducted on my philosophy classmates, most people reflected positively on all aspects of the topic. Moreover, the

application is perfectly designed for English learners based on the experiment. The Chrome extension has proven itself to be effective to solve the challenge brought by reading many articles online, and the effectiveness varies inversely with the English fluency of the reader.

Even though the program has proven to be effective, there is an issue regarding processing speed especially when it comes to a website that includes a lot of uncommon words. Also, some websites have technology that prevents web scraping for security issues, in which the Chrome extension may not apply properly. The user interface is very limited corresponding with the concise feature, but more customizing options are limited to the user. There is also a limitation that comes from the cloud server and database that the Chrome extension is based on, which is a limitation to speed and the number of users.

The speed issue can be improved by giving the users a choice to scan less uncommon words in an article; it can also be improved by finding a quicker accessing method to the definition or through a better algorithm based on Big O notation for a large webpage. Definitions, servers, and databases can be customized to further complement the Chrome extension, increasing the stability of the program [15].

REFERENCES

- [1] Coiro, Julie. "Rethinking online reading assessment." *Educational Leadership* 66.6 (2009): 59-63.
- [2] Baron, Robert S. "Distraction-conflict theory: Progress and problems." *Advances in experimental social psychology* 19 (1986): 1-40.
- [3] Carlini, Nicholas, Adrienne Porter Felt, and David Wagner. "An evaluation of the google chrome extension security architecture." *21st USENIX Security Symposium (USENIX Security 12)*. 2012.
- [4] Rangunath, P. K., et al. "Evolving a new model (SDLC Model-2010) for software development life cycle (SDLC)." *International Journal of Computer Science and Network Security* 10.1 (2010): 112-119.
- [5] Claessens, Brigitte JC, et al. "A review of the time management literature." *Personnel review* (2007).
- [6] Stevenson, Julie S., Gordon C. Bruner, and Anand Kumar. "Webpage background and viewer attitudes." *Journal of advertising research* 40.1-2 (2000): 29-34.
- [7] Florian, Lani, and Kristine Black-Hawkins. "Exploring inclusive pedagogy." *British educational research journal* 37.5 (2011): 813-828.
- [8] Sanosi, Abdulaziz B. "The effect of Quizlet on vocabulary acquisition." *Asian Journal of Education and e-learning* 6.4 (2018).
- [9] Caspar, D. L. D., et al. "Proposals." *Cold Spring Harbor Symposia on Quantitative Biology*. Vol. 27. Cold Spring Harbor Laboratory Press, 1962.
- [10] Maltzman, Irving, Seymore Simon, and Leonard Licht. "Verbal conditioning of common and uncommon word associations." *Psychological Reports* 10.2 (1962): 363-369.
- [11] Walsh, Maureen, Jennifer Asha, and Nicole Spranger. "Reading digital texts." *Australian Journal of Language and Literacy*, The 30.1 (2007): 40-53.
- [12] Manning, Christopher D., et al. "The Stanford CoreNLP natural language processing toolkit." *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 2014.
- [13] Ok, Min Wook, and Kavita Rao. "Digital tools for the inclusive classroom: Google chrome as assistive and instructional technology." *Journal of Special Education Technology* 34.3 (2019): 204-211.
- [14] Chia, Robert. "Philosophy and research." *Essential skills for management research* (2002): 1-19.
- [15] Berg, Kristi L., Tom Seymour, and Richa Goel. "History of databases." *International Journal of Management & Information Systems (IJMIS)* 17.1 (2013): 29-36.

DEEP MULTIPLE INSTANCE LEARNING FOR FORECASTING STOCK TRENDS USING FINANCIAL NEWS

Yiqi DENG and Siu Ming YIU

Department of Computer Science, The University of Hong Kong,
Hong Kong, China

ABSTRACT

A major source of information can be taken from financial news articles, which have some correlations about the fluctuation of stock trends. In this paper, we investigate the influences of financial news on the stock trends, from a multi-instance view. The intuition behind this is based on the news uncertainty of varying intervals of news occurrences and the lack of annotation in every single financial news. Under the scenario of Multiple Instance Learning (MIL) where training instances are arranged in bags, and a label is assigned for the entire bag instead of instances, we develop a flexible and adaptive multi-instance learning model and evaluate its ability in directional movement forecast of Standard & Poor's 500 index on financial news dataset. Specifically, we treat each trading day as one bag, with certain amounts of news happening on each trading day as instances in each bag. Experiment results demonstrate that our proposed multi-instance-based framework gains outstanding results in terms of the accuracy of trend prediction, compared with other state-of-art approaches and baselines.

KEYWORDS

Multiple Instance Learning, Natural language Processing, Stock Trend Forecasting, Financial News, Text Classification.

1. INTRODUCTION

Stock trend prediction has always been a hotspot for both investors and researchers to facilitate making useful investment decisions, conducting investment, and gaining profits. Normal trend prediction tasks mainly take direct views on the stock prices. Based on stock prices, fundamental analysis [1], technical analysis [2, 3], and historical price time series analysis [4-6] have been used to aid in previous stock analysis. In addition to the direct quantitative information the numeric price brings on stock trends, financial news implies qualitative relations between daily events and their effect on the stock prices. Intuitively, people intend to buy stocks on hearing positive news and sell on negative news. Literature in [7-9] has also indicated that events reported in financial news play important roles concerning the stock trends in the financial market.

Using financial news to predict stock trends can be regarded as one text binary classification task. Take one trading day as an example, the trend of the stock is up if the closing price of the day is higher than the previous day, otherwise, it will have a downward trend. However, the uncertainty in daily financial news presents challenges to our normal financial text analysis. The financial news uncertainty on each day comes from two sources: uncertainty in the news amounts

happening on each day and uncertainty in the number of positive and negative news each day. For the uncertainty in daily financial news occurrence, financial news appears randomly most of the time. Sometimes there is no relevant news in one day while sometimes there can be more than ten or twenty news in one day (see Figure 1). As it can be seen from Figure 1, the number of news articles appearing each day, each month presents certain randomness. Sometimes there is no relevant news in a day, and sometimes there are hundreds of news articles in a day. In view of the uncertainty in the number of positive and negative news each day, the stock trend in a day is generally related to a certain amount of specific financial news instead of one piece. On a day with stock trends going up, it is quite unrealistic to consider that all the news within this day is positive, in that there can be lots of positive news, as well as some negative news, neutral or even unrelated news on this day.

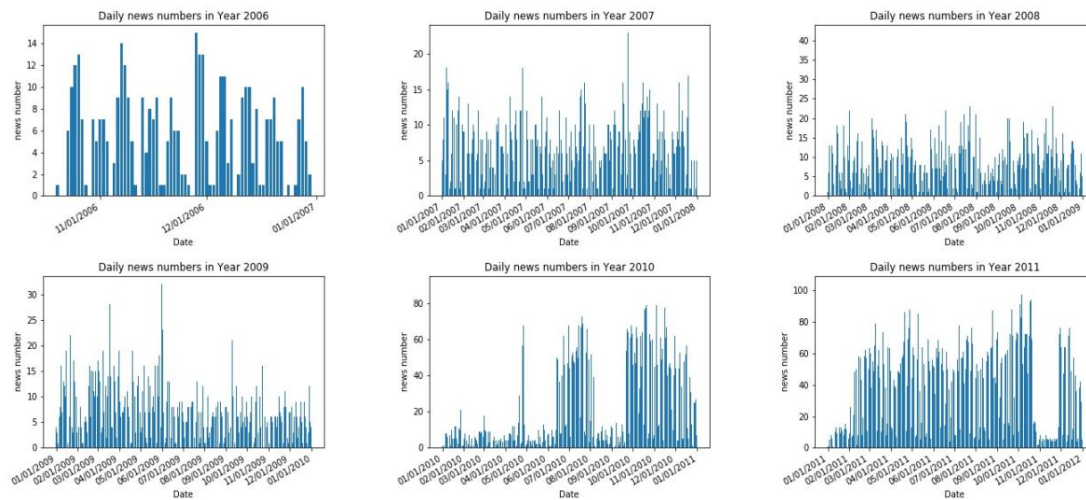


Figure 1. Random appearing news amounts in each year.

A good example can be illustrated through a direct look at the news published on a trading day (See in Table 1, Figure 2). There're almost 100 pieces of news on September 20, 2011. The stock closing price is lower than that of the previous day, which we considered a downward trend on this trading day. In all the news on this day, some are conveying a positive signal, say 'boost', 'increase', while some are sending neutral or even negative aspects ('cost', 'falter', 'low') to investors. The uncertainty on the distribution of positive and negative news within one trading day is seldom discussed in the previous studies. The reason behind this lies in the lack, intact labelings of single news. Indeed, considering the rapid changes in news amounts and random occurrences, the labeling of individual news (instance labels) is quite expensive and impractical. Besides, investors with different risk preferences treat and mark differently on the aspects of each piece of news without uniform standards, which also increases the difficulty of labeling individual news. However, the weak unknown news label relationships are indirect yet hardly negligible. Sometimes, a sudden piece of good news could alter investors' previously bearish decisions when investors do tradings. Therefore, ignoring news labels can do harm to precise stock prediction.

At the most of time, we can only gain class labels for groups of news in a day (bag-level labels) instead of each piece of news (instance-level labels). In the scenario of classifying stock trends using one day's news, labels for groups of news within a day (bag labels) are reflected as the annotation of stock trend, which can be clearly and easily clarified through the change of two consecutive stock closing prices. With full-annotated, complete daily group class labels,

supervised learning has dominated in previous literature [10-15] pertaining to stock trend forecasting tasks

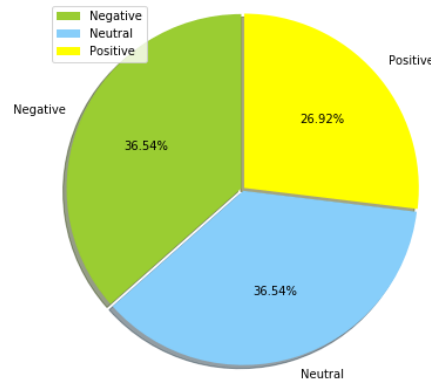


Figure 2. Sentiment values of news items on September 20, 2011, calculated by Natural Language Toolkit (NLTK)

Table 1. News posing on Sep. 20, 2011

	News
++ (pos)	China's Stocks Rise From 14-Month Low; Commodity Producers Gain.
++ (pos)	Obama's Home State Illinois Turns to China for Economic Boost.
++ (pos)	U.S. Gulf Crude-Oil Premiums Increase as Brent-WTI Gap Widens.
++ (pos)	U.S. Natural Gas Fund Premium at 0.31% on Sept. 19
...	
-- (neg)	China Endorsing Tobacco in Schools Adds to \$10 Trillion Cost.
-- (neg)	China Stock-Market Sentiment at Historic Low, Citigroup Says.
-- (neg)	Hurricane Irene Cost NYC at Least \$55M: Official.
-- (neg)	Oil Slides in New York on Speculation Demand to Falter; Brent Erases Drop.
...	
+ - (neu)	Oil Trades Near a Three-Week Low in New York; Brent Crude Climbs in London.
+ - (neu)	Short-Term Stimulus Won't Help U.S. in Long Run: Glenn Hubbard.
+ - (neu)	U.S. August Building Permits by Type and Region.
+ - (neu)	U.S. Solar Power Rises 69 Percent, Led by Commercial Projects.
+ - (neu)	China Jan.-Aug. Average Export Prices Rise 10.3%.
...	
Trend	DOWN TREND

through financial news. However, few current studies consider the randomness occurrence of financial news and include weak unknown news label relationships within a day in their modeling due to the lack of individual news (instance) labels.

Actually, as one type of weakly supervised learning algorithm, multiple instance learning (MIL) can be utilized to infer unknown news (instance) labels and the weak correlation between them. It helps ameliorate the limitation on the uncertainty of financial news mentioned above and models the financial news dataset at the instance, bag levels. Hence, in this work, we aim to adopt *Multiple Instance Learning* (MIL) [16] and consider the effects of financial news on stock trends from the perspective of Multiple Instance Learning. Related to the earlier work, in this paper we make the following contributions:

- A summarization of MIL principles used in scenarios of stock trend prediction using the financial news.
- Build up a novel MIL model to alleviate the problems of finance news uncertainty and insufficient individual news labels within one day when predicting the stock trends.
- Empirical evidence that our proposed MIL model can achieve impressive results on the S&P 500 stock index prediction, competing with other conventional neural architectures and previous MIL methods.

The paper is organized as follows. We review related work and earlier mainstream approaches in news stock trend prediction in Section 2. In Section 3, we introduce our proposed multiple instance learning (MIL) framework, with a description of how to represent news(instance), how to deduce the possibility of constituent news(instances), as well as the day(bag) vectors and day(bag)-level supervision. Section 4 presents our experimental results and discussion by using the financial news datasets. We also compare our method against previous approaches in this section. Finally, we conclude and summarize the paper in Section 5.

2. RELATED WORK

2.1. Multiple Instance Learning

Multiple instance learning was originally introduced by Dietterich et al., [17] in investigating the problem of drug activity prediction. In multiple instance learning, the training set comprises labeled ‘bags’, each bag is a collection of unlabeled instances. The exact label of every training instance is unknown, instead, the labels are provided for the entire bag. The appearance of multiple-instance learning is gaining interest by researchers, since in the real world, on account of tedious annotation by hand, limited label sources gained, there are a variety of classification problems where class labels are not complete at the instance level but only available for groups of instances. Alleviating the burden of obtaining limited-labeled datasets, Multi-instance learning has been successfully put into practice in areas of image classification [18, 19, 20], document modelling [21], event extraction [22], sound event detection (SED) [23], etc. The multi-instance learning approach also shows its feasibility in the application of text mining tasks. He Wei et al., [24] treat each document as a bag, the sentences in the document as each instance, to investigate text classification problems. They use Bag of Sentences (BOS) as text representation. Dimitrios et al., [25] adopt multi-instance learning on the problem of predicting labels for sentences given labels for reviews. They put forward learning classifiers to predict at the instance level instead of at the bag level. Based on instance-level similarity and group-level labels, a novel objective function ‘Group Instance Cost Function’(GICF) is proposed to encourage smoothness of inferred instance-level labels. Nikolaos et al., [21, 26] introduce a weighted multiple-instance regression (MIR) framework for document modeling and instance predicting aspect ratings in reviews. The MIR model captures meaningful structural information which is helpful for text understanding tasks and increases the performance of lexical and topical features for review segmentation and summarization. Stefanos et al., [27] present a neural network model for fine-grained sentiment analysis within the framework of multiple instance learning. Without the need for segment-level labels, their neural model is trained on document sentiment labels and learns to predict the sentiment of text segments. It can be indicated from the above literature that multiple instance learning, even though imperfect labels are employed, can nonetheless be used to create strong predictive models. However, in the field of financial news text analysis for stock trend prediction, little research has been done with this challenging yet potentially powerful variant of incomplete supervision learning.

2.2. Financial news for stock trend prediction

Financial news plays an important role with respect to the stock trends in the financial market. By means of deep learning and natural language processing (NLP), existing methods on stock market prediction by analyzing financial news have proven to be quite effective.

Financial news contains useful information in unstructured textual form. When representing each news title, it is non-trivial to extract semantic information and context information within each news title. The vector representation of words [28, 29] facilitates feature extraction from not only words but also sentences and documents. Classical methods such as averaging word vectors [30], training paragraph vectors [31] can be efficient, yet they have been indicated incapable of preserving semantics and gaining interpretation of linguistic aspects such as word order, synonyms, co-reference in the original news. To overcome this limitation, some improved representation techniques have been advanced in the following studies. Ding et al. [32] use open information extraction (Open IE) to obtain structured events representations in news. Later in [33], he put forward a novel neural tensor network to extract events in financial news. Get inspired by work [34], works such as [35, 36] adopt hierarchical structures to perform the classification: Hu et al. [35] adopt a hierarchical structure called Hybrid Attention Networks (HAN) to catch more features and help address the challenge of low-quality, chaotic online news. Liu et al. [36] extract news text features and context information through Bidirectional-LSTM. A self-attention mechanism is applied to distribute attention to most relative words, news and days. Ma et al. [37] develop a novel Distributed Representation of news (DRNews) through creating news vectors that describe both the semantic information and potential linkages among news events in an attributed news network. News vector representation has achieved state-of-art performances on various financial text classification tasks. A better text representation on news titles is vital in financial news analysis to capture features related to stock trends forecasting.

As predictive methods, deep learning models present high performances in traditional natural language processing tasks, namely, Convolution Neural Network (CNN) [38-40], Recurrent Neural Network (RNN) [41, 42], etc. In recent studies, authors in [30] propose a recurrent convolutional neural network (RCNN) model on stock price predictive tasks. Word embedding and sentence embedding are made as better embedding vectors for each piece of news. Huy et al. in [43] utilize a new Bidirectional Gated Recurrent Unit (BGRU) model for the stock price movement classification. Xu Y et al. [44] propose a stock price prediction model with the aid of news event detection and sentiment orientation analysis, through introducing Convolutional Neural Network (CNN) and Bi-directional Long Short-Term Memory (Bi-LSTM) in their predictive model. Most recently, a recurrent state transition model, integrating the influence of news events and random noises over a fundamental stock value state, is constructed in [45] for the task of news-driven stock movement prediction. A tensor-based information framework for predicting stock movements in response to new information is also introduced in [46]. From neural-network-based approaches to hierarchical structures-based models, to tensor-based networks, these methods have grown the mainstream and state-of-art techniques in the field of stock trend prediction through financial news texts.

3. METHODOLOGY

In this section, we describe the framework of our proposed multiple instance learning model. To further, we relate how to obtain news (instance) representations to better extract keywords and context information within the news, as well as how to apply multiple instance learning to address some of the pitfalls mentioned in previous parts, which are: the uncertainty of news relating to the randomness occurrences and the unknown annotation for each piece of news. The

model design has 4 stages: word embedding, news(instance) encoding, news (instance)-level classifiers, and bag-level representation and final classification.

3.1. Definitions & Formulation

According to the principle of multiple instances learning (MIL), given an input dataset D , the dataset D contains a set of labeled bags $B = \{B_1, B_2, \dots, B_M\}$, where each bag is a collection of unlabeled instances. In our multiple instance learning (MIL) framework, we regard news as instances and all the news that appears on that day as a bag. Now we consider the prediction of the stock trend over M trading days, M trading days represent M bags in D . Each bag $B_k, k = 1, 2, \dots, M$, contains n_k pieces of news, where each news text(instance) $n_k^i \in R^d, i = 1, \dots, n_k, k = 1, \dots, M$ is a d -dimensional vector learning from neural networks. With numerical labels $Y_k, k = 1, \dots, M$ derived from the daily stock close price, we are given bag labels for the stock trends each day. Then we have:

$$D = \{(B_k, Y_k)\}, k = 1, 2, \dots, M$$

where $B_k \in B$ and Y_k is a bag label assigned to day k . We assume binary classification in this paper, then we have $Y_k \in \{0, 1\}$, where 0 represents a downward stock trend for the day k , where the stock close price of current day k is lower than the previous day $k - 1$, and 1 shows the upward stock trend, where the stock closing price of current day k is higher than the previous day $k - 1$.

Previous studies mainly bring into focus on the relevance of news. They divide news into related or unrelated parts. Indeed, each piece of news conveys some of the information that drives the stock price trend, up or down. Therefore, our model attempts to predict how likely each piece of news is to move the stock upwards or downwards. The philosophy of multi-instance learning is to build classifiers to predict the labels of unknown bags by analyzing the label-known bags and its multiple instances. Based on that, in our work, we promote the relevance of news to the inference of individual news probability of being up and down.

3.2. Proposed Model

3.2.1. Word embedding

To obtain vector representation of each news text (instance), one key step is the use of embedding techniques. The embedding techniques map words into numerical vector spaces through an embedding matrix. Through the mapping, richer numerical representations of text input are created, enabling the deep multi-instance models to rely on these vector representations and improve performances in specific tasks. In our paper, the embedding takes a sequence as input, corresponding to a set of news titles. Assume that on a trading day $B_k, k = 1, 2, \dots, M$ with n_k news titles (instance), each news title n_k^i contains T_i words. $\{w_k^i\}^t, t \in [1, T_i]$ stands for the t th words in the i th news item of day k . We first embed the individual words $\{w_k^i\}^t$ to vectors through word embedding matrix:

$$L_w \in R^{d \times |V|}$$

where d is the dimension of word vector and $|V|$ is vocabulary size. Then the embedded vectors for word $\{e_k^i\}^t \in R^d$ is gained through

$$\{e_k^i\}^t = L_w \times \{w_k^i\}^t$$

The word vectors can be either randomly initialized or be pre-trained with embedding learning algorithms such as Glove and Word2Vec. Here, we adopt Glove [47] for better use of semantic and grammatical associations of words. In details, the Glove file that pre-trained 100-dimension word vectors on 6 billion tokens, 400K vocabulary, has covered most of the words in our news texts.

3.2.2. News(instance) encoding

Drawing inspiration from [34, 36], we exploit a Bidirectional-LSTM after word embedding to incorporate the contextual information from both directions for words. The recurrent structure in LSTM promotes the capture of context information. Compared with standard recurrent neural network (RNN), the gated mechanism in LSTM prevents the unbounded cell state and tackles the problem of vanishing/exploding gradient, which makes it more applicable in modeling semantics of long texts. Hence, we have the following computation of LSTM cells:

$$\begin{aligned}
 f_t &= \sigma(W_f[\{h_k^i\}^{t-1}, \{e_k^i\}^t] + b_f) \\
 i_t &= \sigma(W_i[\{h_k^i\}^{t-1}, \{e_k^i\}^t] + b_i) \\
 \tilde{C}_t &= \tanh(W_c[\{h_k^i\}^{t-1}, \{e_k^i\}^t] + b_c) \\
 C_t &= f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \\
 o_t &= \sigma(W_o[\{h_k^i\}^{t-1}, \{e_k^i\}^t] + b_o) \\
 \{h_k^i\}^t &= o_t \otimes \tanh(C_t)
 \end{aligned}$$

In LSTM, there are three gates, i.e. input gate i_t , forget gate f_t , and output gate o_t . For current input $\{e_k^i\}^t$ at time t and previous hidden state $\{h_k^i\}^{t-1}$ at time $t-1$, the calculation in forget gate f_t indicates the ability to forget old information. This gate decides what information should be forgotten or kept. Input gate i_t is derived from input data $\{e_k^i\}^t$ and previous hidden node $\{h_k^i\}^{t-1}$ through a neural network layer. \tilde{C}_t represents the cell state update value. Through the forget gate and the input gate, the cell state C_t is gained, with information of C_{t-1} and \tilde{C}_t . The output gate o_t decides what the next hidden state $\{h_k^i\}^t$ should be. $\{h_k^i\}^t$ is obtained from the output gate o_t and cell state C_t , where o_t is calculated in the same way as f_t and i_t . σ represents the sigmoid activation function.

The bidirectional LSTM contains the past and future context of the word. Through two hidden states \overrightarrow{LSTM} , \overleftarrow{LSTM} , information can be preserved, at any point in time, from both past and future. The forward \overrightarrow{LSTM} makes news be read from the first word to the last word, and the backward \overleftarrow{LSTM} allows information in news to flow from $\{w_k^i\}^{T_i}$ to $\{w_k^i\}^1$. Therefore,

$$\begin{aligned}
 \overrightarrow{\{h_k^i\}^t} &= \overrightarrow{LSTM}\{e_k^i\}^t, & t \in [1, T_i] \\
 \overleftarrow{\{h_k^i\}^t} &= \overleftarrow{LSTM}\{e_k^i\}^t, & t \in [1, T_i]
 \end{aligned}$$

We concatenated two hidden vectors $\overrightarrow{\{h_k^i\}^t}$ and $\overleftarrow{\{h_k^i\}^t}$ into

$$\{h_k^i\}^t = [\overrightarrow{\{h_k^i\}^t}, \overleftarrow{\{h_k^i\}^t}] \in R^{2 \times u},$$

which represents i th news title in the k th day(bag). n_k refers to the total amount of news items on k th day, and u is hidden units of LSTM.

Words within the news are not equally informative to investors. Investors usually pay more attention to keywords whenever they see a news story. Hence, we introduce an attention mechanism on top of the Bi-LSTM layer, so that it can reward the words offering critical information in our news(instance) representation. In details:

$$\begin{aligned} \{u_k^i\}^t &= \tanh(W_w \{h_k^i\}^t + b_w) \\ \{\alpha_k^i\}^t &= \frac{\exp(\{u_k^i\}^t)}{\sum_t \exp(\{u_k^i\}^t)} \\ n_{ik} &= \sum_t \{\alpha_k^i\}^t \times \{h_k^i\}^t \in R^{2u} \end{aligned}$$

We output the news(instance) vector as a weighted sum of the encoder hidden states. Then we compute the attention scores $\{\alpha_k^i\}^t$ and take softmax to get attention scores into a probability distribution. Finally, we take a weighted sum of values using attention distribution, obtaining the attention output as our news(instance) vector. The attention output mostly contains information from the hidden states that received high attention. Thus, the news(instance) vector is beneficial to aggregate the representation for informative words and better focus on keywords within the news.

3.2.3. News(Instance)-level classifiers

After we obtain the dense representations n_{ik} for each piece of news (instance) in bag k , the news-level classifiers, albeit labels are unobserved in the training set, are constructed to make predictions at the news(instance) level and infer the probability of each unseen individual news driving the stock up or down. For the classifiers of news(instances), we feed the news vectors n_{ik} into a one-layer MLP with sigmoid activation:

$$\widehat{p}_k^l = \text{sigmoid}(W_{news} n_{ik} + b_{news})$$

where \widehat{p}_k^l represents a real-valued score, demonstrating the predicted probability that an instance, i.e., one piece of news belongs to a particular class label. W_{news} , b_{news} are the parameters of new-level classifiers.

3.2.4. Bag-level vector representation and classifiers

In classical MIL problems, once setting up instance-level classifiers to get inferred instance labels in the bag, bag labels can be derived from gathering its individual instances labels through *The aggregation functions*. For commonly used aggregation functions, maximum operation, mean or weighted averaging have been chosen in many previous works of literature [22, 23, 25, 26]. In fact, the instances vectors don't need to be processed any further at the most of the time. However, for financial news text classification problems, additional steps for the bag feature extraction and bag-level representation are still necessary.

To make multi-instance learning more suitable in the classification problems of financial news text, we are going to take a novel approach to learn vector representations of days(bag). To be specific, after we deduce the news(instance) probabilities from news(instance)-level classifiers, we do not directly aggregate them into the stock trend probabilities of a day(bag) and get bag-level predictions. Instead, we encode the possibility of its composed news(instance) into the day(bag) vector representation, and then build the day(bag)-level classifiers on top of that. In this case, a class of transformations from instances to bags can be parameterized by neural networks, making MIL more flexible and being trained end-to-end. Hence, in the following steps we have vector-based representations for day (bag) k :

$$z_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \widehat{p}_k^i n_{ik}$$

Bag representation $z_k, k = 1, 2, \dots, M$ is based on the prediction probability of its component new(instance). Using these vector representations, we can get the predicted day-level probabilities through day-level classifiers. By comparing the predicted day-level probabilities against the actual day labels, we can compute a cost function, and the network is then trained to minimize the cost. Detailed design of our proposed multiple instance learning model is displayed in Figure 3.

4. EXPERIMENTS

4.1. Datasets

To conduct our experiments, the dataset we use is a set of financial news released by Ding et al. [32], between October 2006 and November 2013 in daily frequency. This dataset contains 106,521 news from Reuters and 447,145 news from Bloomberg. According to [32, 36], news titles alone are more predictive than adding news contents for trend forecasting tasks. Therefore, we extract the publication timestamps, the title for each piece of news from this dataset for our experiment. To catch the time period of the financial news, the historical stock price data for shares in Standard & Poor's 500 (S&P 500) index at the same period are also collected from Yahoo Finance to conduct our experiment on forecasting tasks.

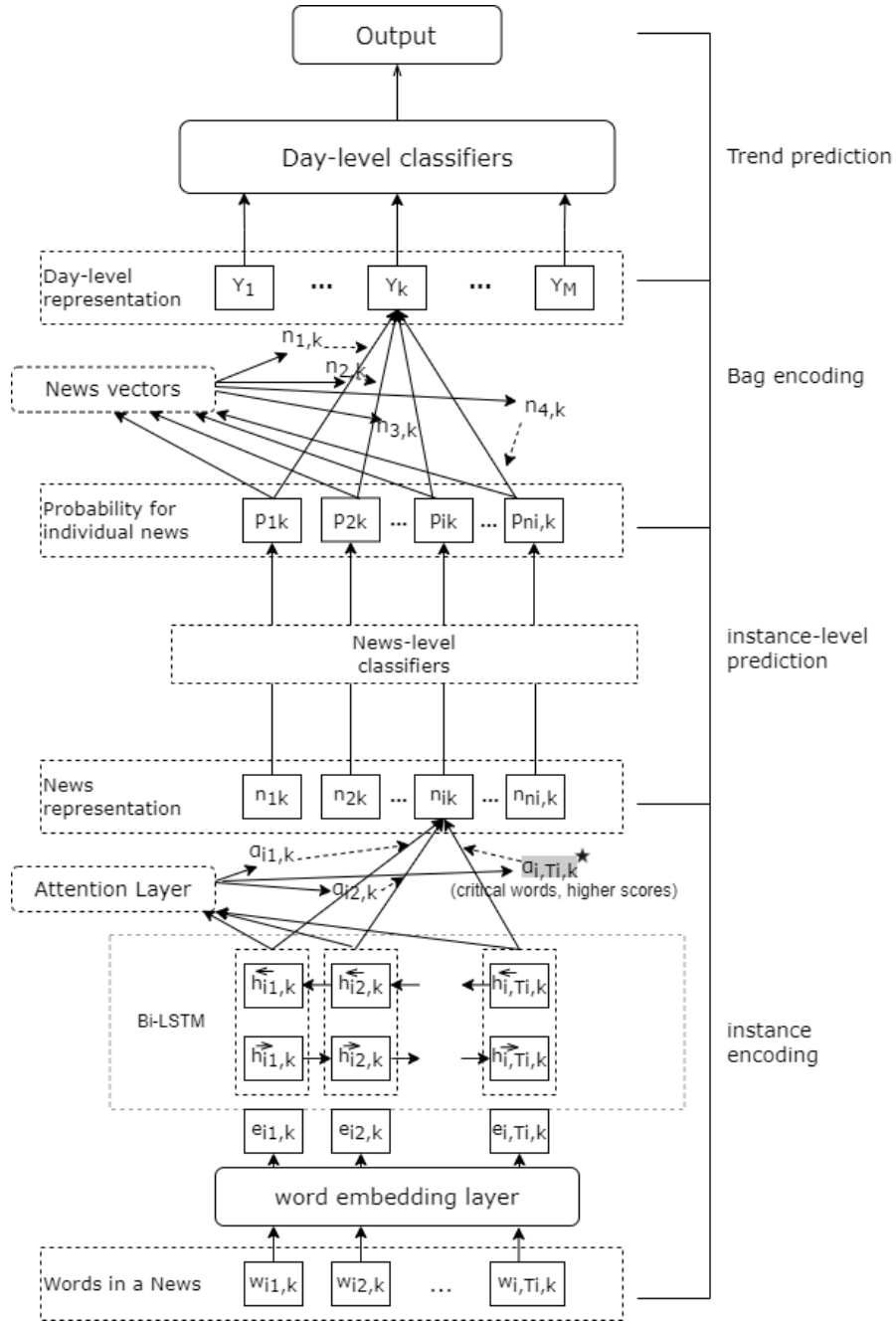
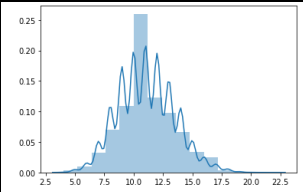
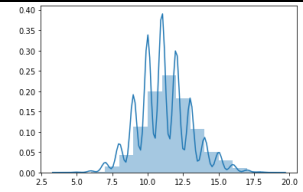
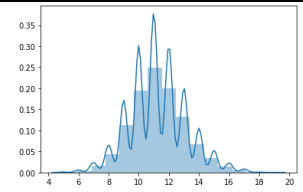


Figure 3. The overall architectures of our proposed Multiple Instance Learning Network

4.1.1. Data pre-processing

In the following experiments, we conduct a series of basic text pre-processings such as lowercasing, tokenizing each news and removing the stop words and infrequent words (appearing less than 5 times). Subsequently, we filter out the news without any correlation to stocks, making sure that all related symbols and company names are included. After filtering, we obtained a total amount of 63403 news. Following text pre-processing, we split the dataset into training, validation and test set. Summary statistics of the training, validation and test set are in Table 2.

Table 2. Statistics Details of Datasets

Datasets	Training	Validation	Test
Time period	20/10/2006-27/06/2012	28/06/2012-13/03/2013	14/03/2013-20/11/2013
News amounts	38454	13237	11712
Mean	11.078795	11.127219	11.261783
Std	2.369729	1.834530	1.885941
Min	4.000000	4.000000	5.000000
Max	22.000000	19.000000	19.000000
Distribution			

4.1.2. Experiment setup

To train our model, we use Adadelta algorithm as our optimization algorithm. Unlike the commonly used Stochastic Gradient Descent (SGD), the fixed global learning rate shared by all dimensions is less conducive to speeding up progress. The training progress can be slow when the gradient magnitude is small. Adadelta, as an optimization method using the adaptive learning rate, can converge faster and be used when training deeper and more complex networks. To further, we set the initial learning rate α as 0.1. Mini-batches of 32 is organized through the training process. In news (instance) representation, we adopt GloVe embeddings [47] as the pre-trained word embeddings, where the vector size of the word embedding is $|e| = 100$. The LSTM hidden vector dimensions for each direction were set to 50 and the attention vector dimensionality to 100. In the stage of the day-level (bag-level) prediction, the model convolves its input with news(instance) embeddings. For hidden layers within the news (instance) classifiers, the hidden units are set to 150. Additionally, to prevent overfitting, we adopt a dropout rate of 0.5 after both the instance level and the bag level.

4.2. Model Comparison

To evaluate our proposed model, in this section we set up a few baselines in contrast to our hybrid model. Our method is compared with preceding mainstream models and previous MIL models with differences in aggregation approaches for instances and bags (mean operation and encoding with instance-level results). For the sake of simplicity, the following notation identifies each model:

- BW-SVM: bag-of-words and support vector machines (SVMs) prediction model
- E-NN: structure events tuple input and NN prediction model, originally put forwards in Ding et al. [32]
- EB-NN: Event embedding input and NN prediction model in Ding et al. [33]
- S-NN: Following models in [30], taking the entire news corpus as the whole input. A mean operator is per formed on word embedding vectors within each piece of news as news vectors. On top of that, an averaging is added on news vectors in one day. A standard neural network is used as the prediction model.

- S-LSTM: The same vector representations as S-NN model, except to use LSTM as the prediction model instead of NN.
- Att-NN: Leverage a hierarchical attention network (HAN) analogous with the one in Yang [34]. Following Yang's HAN structure, we see each day as each document, news in one day as sentences in each document. A day representation is constructed by first building representations of news and then aggregating them into a day representation. Besides, a word-level attention layer and a new-level attention are added to differentiate keywords in each news and important news in one day. A standard neural network is used to make predictions.
- Att-LSTM: The same encoding and HAN's structure embeddings as Att-NN model except to use LSTM as the prediction model
- S-GICF: The multiple instance learning framework proposed in [25]. According to [25], an objective function
- ATT-GICF: Same multiple instance learning setup and GICF cost function as in S-GICF except to represent news(instance) by encoding news titles through Bi-LSTM and attention mechanism.
- MIL-s: Use averaging on word embedding vectors within each piece of news to represent news (instance) vectors. On top of that, the multiple instance learning model is exploited. News(instance)-classifiers are set to infer class possibilities of each individual news. Day(bag) representation is built using vectors representation and inferred stock trend probability of its component news(instance).
- MIL-rep: Our proposed model. Using Bi-LSTM and attention mechanism to encode each piece of news (instance). On top of that, news(instance)-classifiers are set to infer class possibilities of each individual news. Day(bag) representation is built using vectors representation and inferred stock trend probability of its component news (instance).

In order to compare the model performances, we use the classification accuracy as our evaluation and to prove if our approach can compete with the best state-of-art methods on our benchmark dataset. Table 3 shows the results compared with the above baseline models in the previous literature.

Table 3. Final results on the test dataset

Models	Test Accuracy (Max)
BW-SVM	56.38%
E-NN	58.83%
EB-NN	62.84%
S-NN	57.92%
S-LSTM	60.79%
ATT-NN	57.59%
ATT-LSTM	61.93%
S-GICF	58.52%
ATT-GICF	59.09%
MIL-s	60.23%
MIL-rep	63.06%

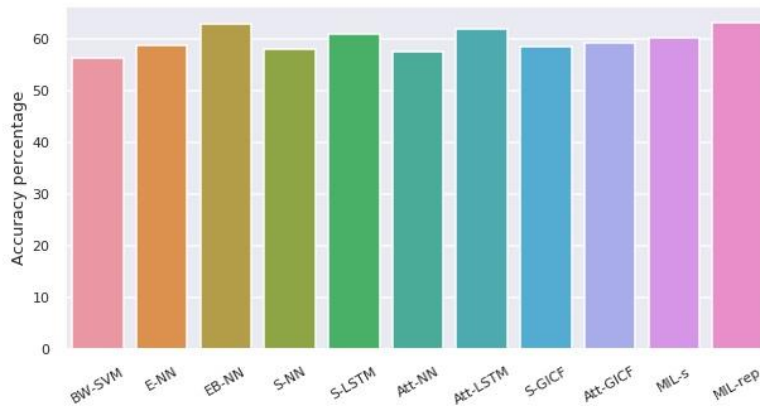


Figure 4. Performances in different models

4.3. Discussion

Through all the experiments above, our proposed framework, MIL-rep, can achieve a predominant accuracy of 63.06% (see in Table 3 and Figure 4), outperforming all the baseline models. Although event embedding (EB) in EB-NN is competitive, our proposed multiple instance learning framework is still powerful with event embedding (EB) in the extraction of financial news for stock trend forecasting. By making the comparison, we conclude the following discussions with respect to the following aspects:

Discussion on news representations: For the expression of news, in the non-MIL methods, we use simple averaging embedding (S) and encoding with Bi-LSTM, attention mechanism (ATT) to get the vector representation of news. In the MIL methods (S-GICF, ATT-GICF, MIL-s, MIL-rep), we take daily news as an instance. The representation of news is the instance representation. For MIL models, the results on the comparison between the models S-GICF vs ATT-GICF, MIL-s vs MIL-rep lead to the conclusion that better performances can be derived from the news(instance) representation of Bi-LSTM and attention (ATT) encoding, than the simple average embedding. In terms of the non-MIL method, in comparison with models S-LSTM vs ATT-LSTM, it can be inferred that ATT encoding is slightly better than simple averaging as input of the model, obtaining higher accuracy. Although something special in ATT-NN for case S-NN vs ATT-NN, we will discuss it later. It is not hard to see that the ATT representation for news is conducive to predicting the stock trend using financial news, especially in MIL models. This can be explained by the fact that the input to the model is organized in sequential contexts through Bi-LSTM. Keywords that show important trend signals in the news title are greatly extracted by the attention mechanism.

Discussion on predictive models: We mainly take LSTM as our primary predictive model in the experiments. In comparisons for non-MIL models, i.e., S-NN vs S-LSTM; ATT-NN vs ATT-LSTM, different predictive neural networks are constructed under the same news representation input. From Table 3, we can see that S-LSTM outperforms S-NN, and ATT-LSTM is better than ATT-NN. The structure in LSTM is effective at capturing long-term temporal features of the input sequences, which helps to enhance performances for the index prediction task.

Discussion on effects of multi-instance learning: The variable number of financial news (instances) in one trading day (bag) and the lack of news (i.e., instance-level) labels hinder us from inferring the stock trend labels of new days(bags). That's the reason why we move to

multiple instances learning to address news(instance) level predictions. There are two kinds of multiple instance frameworks we adopt in the proposed tasks. One is MIL composition using GICF cost function from [25], another is the MIL construction we put forward in this paper.

In S-GICF and ATT-GICF models, we use the MIL methods with GICF cost function. Based on the intuition that news(instances) pairs with higher similarity in one day will be more likely to assign the same labels, news(instance) similarity is exploited to read the combination possibility and assign news(instance) labels in a day(bag). After that, a simple averaging aggregation function is chosen to gather instance labels as the predicted day(bag) labels. From the possible assignment of its individual instance labels, GICF model is quite suitable to solve the news uncertainty, as previously mentioned, in both random appearing and unknown combination aspects. However, this approach is instance-label-oriented, without further vector operations at bag levels. The generalization capability of the averaging aggregation function is relatively poor. There is a risk the model is not adequately trained. Labels gathering may omit some potential information through the averaging aggregation, failing to identify complex patterns in the proposed task. Take simple averaging, ATT-encoding approach as instance representation and separately calculate pair similarity, the results in S-GICF and ATT-GICF are modest, with less than 60\% accuracy. Compared with non-MIL methods, the MIL composition with GICF in performances of S-GICF, ATT-GICF are also not as good as the LSTM structures in S-LSTM, ATT-LSTM.

In our proposed MIL construction, we determine to use a joint representation of a bag instead of gathering labels of instances. Predicted by instance-level classifiers, the probability of the class to which each instance(news) belongs is used to encode into the bag vectors. In this way, our proposed model can not only establish bag-level classifiers, allowing full, end-to-end training, but also include instance features of news uncertainty. According to the results of MIL-s and MIL-rep, the fitting effects of financial news are greatly enhanced by this embedding manner. The learning and generalization capability of our MIL models (MIL-s and MIL-rep) are improved, better than not only S-GICF and ATT-GICF models but also other non-MIL baselines.

Discussion on hierarchical attention in ATT-NN: In ATT-NN, two levels of attention mechanisms are used to identify keywords in each news text and significant news in a trading day. However, the performances of ATT-NN are apparently worse than the others. ATT-NN model has hierarchical structures, embed vectors in common with ours. In order to distinguish ATT-NN model from ours, we use the predicted class probability of news(instance) to encode the day representations. To get the predicted class probability of each news text, we set up the news-level classifiers. In contrast to our approach, Att-NN continues to encode the day vectors through GRU and attention mechanism on top of news vectors, without establishing news-level classifiers. Furthermore, in compliance with HAN's structure, we automatically assume that there exists a strong correlation between each news item within one day. All news items in a day are highly context-dependent, i.e., the same news may be differentially important in a different context. However, the assumption can be too strong for news, which leads to poor performances and inferior outcomes in the training of this model. Since in real life, the news is relatively independent of each other. There is no semantic, context relationship between news pairs on the same day as there is between sentences in one document. Therefore, HAN's structure is more suitable in document modeling than financial news analysis. For our model, we reduce this sensitivity and treat news every day as relatively independent variables, with their effects being related and synergistic to the day's prediction performances. The experiment results show that our approach has the merits of high learning efficiency, high classification accuracy, and high generalization capability.

5. CONCLUSION AND FUTURE WORK

In this paper, we point out the challenges in dealing with the financial news in the stock trend prediction, particularly, the uncertainty of financial news in terms of random daily occurrences and unknown individual labels composition. To address these issues, we propose to adopt the principle of multi-instance learning, and solve the problem of news-oriented stock trend forecasting from the perspective of multiple instance learning for the first time. With this end in view, an adaptive, end-to-end MIL framework is developed in this paper to achieve better performances. Experimental results on S&P 500 index demonstrate that our proposed model is powerful and effectively increases performance. Nowadays, multi-instance learning is receiving moderate popularity for its applicability in learning problems with label ambiguity. In a variety of application scenarios, some supervised learning methods have also been included or extended to the MIL setting. Nevertheless, there is still much need to explore in the field of financial forecasting for multi-instance learning, which could be a direction for our future research. In addition, in this paper, we only focus on the impact of news on stock trends. Taking more online data resources from social media into account in the trend prediction model is also a potential research site in the future.

REFERENCES

- [1] Tsai, M.-C., Cheng, C.-H., Tsai, M.-I., & Shiu, H.-Y. (2018). Forecasting leading industry stock prices based on a hybrid time-series forecast model. *PloS one*, 13(12), e0209922.
- [2] Gao, T., & Chai, Y. (2018). Improving stock closing price prediction using recurrent neural network and technical indicators. *Neural computation*, 30(10), 2833-2854.
- [3] Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, 53(4), 3007-3057.
- [4] Hiransha, M., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE stock market prediction using deep-learning models. *Procedia computer science*, 132, 1351-1362.
- [5] Luo, Z., Guo, W., Liu, Q., & Zhang, Z. (2021). A hybrid model for financial time - series forecasting based on mixed methodologies. *Expert Systems*, 38(2), e12633.
- [6] Qiu, J., Wang, B., & Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PloS one*, 15(1), e0227222.
- [7] Antoniou, A., Holmes, P., & Priestley, R. (1998). The effects of stock index futures trading on stock index volatility: An analysis of the asymmetric response of volatility to news. *The Journal of Futures Markets* (1986-1998), 18(2), 151.
- [8] Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of finance*, 62(3), 1139-1168.
- [9] Hussain, S. M., & Omrane, W. B. (2021). The effect of US macroeconomic news announcements on the Canadian stock market: Evidence using high-frequency data. *Finance Research Letters*, 38, 101450.
- [10] Kumar, B. S., & Ravi, V. (2016). A survey of the applications of text mining in financial domain. *Knowledge-Based Systems*, 114, 128-147.
- [11] Yoshihara, A., Seki, K., & Uehara, K. (2016). Leveraging temporal properties of news events for stock market prediction. *Artif. Intell. Res.*, 5(1), 103-110.
- [12] Xing, F. Z., Cambria, E., & Welsch, R. E. (2018). Natural language based financial forecasting: a survey. *Artificial Intelligence Review*, 50(1), 49-73.
- [13] Velay, M., & Daniel, F. (2018). Using NLP on news headlines to predict index trends. *arXiv preprint arXiv:1806.09533*.
- [14] Minh, D. L., Sadeghi-Niaraki, A., Huy, H. D., Min, K., & Moon, H. (2018). Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. *Ieee Access*, 6, 55392-55404.
- [15] Eck, M., Germani, J., Sharma, N., Seitz, J., & Ramdasi, P. P. (2021). Prediction of Stock Market Performance Based on Financial News Articles and Their Classification. In *Data Management, Analytics and Innovation* (pp. 35-44). Springer, Singapore.

- [16] Keeler, J. D., Rumelhart, D. E., & Leow, W. K. (1991). Integrated segmentation and recognition of hand-printed numerals (pp. 557-563). Microelectronics and Computer Technology Corporation.
- [17] Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2), 31-71.
- [18] Sudharshan, P. J., Petitjean, C., Spanhol, F., Oliveira, L. E., Heutte, L., & Honeine, P. (2019). Multiple instance learning for histopathological breast cancer image classification. *Expert Systems with Applications*, 117, 103-111.
- [19] Khatibi, T., Shahsavari, A., & Farahani, A. (2021). Proposing a novel multi-instance learning model for tuberculosis recognition from chest X-ray images based on CNNs, complex networks and stacked ensemble. *Physical and Engineering Sciences in Medicine*, 44(1), 291-311.
- [20] Zhu, W., Sun, L., Huang, J., Han, L., & Zhang, D. (2021). Dual Attention Multi-Instance Deep Learning for Alzheimer's Disease Diagnosis with Structural MRI. *IEEE Transactions on Medical Imaging*.
- [21] Pappas, N., & Popescu-Belis, A. (2017). Explicit document modeling through weighted multiple-instance learning. *Journal of Artificial Intelligence Research*, 58, 591-626.
- [22] Wang, W., Ning, Y., Rangwala, H., & Ramakrishnan, N. (2016, October). A multiple instance learning framework for identifying key sentences and detecting events. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (pp. 509-518).
- [23] Wang, Y., Li, J., & Metze, F. (2019, May). A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 31-35). IEEE.
- [24] He, W., & Wang, Y. (2009, September). Text representation and classification based on multi-instance learning. In *2009 International Conference on Management Science and Engineering* (pp. 34-39). IEEE.
- [25] Kotzias, D., Denil, M., De Freitas, N., & Smyth, P. (2015, August). From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 597-606).
- [26] Pappas, N., & Popescu-Belis, A. (2014, October). Explaining the stars: Weighted multiple-instance learning for aspect-based sentiment analysis. In *Proceedings of the 2014 Conference on Empirical Methods In Natural Language Processing (EMNLP)* (pp. 455-466).
- [27] Angelidis, S., & Lapata, M. (2018). Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association for Computational Linguistics*, 6, 17-31.
- [28] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- [29] Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems* (pp. 3294-3302).
- [30] Vargas, M. R., De Lima, B. S., & Evsukoff, A. G. (2017, June). Deep learning for stock market prediction from financial news articles. In *2017 IEEE international conference on computational intelligence and virtual environments for measurement systems and applications (CIVEMSA)* (pp. 60-65). IEEE.
- [31] Akita, R., Yoshihara, A., Matsubara, T., & Uehara, K. (2016, June). Deep learning for stock prediction using numerical and textual information. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)* (pp. 1-6). IEEE.
- [32] Ding, X., Zhang, Y., Liu, T., & Duan, J. (2014, October). Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1415-1425).
- [33] Ding, X., Zhang, Y., Liu, T., & Duan, J. (2015, June). Deep learning for event-driven stock prediction. In *Twenty-fourth international joint conference on artificial intelligence*.
- [34] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016, June). 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).
- [35] Hu, Z., Liu, W., Bian, J., Liu, X., & Liu, T. Y. (2018, February). Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 261-269).

- [36] Liu, H. (2018). Leveraging financial news for stock trend prediction with attention-based recurrent neural network. arXiv preprint arXiv:1811.06173.
- [37] Ma, Y., Zong, L., & Wang, P. (2020). A novel distributed representation of news (drnews) for stock market predictions. arXiv preprint arXiv:2005.11706.
- [38] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. EMNLP.
- [39] Lu, W., Duan, Y., & Song, Y. (2020, December). Self-Attention-Based Convolutional Neural Networks for Sentence Classification. In 2020 IEEE 6th International Conference on Computer and Communications (ICCC) (pp. 2065-2069). IEEE.
- [40] Guo, B., Zhang, C., Liu, J., & Ma, X. (2019). Improving text classification with weighted word embeddings via a multi-channel TextCNN model. *Neurocomputing*, 363, 366-374.
- [41] Lai, S., Xu, L., Liu, K., & Zhao, J. (2015, February). Recurrent convolutional neural networks for text classification. In Twenty-ninth AAAI conference on artificial intelligence. [42] Kotzias, D., Denil, M.,
- [42] Li, S., Zhang, Y., & Pan, R. (2020). Bi-directional recurrent attentional topic model. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(6), 1-30.
- [43] Huynh, H. D., Dang, L. M., & Duong, D. (2017, December). A new model for stock price movements prediction using deep neural network. In Proceedings of the Eighth International Symposium on Information and Communication Technology (pp. 57-62).
- [44] Xu, Y., Lin, W., & Hu, Y. (2020, December). Stock Trend Prediction using Historical Data and Financial Online News. In 2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom) (pp. 1507-1512). IEEE.
- [45] Liu, X., Huang, H., Zhang, Y., & Yuan, C. (2020). News-driven stock prediction with attention-based noisy recurrent state transition. arXiv preprint arXiv:2004.01878.
- [46] Li, Q., Tan, J., Wang, J., & Chen, H. (2020). A multimodal event-driven lstm model for stock prediction using online news. *IEEE Transactions on Knowledge and Data Engineering*.
- [47] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

CALIXBOOST: A STOCK MARKET INDEX PREDICTOR USING GRADIENT BOOSTING MACHINES ENSEMBLE

Jarrett Yeo Shan Wei and Yeo Chai Kiat

School of Computer Science and Engineering,
Nanyang Technological University, Singapore, Singapore

ABSTRACT

The potential of machine learning has sustained the interest of both academia and industry in stock market prediction over the past decade. This paper aims to integrate modern techniques such as Gradient Boosting Machines (GBMs) into a novel ensemble called CalixBoost which is a resource-efficient and accurate stock index predictor. Data comprising macro-economic metrics and technical financial indicators, as well as sentiment analysis of social media using a simple and fast but effective rule-based model are used in this paper. Other techniques include model tuning with Bayesian Optimization, temporal consistency analysis for invariant feature selection over random trial-and-error, feature importance and inter-feature relationships analysis using a unified game theory approach using Shapley values. Lastly, the model will be evaluated using a novel holdout method, viz. on two separate test datasets whose datapoints are collected under (i) normal economic activity and (ii) during a black swan (financial downturn). The experimental results show that our model outperforms previous methods and can achieve a good prediction performance with 84.88% accuracy, 0.0956 RMSE, 0.0573 MAE and 4.19% MAPE.

KEYWORDS

Gradient Boosting Machines; Time Series Prediction; Game Theory; Ensemble; Bayesian Optimization.

1. INTRODUCTION

Using Artificial Intelligence to predict the stock market is hardly a new science – the first use of neural networks in this field dates to ca. 1990 by Kimoto, Asakawa, Yoda, and Takeoka [1]. It has come a long way from 1970 when the Efficient Market Hypothesis (EMH) was first popularized which asserts that market prices always reflect all the information available, and thus, markets are generally efficient [2].

Academia has since, to considerable success, explored otherwise to “beat the market” using a variety of models such as ARIMA [3], ANN [4], SVM [5], LSTM [6] and GBM [7], and also challenged EMH in behavioural psychology studies such as social media sentiment analysis [8].

However, despite the proliferation of successful studies in stock market prediction, research conducted on integrating the application of these novel GBMs, which are also known as Gradient Boosted Regression Trees (GBRTs), as well as other best practices applied outside of research such as temporal consistency analysis in this field are few and far between at the time of writing.

Possible reasons include the perception that such machines are still in their infancy, and the hesitation in using deep learning models since they are sometimes viewed as black boxes whose outputs are difficult to decipher and whose hyperparameters are resource-intensive to tune.

To address these issues, we gather data from Yahoo! Finance for the New York Stock Exchange (NYSE) stock market index price data, U.S. Federal Reserve for macro-economic metrics and Twitter for social media posts from the most influential accounts. Technical financial indicators based on price data were created and sentiment analysis of the Twitter posts was conducted using a rule-based model specifically tuned to analysing social media posts using the VADER library; temporal consistency analysis is finally conducted for invariant feature selection. Next, an ensemble named CalixBoost is assembled using three untuned GBMs – CatBoost, LightGBM and XGBoost – to predict the stock price index. Bayesian Optimization is used to tune the ensemble. The model is evaluated using a holdout method of two separate test datasets whose datapoints are collected under different conditions: first, normal economic activity; and second, during a black swan (financial downturn). The SHAP library based on game theory is used to understand feature importance and inter-feature relationships.

The remainder of this paper is organized into the following sections: Section II reviews related literature, Section III illustrates data collection and pre-processing, Section IV describes the model design and tuning of CalixBoost, Section V shows the experimental results and comparisons with other models, and Section VI concludes this paper.

2. RELATED LITERATURE

Research articles which serve as inspiration for this paper but are not already referenced in later sections or warrant a lengthier discussion are listed below:

2.1. Leveraging social media news to predict stock index movement using RNN-boost

Chen, Yeo, Lau, and Lee [9] proposed a novel hybrid model, RNN-boost, built on top of sentiment analysis of social media posts which can produce good predictions of up to 66.54% relative to other traditional methods. Boosting algorithm Adaboost was used as well. This article is the main inspiration for the project to challenge the status quo by exploring hybrid models and tapping on microblogging data to predict price movements in the stock market.

2.2. On Stock Market Movement Prediction Via Stacking Ensemble Learning Method

Stacked ensemble along with engineered features such as difference can produce good classification results for predicting stocks at 78.10% accuracy [7]. This project will also utilize some of said features as technical indicators.

2.3. Ensemble methods foundations and algorithms

While ensemble is a technique used to assemble weak learning algorithms into an arbitrarily strong learner [10], this project will assemble a meta-classifier and regressor using already-accurate GBMs to achieve a final performance boost.

2.4. Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting

The GBM, XGBoost, proved to be an efficient algorithm with over 87% of accuracy for predicting stock price changes lookback of 60-day and 90-day periods [11]. The inspiration for using the area under curve (AUC) of receiver operating characteristic (ROC) came from this project which was used to evaluate the performance of XGBoost after model training is done. This project will instead use AUC ROC in feature selection first to get rid of features with high false positive rates to create a more resource-efficient model. It would be remiss not to point out that the authors have achieved a classification accuracy of over 99.9% for some stocks which they have raised their concerns of bias during model training. To mitigate such a problem, this project will instead use two datasets for a more robust evaluation of models – one under normal economic conditions (simply labelled as “test” dataset), and the other during a bear stock market (labelled as “black swan”).

2.5. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text

Hutto & Gilbert [12] proposed a rule-based model called VADER for sentiment analysis of social media posts. The authors contend that it is simple yet generalizes more effectively than other state-of-practice methods such as Support Vector Machine (SVM) algorithms, boasting an F1 Classification Accuracy score of 96%. This project will use VADER for sentiment analysis of Tweets for its speed and its attuning to social media posts.

2.6. A Unified Approach to Interpreting Model Predictions

Lundberg and Lee [13] proposed a novel unified game theory framework using Shapley Additive exPlanations (SHAP) values to explain predictions of machine learning models which is widely cited. This project will use SHAP to analyse the feature importance and variable relationships in all models.

3. DATA COLLECTION AND PRE-PROCESSING

3.1. Data Collection

Data used in this paper are summarized as follows:

3.1.1. New York Stock Exchange Index (^NYA)

To offer a fairer basis of comparison, this project will use the same stock index and data as previous projects. ^NYA comprises the major indexes in the United States and is also home to stocks from various industries both local and foreign. It is therefore a suitable weathervane for detecting international market sentiment. The features – Daily Open, High, Low, Close, Adjusted Close prices and Volume – are available in the dataset.

3.1.2. Twitter Posts (Tweets)

Research has shown that microblogging data, especially those from Twitter posts (Tweets), are useful in predicting stock market behaviour [14, 15, 16]. Additional advantages are that Tweets, with their 280-character limit, are relatively easy to analyse than say blog posts or Facebook posts. The Tweets of 25 of the most influential financial, political and news accounts were

scrapped. The accounts are: AP, Benzinga, bespokeinvest, BreakoutStocks, Business, CiovaccoCapital, CNBC, FXCM, IBDinvestors, LiveSquawk, LizAnnSonders, MarketCurrents, Markets, MarketWatch, Nyse, nytimesbusiness, Nytimes,realDonaldTrump, ReutersBiz, ReutersUS, Schuldensuehner, Stephanie_Link, Stocktwits, WSJmarkets and YahooFinance.

3.1.3. Macro-economic Indicators

Macro-economic indicators have proven to be useful barometers of stock returns [17], as well as good predictors during economic recession [18, 19]. Therefore, these data are also included in this project to make the models more robust even in a bear stock market. Data were downloaded from the U.S. Federal Reserve and include indicators such as interest rate and bank prime loan rates. The complete list of features is given in Table 1 in which Maturity represents the following: Mi: i-monthly, WKi: i-weekly, Yx: i-yearly, YiP: >i years.

Table 1. U.S. Macro-economic indicators.

S/N	Instrument	Maturity	Frequency
1	FF Federal Funds	Overnight	Daily
2	NFCP Nonfinancial commercial paper	M1, M2, M3	Business Day
3	FCP Financial commercial paper	M1, M2, M3	Business Day
4	PRIME Bank prime loan	N/A	Daily
5	DWPC Discount window primary credit	N/A	Daily
6	TB US government securities / Treasury bills (secondary market)	WK4, M3, M6, Y1	Business day
7	TCMNOM US government securities / Treasury constant maturities / Nominal	M1, M3, M6, Y1, Y2, Y3, Y5, Y7, Y10, Y20, Y30	Business day
8	TCMII US government securities / Treasury constant maturities / Inflation indexed	Y5, Y7, Y10, Y20, Y30	Business day
9	LTAVG US government securities / Inflation-indexed / Long-term average	Y10P	Business day

3.2. Data Pre-Processing

3.2.1. Sentiment Analysis using Valence Aware Dictionary and sEntiment Reasoner (VADER)

Sentiment analysis is conducted by using the five-rule VADER model [12] and implemented as the vader Sentiment library. VADER includes off-the-shelf support for emoticons like :D, emojis like 🍷 and 😊, slang words like “sux”, initialisms like “lol”, negations like “not good”, punctuation like “good!!!” and degree modifiers like “kind of good”.

VADER is constructed by an optimal list of lexical features specially attuned to understanding sentiment in social media content by generalizing into five rules according to the grammar and syntax of a given text:

- i. Lexical features are heavily adapted from dictionaries such as the popular Linguistic Inquiry and Word Count (LIWC) created to understand social media linguistic contexts [20, 21].
- ii. Social media-specific lexicon is added to the dictionary such as “LOL” and “WTF”.

iii. Using a crowd-sourced approach, human raters indicate both the sentiment polarity (positive or negative) as well as the sentiment valence (how intense of a sentiment a word is) of 9,000 lexical feature candidates.

iv. Lexical features with ratings aggregated across all human raters of ≥ 2.5 standard deviation are removed. Some examples of lexical features with their sentiment polarity and intensity taken from the VADER lexical dictionary [12] are given in Table 2.

Table 2. Examples of Lexical Features with their Sentiment Polarity and Intensity (Hutto & Gilbert, 2014).

Lexical Feature	Maturity
“okay”	+0.9
“good”	+1.9
“great”	+3.1
“horrible”	-2.5
“☹”	-1.5
“sucks”	-1.5
“sux”	-1.5

v. General rules are drafted as five sets of heuristics which capture the meaning of texts better by analysing word-order sensitive relationships (cf. bag-of-words) and are given in Table 3 [12].

Table 3. Five Heuristics Rules Used In Vader [12].

S/N	Rule	Example	Δ Polarity	Δ Valence
1	!	“Good!”	Same	Stronger
2	CAPS	“GOOD”	Same	Stronger
3	Degree Modifiers	“Very good”	Same	Stronger
		“Marginally good”	Same	Weaker
4	Contrasting Conjunctions	“X is good but Y is bad”	Negative	Stronger
		“X is bad but Y is good”	Positive	Stronger
5	Negation	Positive	Flipped	-

vi. The body of every scrapped Tweet is run through VADER to derive a “composite score”. This is a normalized, weighted composite sentiment score and is interpreted in Table 4.

Table 4. Interpretation of Sentiment from VADER Score.

Sentiment	Compound Score
Positive	score ≥ 0.05
Neutral	$-0.05 \leq \text{score} < 0.05$
Negative	score ≤ -0.05

3.2.2. Technical Price Indicator Engineering

Using the original daily prices and volumes from Yahoo! Finance, many other technical price indicators are derived. The full list of indicators is given in Table 5.

Table 5. Technical Price Features Engineered.

Feature (Symbol)	Formula
High (H)	Raw data from Yahoo! Finance
Close (C)	Raw data from Yahoo! Finance
Open (O)	Raw data from Yahoo! Finance
Low (L)	Raw data from Yahoo! Finance
Adj Close (AC)	Raw data from Yahoo! Finance
Volume (V)	Raw data from Yahoo! Finance
Amplitude	$(H_t - L_t) / AC_{t-1}$
Difference	$(C_t - O_t) / AC_{t-1}$
Intraday (I)	$O_{t-1} - AC_{t-1}$
Δ Adj Close (ΔAC)	$AC_t - AC_{t-1}$
Δ Volume (ΔV)	$V_t - V_{t-1}$
Intraday MA_n	n -day Moving Average of I
Δ Adj Close MA_n	n -day Moving Average of ΔAC
Δ Volume MA_n	n -day Moving Average of ΔV
$\pm \Delta$ Open	1 if $O_t - O_{t-1} > 0$, else -1
Daily Return (DR)	% ΔC
Cumulative Daily Return	Cumulative product of DR
H-L	$H_t - L_t$
C-O	$C_t - O_t$
RSI	14-day period of Relative Strength Index on C
Williams %R	$(H_{max} - C_t) / (H_{max} - L_{min}) * -100$
MA_n	n -day Moving Average of C
EMA_n	n -day Exponential Moving Average of C
MACD	$EMA_{12} - EMA_{26}$
BB High	21-day $C_{avg} + 2 * 21$ -day C_{std}
BB Low	21-day $C_{avg} - 2 * 21$ -day C_{std}
EMA	Exponential moving average of C with 0.5 decay
Momentum	$C_t - 1$
Feature (Symbol)	Formula
High (H)	Raw data from Yahoo! Finance
Close (C)	Raw data from Yahoo! Finance
Open (O)	Raw data from Yahoo! Finance
Low (L)	Raw data from Yahoo! Finance
Adj Close (AC)	Raw data from Yahoo! Finance
Volume (V)	Raw data from Yahoo! Finance

3.2.3. Lookback Features

This project will utilize time series forecasting for stock market index prediction, where the models will predict output y at time $t+1$ given in (1).

$$\hat{y}_{t+1} = f(y_{t-k:t}, \mathbf{x}_{t-k:t}) \quad (1)$$

\hat{y}_{t+1} is the model forecast while $y_{t-k:t}$ and $x_{t-k:t}$ are the observations of the output and inputs respectively over lookback window k [22, p. 2]. In this paper, $k = 60$ is used.

3.2.3.1. Feature Scaling: Quantile-based Scaling

Scaling is applied through standardization for all features. While this is not needed in practice for GBMs since they are decision trees which use gradient boosting and not gradient descent [23], scaling is still applied since RNNs like GRU and LSTM require it to achieve faster convergence rates [24].

To deal with outliers in the data, a quantile-based scaler is used to scale all features with the 25th quantile as 0 and the 75th quantile as 1 so that the scaling is not disproportionately influenced by very large outliers.

3.2.3.2. Feature Selection: Temporal Consistency Analysis

This paper conducts a study of time consistency across all features to weed out columns which exhibit poor temporal inconsistency using the Area Under Curve (AUC) of its Receiver Operating Characteristic (ROC) curve.

An ROC curve measures the performance of a binary classification system by plotting the True Positive Rate (TPR) on the y-axis against the False Positive Rate (FPR) on the x-axis. Every data point of the ROC curve indicates an observation in the confusion matrix [11].

Its AUC value is of interest and ranges from 0 to 1, and its results are interpreted as follows [25]:

- AUC = 0: A perfectly inaccurate test
- AUC = 0.5: A test with the results of TPR = FPR, represented by the diagonal line of $y = x$ in the ROC graph, and usually interpreted as the threshold which classification tests should minimally cross
- AUC = [0.7,0.9]: A considerably acceptable / excellent classification test
- AUC > 0.9: An outstanding classification test
- AUC = 1: A perfectly accurate test

The target threshold in this paper is set at AUC=0.5.

As there are only ~20 business days and thus ~20 data points per month per feature, a monthly analysis could not be conducted since comparing data between months will not be statistically significant enough to make a robust assessment.

Therefore, a thorough trimonthly analysis is conducted instead. All data is split into consecutive 3-month periods, then every non-overlapping combination is permuted with all others to determine the AUC score using scikit-learn.

3.2.3.3 Feature Importance Analysis

The Shapley Additive exPlanations (SHAP) framework is used to analyse the feature importance of all models using a unified game theory approach by introducing novel additive feature importance measures which helps to solve the struggle between accuracy and interpretability which has been a problem for deep learning or ensemble models [13, p. 4765]

SHAP introduces a meta-model called an explanation model when predicting a given model [13, p. 4765]. It simplifies the original model by using local methods termed as Additive Feature Attribution Methods (AFAMs) which are conceived to explain the prediction of the original model based on one input feature [13, p. 4766]. AFAMs have one explanation model that is a linear function of binary variables [13, p. 4766].

The individual relationship between inputs of a feature in the original model and the output function is simplified using combined cooperative game theory [13, p. 4768] to derive special Shapley values called SHAP values for every feature when conditioning on a given feature [13, p. 4769]. SHAP values are basically a unique set of simplified inputs which still obey the Shapley value properties of local accuracy, missingness and consistency [13, p. 4768], and thus can be used as a suitable proxy for every given feature [13, p. 4769].

Figure 1 shows a sample SHAP Explanation Model for a single feature. SHAP values are produced which illustrates the change in predicted output when conditioning on said feature. The model explains the change from predicted base value $E[f(z)]$ to current output $f(x)$ if all other features are unknown. To account for non-linear models or interdependent input features, SHAP averages ϕ_i values from all permutations.

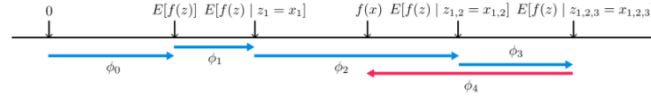


Figure 1. SHAP Explanation Model for 1 Feature [13].

The above can be illustrated using the local accuracy property of AFAM below:

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^M \phi_i x'_i \quad (2)$$

For a given input x , explanation model $g(x') = \text{original model } f(x)$ when $x = h_x(x')$ and where $\phi_0 = E[f(z)]$ which is the model output when we have no non-zero input entries [13, p. 4768].

Thus, with every non-zero input entry z_i , a unique Shapley value ϕ_i (SHAP value) is produced as an approximation in a conditional expectation function of the original model [13, p. 4769].

Since Figure 1 is a single explanation model of a single feature, when one iteration of that of all features are stacked together, we get Figure 2.

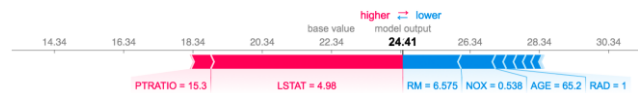


Figure 2. SHAP Explanation Model for 1 Iteration of All Features [13].

The final Explanation Model for N Iteration of all input features is simply N iterations of Figure 2's stacked on top of one another. The final explanation model is shown in Figure 3. The terms on the left of the graph are the input variables, and the x -axis represents the feature value. The importance of each feature is thus finally derived here.

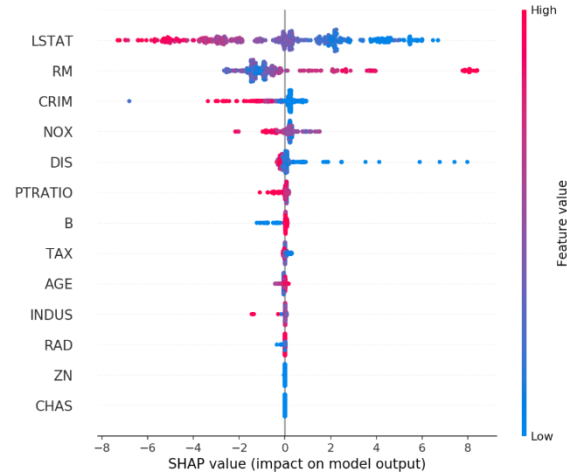


Figure 3. SHAP Explanation Model for All Features [13].

This paper uses the Python implementation “shap” written by the same authors of SHAP.

4. MODEL DESIGN AND TUNING

4.1. Model Design

This section introduces decision trees, ensembles, the three GBMs – CatBoost, LightGBM and XGBoost – which are decision tree ensembles, and our novel CalixBoost which is a stacked ensemble on the GBMs. Default hyperparameters were used in the GBMs.

4.1.1. Decision Tree

Decision Trees, also known as Classification Trees, are trees in which every node is labelled with an input variable, and all arcs stemming from a node represent the possible values of said input. When a set of inputs is given, the tree is traversed until it terminates at a leaf which gives the final class of this observation [26, p. 298]. The objective is to create a decision tree which gives the highest proportion of correct predictions, viz. a tree which minimizes the error of actual results vis-à-vis predicted results. An example of a decision tree is given in Figure 4.

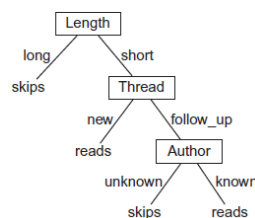


Figure 4. Example of a decision tree [26, p. 298].

4.1.2. Ensemble

Ensembles are a type of supervised learning composite model. Ensemble learning combines weak learning models into an arbitrarily stronger meta-model [26, p. 319]. The two common methods of creating ensembles are bagging and boosting. In bagging, m instances of training data will be bagged into m random sets with replacement [26, p. 319]. On the other hand, in boosting, classifiers which produce wrong results subsequently have a higher probability of being chosen as the next set of training examples [26, p. 320].

4.1.3. XGBoost

XGBoost is an open-source tree boosting machine learning system released in 2014 [27] inspired by the GBRT which was used in the solution that clinched the Netflix Prize in 2009. Since its inception, XGBoost has experienced explosive adoption, having been used by most winning solutions on Kaggle's data science competitions since 2015.

While the concept of GBRT is not new, XGBoost popularized it through its main advantage of scalability and memory efficiency through algorithmic optimizations such as a novel tree learning algorithm for handling sparse data, a weighted quantile sketch procedure which handles instance weights in approximate tree learning, and parallel computing to accelerate model exploration [27, p. 1].

4.1.4. LightGBM

LightGBM is a GBRT released in 2017 by Ke et al. [28] which is attuned for efficiency and scalability when in high feature dimensions and huge datasets. It is reported to quicken training by up to 20 times with similar accuracy vis-à-vis other conventional GBMs. Its authors assert that its innovations over XGBoost and other GBMs are that a small dataset can be used since data samples with larger gradients are used to compute information gain approximately, and that mutually exclusive features are combined into a single feature to reduce the dimensionality of data [28, p. 1].

4.1.5. CatBoost

Prokhorenkova et al. introduced CatBoost in 2019 to resolve the prediction shift problem caused by target leakage present in older GBMs [29]. It proposes permutation-based boosting over classical boosting and guesses categorical features efficiently by using an ordered version of "group(ing) categories by target statistics that estimate expected target value in each category" [29, p. 2].

4.1.6. CalixBoost

CalixBoost Ensemble is a novel Grid Search-based Weighted Average Ensemble proposed in this paper. It is a stacked ensemble of XGBoost, LightGBM and CatBoost. The program first trains the XGBoost, LightGBM and CatBoost models, stores their predictions in memory, then iterates through a pre-set list of weight combinations and calculates the weighted sum of the predictions to get the final ensemble prediction.

The best ensemble used the weights of 0.2:0.1:0.7 for XGBoost:LightGBM:CatBoost.

While Grid Search is a brute force method, the calculation of the ensemble prediction is computationally cheap, thus it is used.

4.1. Model Tuning: Bayesian Optimization

Bayesian Optimization is an automatic approach for maximizing the performance of machine learning algorithms. By generalizing an algorithm's performance using a Gaussian process, users can avoid trial-and-error tuning which is vastly inefficient. Bayesian Optimization can be on par with or exceed optimizations tuned by human experts and other algorithms like Latent Dirichlet Allocation (LDA) and Convolutional Neural Networks (CNNs) [30]. The library used in this paper for tuning hyperparameters (cf. brute-force methods such as Grid Search) is bayesian-optimization.

Bayesian Optimization uses a “Bayesian technique of setting a prior over the objective function and combining it with evidence to get a posterior function” [31]. A 1D Gaussian process estimation of the target function over nine observations is illustrated in Figure 5, along with its ~95% confidence interval shaded area (posterior uncertainty) representing the mean $\mu \pm$ the standard deviation σ . The solid line represents the target function, while the dotted line shows the prediction.

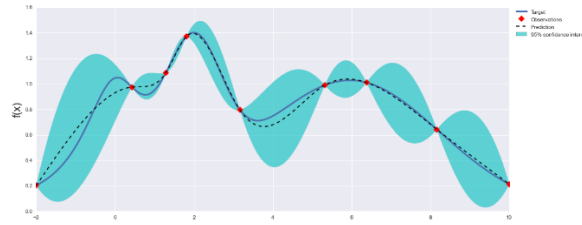


Figure 5. Gaussian Process after 9 Observations [32].

With more iterations, the *posterior uncertainty* decreases. Figure 6 shows the next best space to explore the point maximizing the utility function after said nine observations in Figure 5.

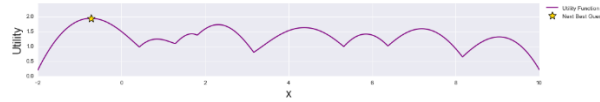


Figure 6. Utility Function after 9 Observations [32].

The algorithm strives to strike a balance between exploration and exploitation using the *Upper Confidence Bound* (UCB)-maximizing strategy used by the *Sequential Design for Optimization* (SDO) algorithm which suggests the best points for evaluation proposed by Cox and John [33]. The formula for UCB is given below:

$$UCB(x) = \mu(x) + \kappa \cdot \sigma(x) \quad (3)$$

κ is the kappa which is used to tune the parameterized acquisition model. In this project, $\kappa = 2.576$ is used which is the z-score when confidence level = 99%.

Figure 7 shows the end of Bayesian Optimization after ~80 observations and we observe that the algorithm has indeed balanced the exploration of the entire search space with the exploitation of higher target value areas (marked in dark red) in which more observations are sampled per unit search space.

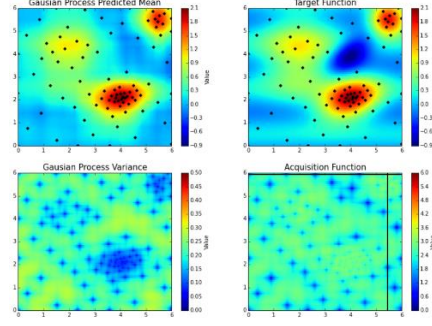


Figure 7. Gaussian Process Predicted Mean, Target Function, Gaussian Process Variance, and Acquisition Function after ~80 observations [32].

5. RESULTS AND COMPARISONS

5.1. Cross Validation

This paper proposes a novel holdout method which splits data into the following partitions:

- Training dataset (“train”): ~85%
- Validation dataset (“validation”): ~5%
- Test dataset #1 (“test”): ~5%
- Test dataset #2 (“blackswan”): ~5%

The term “black swan” stems from the black swan theory by Taleb in 2007. A black swan event is an outlier causing disastrous consequences.

Two test datasets allow for more rigorous evaluation among models: first, the “test” dataset comprises data points collected under normal economic circumstances in the U.S. economy ca. Dec 2019 to Mar 2020, and second, the “blackswan” dataset consists of data points collected under bear market conditions from Mar 2020 to May 2020 [34] during which the U.S. stock market crashed late Feb 2020 due to economic recession from both the COVID19 outbreak and a huge slump in oil prices due to glut.

With this special holdout method, we can evaluate whether each model can not only predict results accurately under ordinary financial situations, but also assess if it is robust enough to generalize well to extraordinary economic circumstances such as COVID-19.

5.2. Model Evaluation

This project evaluates all models based on Root Mean Squared Error (RMSE), Accuracy (Acc), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). Their formulae are given below where O_t and \hat{O}_{t+i} represent the actual and predicted price of Opening at time t .

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (O_{t+i} - \hat{O}_{t+i})^2} \quad (4)$$

$$Accuracy = \begin{cases} 1, & \text{if } |\hat{O}_{t+i} - O_t| = |O_{t+i} - O_t| \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$MAE = \frac{1}{N} \sum_{t=1}^N |O_{t+i} - \hat{O}_{t+i}| \quad (6)$$

$$MAPE = \frac{1}{N} \sum_{t=1}^N \frac{|O_{t+i} - \hat{O}_{t+i}|}{|O_{t+i}|} \quad (7)$$

For experimental repeatability, the libraries “numpy” and “random” are seeded with integers from 0 to 10, i.e., all results are aggregated over ten trials.

5.3. Experimental Results

The results of the “test” dataset (predicting under normal economic circumstances) are given in Table 6. The best performant *CalixBoost* model is compared against the best results of *RNN-GRU* [8] and *RNNBoost* [9] which were proposed in similar studies on stock market index prediction using social media analytics. The mean performance of *XGBoost*, *LightGBM*, *CatBoost* are additionally provided for comparison purposes. CalixBoost has the highest accuracy and lowest errors of all models.

Table 6. Test Dataset Results.

Model	Accuracy	RMSE	MAE	MAPE
RNN-GRU	-	0.8031	0.6254	9.38%
RNNBoost	66.54%	2.0500	1.3200	22.31%
XGBoost	79.19%	0.1052	0.0658	4.87%
LightGBM	80.23%	0.1382	0.0899	6.58%
CatBoost	82.67%	0.1992	0.1630	12.64%
CalixBoost	84.88%	0.0956	0.0573	4.19%

The results for the “blackswan” dataset where the stock market index is predicted in bear market conditions are provided in Table 7 for XGBoost, LightGBM, CatBoost and CalixBoost Ensemble. Models from other studies did not carry out such a test, thus there are no results to show for those. CalixBoost is the most performant; it generalizes well during black swan events such as economic recessions. In fact, CalixBoost’s performance for this dataset is still more performant than RNN-GRU and RNNBoost evaluated under normal economic conditions as seen in Table 6.

Table 7. Blackswan Dataset Results.

Model	Accuracy	RMSE	MAE
XGBoost	75.00%	0.2107	0.1709
LightGBM	73.26%	0.2523	0.2047
CatBoost	73.72%	0.4545	0.3861
CalixBoost	77.91%	0.2002	0.1617

While CalixBoost performs better than other GBMs and models in all areas, we contend that the improvement in performance of 2.2% maximum accuracy is significant but might not be substantial enough to justify training and assembling three GBMs into the CalixBoost ensemble. The authors thus recommend that unless achieving the best performance is of utmost necessity, any of GBMs can be used since they exhibit similar performance and it will be computationally considerably cheaper to train just one model.

ACKNOWLEDGEMENTS

The authors would like to thank Nanyang Technological University for the opportunity to conduct this research.

REFERENCES

- [1] Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990, June). Stock market prediction system with modular neural networks. In *1990 IJCNN international joint conference on neural networks* (pp. 1-6). IEEE.
- [2] Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2), 383-417.
- [3] Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014, March). Stock price prediction using the ARIMA model. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation* (pp. 106-112). IEEE.
- [4] Vui, C. S., Soon, G. K., On, C. K., Alfred, R., & Anthony, P. (2013, November). A review of stock market prediction with Artificial neural network (ANN). In *2013 IEEE international conference on control system, computing and engineering* (pp. 477-482). IEEE.
- [5] Yang, H., Chan, L., & King, I. (2002, August). Support vector machine regression for volatile stock market prediction. In *International Conference on Intelligent Data Engineering and Automated Learning* (pp. 391-396). Springer, Berlin, Heidelberg.
- [6] Chen, K., Zhou, Y., & Dai, F. (2015, October). A LSTM-based method for stock returns prediction: A case study of China stock market. In *2015 IEEE international conference on big data (big data)* (pp. 2823-2824). IEEE.
- [7] Gyamerah, S. A., Ngare, P., & Ikpe, D. (2019, May). On Stock Market Movement Prediction Via Stacking Ensemble Learning Method. In *2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)* (pp. 1-8). IEEE.
- [8] Chen, W., Zhang, Y., Yeo, C. K., Lau, C. T., & Lee, B. S. (2017). *Stock market prediction using neural network through news on online social networks. 2017 International Smart Cities Conference (ISC2)*. doi:10.1109/isc2.2017.8090834
- [9] Chen, W., Yeo, C. K., Lau, C. T., & Lee, B. S. (2018). *Leveraging social media news to predict stock index movement using RNN-boost. Data & Knowledge Engineering*. doi:10.1016/j.datak.2018.08.003
- [10] Zhou, Z. H. (2012). *Ensemble methods: foundations and algorithms*. CRC press.
- [11] Dey, S., Kumar, Y., Saha, S., & Basak, S. (2016). Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting. *PESIT, Bengaluru, India, Working Paper*.
- [12] Hutto, C. J., & Gilbert, E. (2014, May). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media*.
- [13] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (pp. 4765-4774).
- [14] Oliveira, N., Cortez, P., & Areal, N. (2017). The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications*, 73, 125-144.
- [15] Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of computational science*, 2(1), 1-8.
- [16] Mittal, A., & Goel, A. (2012). Stock prediction using twitter sentiment analysis. *Stanford University, CS229* (2011 <http://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf>), 15.
- [17] Narayan, P. K., Narayan, S., & Thuraisamy, K. S. (2014). Can institutions and macroeconomic factors predict stock returns in emerging markets?. *Emerging Markets Review*, 19, 77-95.
- [18] Estrella, A., & Mishkin, F. S. (1998). Predicting US recessions: Financial variables as leading indicators. *Review of Economics and Statistics*, 80(1), 45-61.
- [19] Chen, S. S. (2009). Predicting the bear stock market: Macroeconomic variables as leading indicators. *Journal of Banking & Finance*, 33(2), 211-223.
- [20] Pennebaker, J. W., Francis, M. E., & Booth, R. J. (2001). Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001), 2001.

- [21] Pennebaker, J. W., Boyd, R. L., Jordan, K., & Blackburn, K. (2015). *The development and psychometric properties of LIWC2015*.
- [22] Lim, B., & Zohren, S. (2020). Time series forecasting with deep learning: A survey. *arXiv preprint arXiv:2004.13408*.
- [23] Chen, T. (2015). *Is Normalization necessary?* · Issue #357 · dmlc/xgboost. Retrieved from <https://github.com/dmlc/xgboost/issues/357>.
- [24] Laurent, C., Pereyra, G., Brakel, P., Zhang, Y., & Bengio, Y. (2016, March). Batch normalized recurrent neural networks. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2657-2661). IEEE.
- [25] Mandrekas, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, 5(9), 1315-1316.
- [26] Poole, D. L., & Mackworth, A. K. (2010). *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press. (ISBN-13: 978-0-511-72946-1; ISBN-13 978-0-521-51900-7)
- [27] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).
- [28] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems* (pp. 3146-3154).
- [29] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. In *Advances in neural information processing systems* (pp. 6638-6648).
- [30] Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems* (pp. 2951-2959).
- [31] Brochu, E., Cora, V. M., & De Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.
- [32] Nogueira, F. (2014). *Bayesian Optimization: Open source constrained global optimization tool for Python*. Retrieved from <https://github.com/fmfn/BayesianOptimization>.
- [33] Cox, D. D., & John, S. (1992, October). A statistical method for global optimization. In *[Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1241-1246). IEEE.
- [34] Wearden, G., Jolly, J., Makortoff, K., Brignall, M., Ambrose, J., Kollwe, J., & Sweney, M. (2020, March 12). Wall Street and FTSE 100 plunge on worst day since 1987 – as it happened. Retrieved June 28, 2020, from <https://www.theguardian.com/business/live/2020/mar/12/stock-markets-tumble-trump-europe-travel-ban-ecb-christine-lagarde-business-live>

AUTHORS

Jarrett Yeo Shan Wei is a final year college student at Nanyang Technological University, Singapore.



Yeo Chai Kiat is an Assoc. Prof. at Nanyang Technological University, Singapore.



AN INTRODUCTORY REVIEW OF SPIKING NEURAL NETWORK AND ARTIFICIAL NEURAL NETWORK: FROM BIOLOGICAL INTELLIGENCE TO ARTIFICIAL INTELLIGENCE

Shengjie Zheng^{1,2}, Lang Qian³, Pingsheng Li⁴,
Chenggang He², Xiaoqi Qin⁵ and Xiaojian Li²

¹University of Chinese Academy of Sciences, Beijing, China

²Brain Cognition and Brain Disease Institute (BCBDI), Shenzhen-Hong Kong
Institute of Brain Science, Shenzhen Institute of Advanced Technology,
Chinese Academy of Sciences, Shenzhen, China

³Tsinghua Shenzhen International Graduate School,
Tsinghua University, Shenzhen, China

⁴McGill University, Montreal, Canada

⁵Beijing University of Posts and Telecommunications

ABSTRACT

Stemming from the rapid development of artificial intelligence, which has gained expansive success in pattern recognition, robotics, and bioinformatics, neuroscience is also gaining tremendous progress. A kind of spiking neural network with biological interpretability is gradually receiving wide attention, and this kind of neural network is also regarded as one of the directions toward general artificial intelligence. This review summarizes the basic properties of artificial neural networks as well as spiking neural networks. Our focus is on the biological background and theoretical basis of spiking neurons, different neuronal models, and the connectivity of neural circuits. We also review the mainstream neural network learning mechanisms and network architectures. This review hopes to attract different researchers and advance the development of brain intelligence and artificial intelligence.

KEYWORDS

Spiking Neural Networks, Brain-Inspired Intelligence, Deep Neural Networks, Artificial Intelligence and Biological Intelligence.

1. INTRODUCTION

Spiking neural networks (SNN) based on brain-inspired computation, a model that mimics the brain's intelligent computational mechanism, are considered as one of the paths to achieve general artificial intelligence, and this class of algorithms is gaining widespread attention. Meanwhile, traditional deep neural networks (DNN) have shown extraordinary capabilities in several tasks and seem to have become omnipotent. But there are some key questions: What are the intelligent performances of these two different types of networks? What are the similarities between them?

Inspired by the brain hierarchy and the integration of neural information, artificial neural networks use a multilayer network architecture to transform input information into features, but this precise transformation of data integration is incompatible with the way the brain processes information. Therefore, we discuss here the similarities and differences between biological and artificial intelligence, from the basic information units, network architecture, and learning mechanisms, and how these two types of neural networks represent information as two different ways of processing information.

We first discuss the "biological" aspects, including how neurons integrate information and how information is transmitted between neurons. Then, we explore the biological neuron model and the artificial neuron model, and how these two different types of neurons process information. After that, we discuss how biological neuronal recurrent connected, how different circuits can achieve different functions, and explore the role and embodiment of different neural circuits in biology. Then, we discuss three mainstream machine learning algorithm paradigms and carry out the related biological interpretability discussion, and discuss the advantages and disadvantages of SNN and DNN using different machine learning algorithms. Next, we proceed to discuss the main network architectures of ANN and SNN, discuss that SNN is bio-interpretable, and discuss the importance of network architecture. Finally, we discuss the future of SNN and ANN as different types of networks and how they complement each other, including the generality of SNN and the accuracy of ANN, to achieve a step towards general artificial intelligence.

2. BIOLOGICAL BACKGROUND

2.1. The Neuron and Synapse

The Neuron. The human brain is composed of 86 billion neurons and is the most complex organ within the human body[1]. The highly structured connections between neurons and their interactions form efficient information communication, resulting in neural networks. A classical neuronal structure is composed of three parts: dendrites, soma, and axon. The vast majority of neurons are polarized cells, and cell polarity refers to the spatial variation in shape, structure, and function within a cell. Almost all cell types exhibit some form of polarity, which allows them to perform specialized functions. Dendrites in nerve cells are structured in a dendritic distribution and transmit the received input signals to the soma. With the input of information, the soma changes its own membrane potential in response to all inputs from the dendrites, and when the membrane potential reaches a certain threshold, an action potential is generated. The action potential is transmitted along the axon as an output, and the spike signal is transmitted to the axon terminal (**Figure 1.a**).

The Synapse. Neurons form connections with each other and transmit information through synapses. The neurons before and after the synapse are the presynaptic and postsynaptic neurons, respectively. For chemical synapses, presynaptic neurons are not directly connected to postsynaptic neurons, but rather to a gap called the synaptic gap. When the action potential travels along the axon to the presynaptic terminal, the presynaptic terminal will produce neurotransmitters, molecules that are packaged into structures called vesicles, and the action potential causes these vesicles to fuse to the membrane and finally be released back into the synaptic gap and bind to receptors on the surface of the postsynaptic terminal, and the binding of different receptors to specific neurotransmitters will affect the changes in the postsynaptic neuron (**Figure 1.b**).

2.2. Action Potential

The membrane potential is the difference in electrical potential between the inner and outer membranes of the neuron, and the key to the generation of action potentials lies with substances inside and outside the cell membrane, generally charged ions and molecules, and we focus here only on charged ions, such as sodium, potassium, calcium, and chloride ions. The cell membrane itself is a good insulator with high electrical resistance but is itself filled with many ion channels that allow ions to flow through the various ion channels. In the cell membrane, there is a very important class of channels called voltage-gated channels, which open or close at any given moment depending on the local membrane potential shift (**Figure 1.c**). The membrane potential is initially resting potential, which is the membrane potential of a neuron in the absence of any stimulus. When the neuron is stimulated, the voltage-gated sodium channels will open when the membrane potential reaches a certain stage, causing the membrane potential to rise rapidly, a process called depolarization, until a threshold is reached when the membrane potential rises due to the inactivation of sodium channels and the opening of potassium channels. This process is called depolarization until the threshold is reached, and the membrane potential decreases rapidly due to the inactivation of sodium channels and the opening of potassium channels, which is called hyperpolarization or repolarization, and the membrane potential gradually returns to the resting potential with the closing of potassium channels.

2.3. Synaptic potential and synaptic integration

Synaptic transmission has two basic forms: excitation and inhibition, and these two form carriers are excitatory neurons and inhibitory neurons, respectively (**Figure 1.d**). The signal from an excitatory neuron causes the membrane potential of the postsynaptic neuron to toward a more positive or depolarized value, which prompts the downstream neuron to fire, hence the term excitatory postsynaptic potential (EPSP), for the signal generated by this postsynaptic neuron. Inhibitory neurons cause the membrane potential of the postsynaptic neuron to toward a more negative value. Unlike EPSP, this signal inhibits the downstream neuron in such a way that the membrane potential is farther from the threshold, hence the term inhibitory postsynaptic potential (IPSP). The neuron continuously receives excitatory and inhibitory inputs, and when this input sum reaches or exceeds the threshold, it excites an action potential, otherwise, it remains silent. This process of receiving synaptic inputs is called synaptic integration.

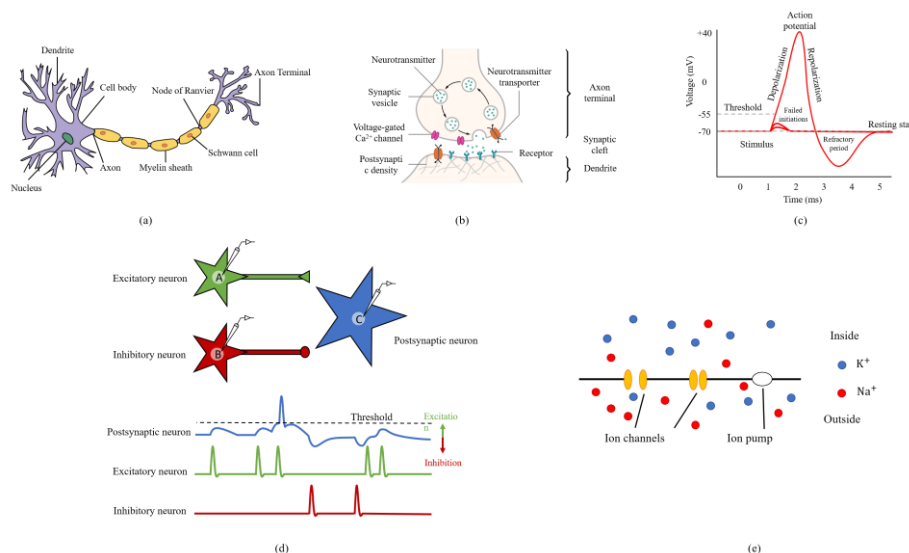


Figure. 1. (a) The Neuron, (b) The Synapse, (c) The Action Potential, (4) Synaptic Integration, (5) A brief diagram of the membrane environment.

3. NEURON MODELS

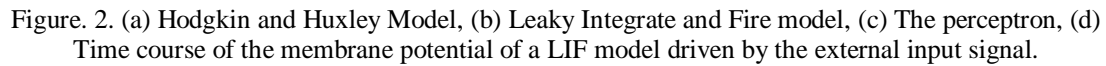
A neuronal model is a mathematical description of certain properties in the nervous system. Biological neurons are known as action potentials because they generate spike signals that last approximately one millisecond in duration. These spike signals carry extremely strong temporal-spatial properties and contain a large amount of information; therefore, the spiking neurons that generate action potentials are considered as information processing units of the nervous system. There are various types of pulse neuron models, such as the Hodgkin and Huxley model, which is based on ion channels and thus simulates membrane potential changes, or the Leaky integrate-and-fire model (LIF) model, which is based directly on stimulus-induced membrane potential changes. A brief description of the basis of the models will be given below.

3.1. Hodgkin-Huxley model

Hodgkin and Huxley performed experimental recordings on a giant axon of a squid in which they injected electrical currents directly into the axon. A series of careful measurements led to a biophysical model description, an equivalent analog circuit, and a mathematical model (**Figure 1.e**) [2]. the Hodgkin-Huxley model (Equation 1) can be represented by the following differential equation.

$$\begin{aligned}
 C \frac{du}{dt} &= -g_{Na}m^3h(u - E_{Na}) - g_Kn^4(u - E_K) - g_l(u - E_l) + I(t) \\
 \frac{dm}{dt} &= \alpha_m(u)(1 - m) - \beta_m(u)m \\
 \frac{dn}{dt} &= \alpha_n(u)(1 - n) - \beta_n(u)n \\
 \frac{dh}{dt} &= \alpha_h(u)(1 - h) - \beta_h(u)h
 \end{aligned} \tag{1}$$

This equivalent simulated circuit corresponds to specific resistances for sodium, potassium, and leak channels, and these resistances change differently over time, with the change in resistance of the simulated circuit corresponding to the opening and closing of the ion channels. Where $I(t)$ represents the input current, u represents the cell membrane potential, E_{Na} , E_K , and E_l correspond to the reversal potential of sodium, potassium, and leaky channels, $g_{Na}m^3h$, g_Kn^4 , and g_l correspond to the conductivity of different channels. m , n , and h are assumed to be the concentrations of certain ion-transport-related particles, and their corresponding α and β symbolize the rates of movement of the particles into and out of the membrane (**Figure 2.a**). Because of the description of neuronal dynamics at the level of ion channels, this result laid the biophysical foundation of neuroscience, for which Hodgkin and Huxley were awarded the Nobel Prize in 1963.



The Leaky integrate-and-fire model (LIF), a simplified version of the neuron model, is widely used as the basic unit of spiking neural networks due to its relatively small computational complexity (**Figure 2.b**). The basic concept of the LIF neuron was proposed by L.E Lapicque in 1907[3].

In the LIF model (Equation 2), τ is the time constant of the differential equation and u_{rest} is a constant parameter represented as the resting potential of the cell membrane. $I(t)$ is the input current and R is the membrane resistance (**Figure 2.d**).

The Izhikevich neuron model (Equation 3 and 4) is a two-dimensional system of ordinary differential equations[4], as shown below.

$$\frac{du}{dt} = a(bv - u) \quad (4)$$

where u is a membrane recovery variable used to describe the ion current behavior in general, and a , b are used to adjust the timescale of u and the sensitivity about the membrane potential v , respectively. By the choice of parameters, the Izhikevich model can demonstrate the firing patterns of almost all known neurons in the cerebral cortex with much less computational overhead than the Hodgkin-Huxley model.

3.4. Perceptron

The perceptron [5], as the most basic computational unit in an artificial neural network, perceives a weighted summation of the inputs and then produces output through an activation function, which is essentially a simulation and simplification of a biological neuron. Biological and artificial neurons have many similarities, but biological neurons need to consider the temporal dimension of information as well as the morphological spatial dimension of the neuron itself, while the input and output have a highly nonlinear relationship (**Figure 2.c**).

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$\mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^n w_i x_i = w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots + w_n x_n \quad (6)$$

The perceptron (Equation 5 and 6) is defined as a binary classifier, which is actually a mapping function for the input \mathbf{x} . \mathbf{w} is the weight value, $\mathbf{w} \cdot \mathbf{x}$ is the dot product, b is the bias, and $\mathbf{w} \cdot \mathbf{x} + b$ is based on the mapping of the binary step function $f(\cdot)$, which yields an output value denoted as "1" or "0", the output value can also be other numbers other number, depending on the chosen activation function $f(\cdot)$.

3.5. Relation between artificial and spiking neuron Model

Both neuron models transform input information into output. However, in fact, this is still a different mechanism from that of biological neurons to process information. Biological neurons need to consider the time-dependence about the input and output information which is also the spike information. But it is also the artificial neuron perceptron that chooses a different way of processing information than the biological neuron, together with a specific activation function, which allows the artificial neuron to compute as well as tune the network parameters efficiently in the network model.

4. NEURONAL CIRCUIT

Neural circuits are neuron populations interconnected by synapses that perform specific functions when the circuits are activated. The specific way in which these synapses are connected provides the physical basis for neural population dynamics, and these circuits are also used in the architectural design of spiking neural networks. A brief description of neural circuits will be given below.

Circuit motifs. Neurons are individual cell units in the nervous system that not only receive input signals from dendrites and process them within the cell body but also send output signals to presynaptic terminals via axons, which neuroanatomist Ramón y Cajal calls "the neuron doctrine"[6].

Neurons do not exist independently within the brain but are highly interconnected in synapses to form circuits that work together to process information, and this connectivity pattern provides the basis for the neuron population to perform specific functions. For example, specific circuits are associated with short-term memory and long-term memory storage, the extent of feature sensory fields, etc. This neuron population, processing information in a similar way to individual neurons, integrate incoming information and then decides whether to perform the output of the information. Also, circuits are modulated by the type of synaptic input they receive, such as excitability as well as inhibition.

Feedforward excitation with Convergence and Divergence. Feedforward excitation allows a neuron to propagate excitatory signals from itself downstream, and a series of feedforward excitatory connections is common in the nervous system, which allows signals to propagate throughout the system internally, by way of Convergence, where a single postsynaptic neuron receives excitatory signals from multiple presynaptic neurons, and by way of divergence, where a single presynaptic neuron signaling with multiple postsynaptic neurons (**Figure 3.a**). Convergent excitation can enable postsynaptic neurons to respond selectively to features not solely or explicitly present in any of the presynaptic neurons. It can also increase the signal-to-noise ratio if multiple input neurons carry the same signal but uncorrelated noise[7].

Feedback/Recurrent excitation. For Feedback excitation, presynaptic neuron A makes an excitatory connection to postsynaptic neuron B, which in turn connects back to presynaptic neuron A (**Figure 3.b**). Moreover, there are also axons of neurons that connect themselves as a recurrent connection.

Mutual excitation. Neuronal cells B, C, D, and E, make excitatory-type interconnections, while neuron B also makes cyclic connections to itself; this excitatory-type connection will allow feedback of excitatory information from the neurons in the network so that the neurons in the network can make prolonged excitatory states corresponding to brief stimuli (**Figure 3.c**). This type of connectivity has been used in computational models of working memory[8], as well as short-term memory encoding, which plays an important role.

Feedforward inhibition. When inhibitory neurons are between excitatory neurons, feedforward inhibition is a form of signaling in neuronal transmission, and when presynaptic cells excite the interneuron as inhibitory, the signal from this inhibitory neuron will inhibit the activity of downstream cells (**Figure 3.d**).

Feedback/Recurrent inhibition. For feedback/Recurrent inhibition, the presynaptic cell is connected to the postsynaptic cell, the postsynaptic cell is connected to the interneuron as an inhibitory effect, and the interneuron is then connected to the presynaptic cell (**Figure 3.e**). This circular connection serves as a feedback inhibitory effect, which inhibits the activity of excitatory neurons.

Lateral inhibition. Presynaptic cells excite inhibitory interneurons, and thus they inhibit adjacent cells of excitatory neurons (**Figure 3.f**). For example, during information processing in the visual system, limbic enhancement is achieved by lateral inhibition of the retina[9, 10].

Mutual inhibition. Two inhibitory neurons interconnect, when neuron A directly inhibits neuron B, and neuron B receives the inhibitory signal and in turn inhibits neuron A (**Figure 3.g**). These mutually inhibitory circuits play a key role in designing the Central Pattern Generator (CPG) computational neural model [11] and in regulating circadian rhythms in the brain [12].

The interconnection of many neurons in a neural system results in changes in both structure and function as the size increases, and such changes have the emergent properties of physical systems, which arise from the interaction of different elements in the system. Currently, artificial neural networks and spiking neural networks are also trying to make neural networks with emergent properties, or intelligent emergent properties, through different network architectures and neural circuit designs.

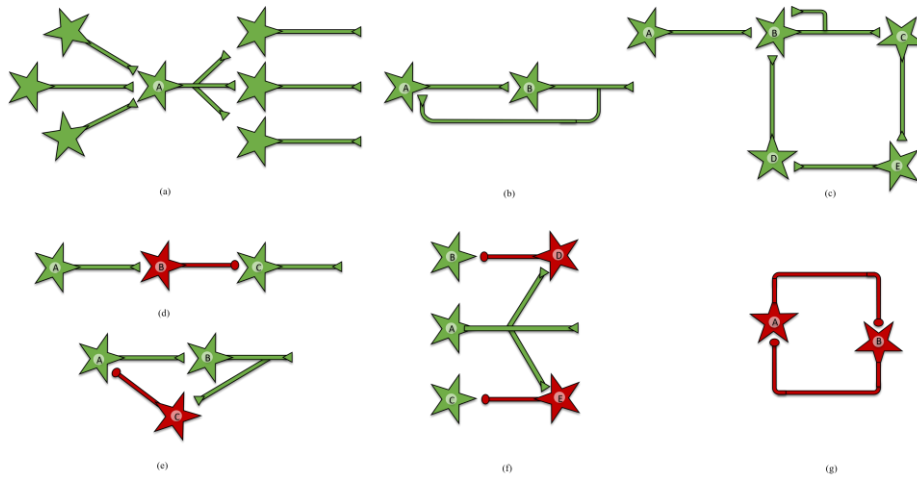


Figure. 3. (a) Feedforward excitation, (b) Feedback/Recurrent excitation, (c) Mutual excitation, (d) Feedforward inhibition, (e) Feedback/Recurrent inhibition, (f) Lateral inhibition, (g) Mutual inhibition.

5. LEARNING TYPES

In machine learning for artificial intelligence, there are three main types of machine learning, namely supervised learning, unsupervised learning, and reinforcement learning, which are also used in artificial neural networks as well as spiking neural networks. These learning methods allow neural networks to learn based on data and thus tune the internal parameters of the network to achieve the ability to solve tasks.

5.1. Supervised learning

In general, supervised learning deals with labeled data, which is a process of finding a mapping from the input to the output space. The tasks of learning can be divided into two categories, classification, and regression. The tasks need to produce the desired output for each input. This learning mechanism has the advantage of being able to obtain patterns in the data based on prior knowledge, but only if high-quality data inputs, as well as corresponding outputs, are required. Therefore, it is one of the types of learning that can achieve efficient learning tasks and obtain the best solutions.

Supervised learning in ANN. Artificial neural networks are interconnected through multiple layers of nodes, and such neural networks can be learned in a supervised method so as to find a mapping through a loss function, sometimes called an error function, which indicates a function of the distance between the current output and the expected output[13]. The network parameters are adjusted according to the loss function by means of gradient descent. The model gives the best answer when the loss function is close to zero.

Gradient descent and backpropagation often occur together, and backpropagation is an algorithm for training ANNs for supervised learning. It efficiently calculates the gradient of the error function with respect to the network weights of a single input-output example. Thus, the weights are updated to minimize the loss function. In 1986, Seppo Linnainmaa proposed automatic differentiation. It was later used in experiments for learning internal representations by D.E Rumelhart et al[14]. The successful application of backpropagation has caused a renaissance in the field of research on artificial neural networks. Today, thanks to powerful GPU computing systems, BP algorithms show great advantages in the training of different networks.

Supervised learning in SNN. In traditional artificial neural networks, labels can be represented as integers (classification) or real numbers (regression). In spiking neural networks, the labels are encoded as spike trains with spatial-temporal properties. However, unlike artificial neural networks, it is difficult and unwise to directly apply gradient descent methods to SNNs due to the discontinuity of spike signals.

1. **Spike Response Model.** The first gradient descent-based supervised learning algorithm for SNNs was proposed by Bohte et al[15]. The method implements a gradient descent-based multilayer spiking neural network error backpropagation method using the temporal encoding of spike intervals.
2. **Spike Pattern Association Neuron (SPAN).** Another very powerful supervised algorithm is SPAN[16], whose core idea is to obtain an analog signal by convolving our chosen kernel function with the spike signal, where the Widrow-Hoff rule (Delta rule) can be directly applied to the transformed signal to adjust the weights in the network
3. **Surrogate Gradient Descent.** Recently, the Surrogate gradient descent was proposed by Zenke et al[17]. It introduces continuous relaxation of gradient estimation without affecting the forward transmission of spike sequences to achieve very good results.

Supervised learning requires the annotation of data, a process that requires a lot of human resources, and data annotation, which is the basis of most artificial intelligence, determines the quality of learning models. However, this differs from biological intelligence in that human beings often do not have a precise annotation during the learning process, and that the parameters of the entire network are updated based on backpropagation in a way that does not exist within biological structures. Thus, supervised learning is different from biological learning.

5.2. Unsupervised learning

Unsupervised learning is an algorithm for learning patterns from unlabeled data, which is important in the biological learning process. At the same time, this learning approach hopes to solve the problem of supervised learning requiring data labeling, because, in real life, it is difficult for labelers to perform manual labeling if they lack sufficient a priori knowledge. The development of unsupervised learning is solving these problems and has even rivaled supervised learning in uncovering hidden structures in the data.

Unsupervised learning in ANN.

Unsupervised learning based on artificial neural networks has gradually become known as a research hotspot in recent years, and has even been called the next stop for AI. Yann LeCun once used a cake analogy, "If intelligence is a piece of cake, then most of the cake is unsupervised learning, the frosting on the cake is supervised learning, and the cherry on the cake is reinforcement learning". Indeed, unsupervised learning with great potential is already producing significant results in several fields.

1. **Autoencoder [18].** It is an unsupervised neural network model in artificial neural networks that learns the implicit features of the input data, which is called encoding while reconstructing the original input data with the learned new features, called decoding. Autoencoder can be used for feature extraction as well as dimensionality reduction.
2. **Generative adversarial network [19].** Generative adversarial networks learn by letting two networks contest each other, consisting of a generative network as well as a discriminative network, respectively. The generative network takes random samples from the latent space as input, and the output needs to be similar to the real samples of the training set. The pur-

pose of the discriminative network is to be able to discriminate the output of the generative network.

3. **Self-organizing map [20].** A self-organizing mapping neural network, which uses unsupervised learning to produce a low-dimensional representation of the input space of discretized training samples, is called mapping and is, therefore, a method of dimensionality reduction. Self-organizing mapping differs from other artificial neural networks because it uses competitive learning in which output neurons compete with each other for activation, with only one neuron being activated at any given time, called winner-takes-all neuron.

Unsupervised learning in SNN.

In the process of biological learning, we know that even when we are in infancy, we are able to achieve recognition as well as comparison of things that are not labeled. This seems to indicate that biological neural networks are capable of unsupervised learning of external input information to build an initial understanding of the outside world. The following will describe the use of spiking neurons to model neural networks using biologically based explanatory learning rules to build the basis for biological learning and memory.

1. **Hebbian learning.** During learning and memory, the weight of synaptic connections between biological neurons is strengthened or weakened in response to the activity between neurons, which is crucial for information storage in the brain. Donald Hebb assumes that the persistence or repetition of a reverberatory activity tends to induce lasting cellular changes that add to its stability[21]. This theory is also known as Hebb's rule, Hebb's postulate, and cell assembly theory. Hebb's rule is also summarized as "fire together wire together".
2. **Spike-Timing Dependent Plasticity (STDP).** STDP refers to the observation that the precise spike timing influences the enhancement and inhibition of synaptic plasticity [22]. For example, in the connections between mammalian pyramidal neurons, when presynaptic spikes occur within a certain time window of postsynaptic spikes, it will lead to long-time-travel potentiation (LTP); if the order is reversed, it will lead to long-time-travel depression (LTD), a phenomenon that occurs in the synapses of biological neurons, although it is not yet applicable to all brain regions and cell types.

$$\Delta w = \begin{cases} a^+ e^{\frac{-(t_{pre}-t_{post})}{\tau}} & t_{pre} - t_{post} \leq 0, a^+ > 0 \\ a^- e^{\frac{-(t_{pre}-t_{post})}{\tau}} & t_{pre} - t_{post} > 0, a^- < 0 \end{cases} \quad (7)$$

Δw represents the amount of synaptic weight change (Equation 7), a^+ and a^- are the learning rate parameters, respectively, τ is the time constant, and $t_{pre} - t_{post}$ represents the time difference between presynaptic neuron spike delivery and postsynaptic neuron spike delivery, respectively (**Figure 4.a**).

3. **Triplets STPD.** In 2006, Triplet was proposed by Pfister and Gerstner[23], in which LTP is constructed as a combination of one presynaptic as well as two postsynaptic spikes, and LTD is based on the combination of two presynaptic as well as one postsynaptic spike instead of a pair of spikes based on STDP rule, this rule can well take into account the spike-timing interactions, but still also does not apply to the interpretation of all cell types (**Figure 4.b**).

For now, biologically interpretable learning rules for unsupervised learning are still to be explored, especially for SNNs, based on the fact that unsupervised learning still has a large gap for complex tasks compared with supervised learning, and the main reason in this regard is still

based on the fact that current unsupervised learning rules are not perfect enough to enable spiking neural networks to learn from data effectively. More explanatory unsupervised learning algorithms will be the key to achieve SNN performance improvement.

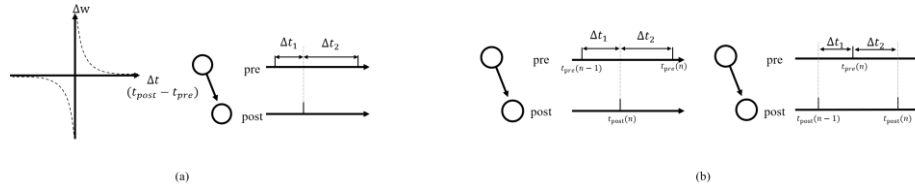


Figure. 4. (a) Spike-Timing Dependent Plasticity, (b) The triplets STDP.

5.3. Reinforcement learning

Reinforcement learning (RL) is a machine learning approach to artificial intelligence that works to create computer programs capable of solving problems that require intelligence. This learning algorithm is inspired by the study of reward mechanisms for animal learning.

Reinforcement learning in ANN.

The unique feature of RL based on artificial neural networks is that it learns from feedback through iterative trials that are simultaneously sequential and evaluable through the use of powerful nonlinear function approximations. As a framework for solving control tasks, it learns from the environment by constructing agents that interact with the environment through iterative attempts and receive rewards as the only feedback.

1. **Value-based.** Learn the state or state-action value. Act by choosing the best action to take in this state. Q-learning is one of the most classic value-based algorithms [24], the Deep Q-Network algorithm, which was proposed by DeepMind in 2015 [25], enables algorithms to play Atari games like humans by combining Q-Learning in reinforcement learning and deep neural networks. They accept several frames of the game as an input and take the output state value of each action as output.
2. **Policy-based.** As REINFORCE Gradient from Williams[26], the policy is learned as a mapping from the state space to the action space, telling the agent the best action to take in each state to maximize its return.

Reinforcement learning in SNN.

In the process of biological learning, many learning processes are accompanied by reward mechanisms. This appears to be a global reinforcement signal acting on multiple regions of the brain, resulting in changes in the connectivity of the neural network. And spiking neural networks have great potential to mimic this way of biological learning, adjusting their own network connections to get the most out of the reward signal.

1. **Three-factor Learning Rules.** This approach works by setting a flag, called an eligibility trace[27], on the synapse upon co-activation of presynaptic and postsynaptic neurons. synaptic weights change only when a third factor, indicating reward, is present when the flag is set.
2. **ANN to SNN.** By matching the firing frequency of the firing neurons and the graded activation of the analog neurons, the trained artificial neural networks can be converted to the corresponding SNNs[28].

Reinforcement learning is extremely biologically interpretable, a theory inspired by the psychology of how agents gradually develop expectations of stimuli in response to rewarding or punishing stimuli given by the environment, producing habitual behaviors that yield the greatest benefit. There is a prevailing view that dopamine neurons enable this function, comparing future expectations with previous mental benchmarks and thus releasing neurotransmitters depending on the result, thus making the creature happy or frustrated, using a reward mechanism as the basis for learning [29]. However, current reinforcement learning using deep learning in practice generally requires a large amount of training time, and how to truly learn based on reward and efficiently like a living creature is something that currently needs to be improved.

6. ARCHITECTURES OF NEURAL NETWORK

Artificial neural networks are inspired by the brain, but compared to the brain, there are fundamental differences in both neuronal topology, neural information processing, and neural learning mechanisms. Within the biological context, neurons communicate with each other by passing spikes, and the information is embedded in the spike trains. ANNs use continuous value as the information transmitted between artificial neurons, and the model architecture is of the human-designed type, which generally has no biological functional properties. With the rapid development of deep artificial neural networks and neuromorphic hardware, these advances have simultaneously led to new research and hypotheses on spiking neural networks.

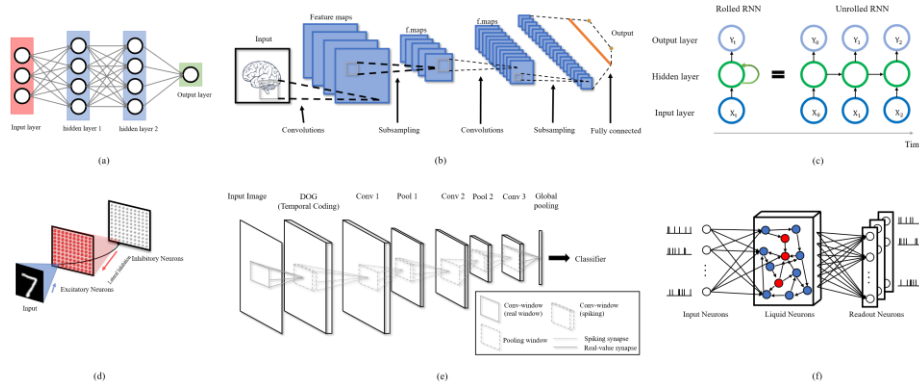


Figure 5. (a) Feedforward Neural Network, (b) Convolutional Neural Network, (c) Recurrent Neural Network, (d) Spiking Feedforward Neural Network with Lateral Inhibition, (e) Spiking Convolutional Neural Network, (f) Liquid state machines.

6.1. Feedforward Neural Network

Artificial Feedforward Neural Network. Feedforward Neural Network, also known as Multi-layer perceptron (MLP) [30] (Equation 8), is essentially a nonlinear composite function $N(\cdot)$ that approximates a certain function $y(\cdot)$, which maps the input x to the output $y(x)$ (**Figure 5.a**). The input x is able to obtain the corresponding output through a series of nonlinear transformations $f^{(k)}(\cdot)$, i.e., the nonlinear activation function $\sigma(\cdot)$. The input is passed forward after the nonlinear transformation in each layer, and finally the output is obtained. Usually, the first layer of the network is called the input layer, the last layer is called the output layer, and the one between the first and the last layer is called the hidden layer, and each unit in the network is also known as the perceptron. As the number of hidden layers of the network increases and the number of units within the hidden layers increases, the complexity of the FNN nonlinear model $N(x)$ increases, so it can approximate any function that can satisfy any nonlinear mapping relationship between the input x and the output $y(x)$.

$$y(x) \approx N(x) = \sigma(\mathbf{w}_k \sigma(\mathbf{w}_2 \sigma(\mathbf{w}_1 \cdot \mathbf{x} + b_1) + b_2) + b_3) = f^{(3)} \left(f^{(2)} \left(f^{(1)}(\mathbf{x}) \right) \right) \quad (8)$$

Spiking Feedforward Neural Network. There are many SNNs based on STDP learning or BP-based supervised learning that have achieved success in different types of pattern recognition, and even some of the STDP-based networks are comparable to BP-based supervised learning, such as Diehl et al[31], who showed that using a two-layer SNN, based on the biological properties of excitatory type neurons and inhibitory neurons as the processing layer, using lateral inhibition as well as winner-take-all properties, enabling the neurons in the processing layer to extract features with significant characteristics from the input signal based on STDP learning rules, with optimal performance of 95% on the MNIST dataset (**Figure 5.d**).

6.2. Convolutional Neural Network

Artificial Convolutional Neural Network. CNN is usually used to process gridded data (e.g., images) and consists of layers that process visual information, the most commonly used layers being convolutional, pooling, and fully connected layers[32]. CNN learns the spatial patterns in an image region by looking at groups of pixels in it. the convolutional layer looks for spatial features from the input to perform feature extraction, and this operation is performed through a series of filters, also known as convolutional kernels, convolves the input, which is essentially a cross-correlation operation, followed by a nonlinear activation function, and multiple filters to obtain multiple corresponding outputs. The convolution layer is followed by the pooling layer, which reduces the size of the feature space to reduce the number of parameters in the network and the amount of computation, while it helps to extract dominant features of translation invariance, thus making the model training effective. The final layer is the fully-connected layer, where the initial input is transformed into highly abstract and low-dimensional information representing the input through a series of convolutional and pooling layers. The fully connected layer transforms the output of the feature extraction layer into a vector and connects the feedforward neural network FNN, and the last layer classifies the output by a softmax function (**Figure 5.b**).

The convolutional layers as well as the pooling layers in CNN come from the concept of simple and complex cells in neuroscience[33]. The overall architecture of CNN has some similarities with the visual ventral pathway LGN-V1-V2-V4-IT, for example, the layers near the input may represent the contour information of a picture, while the layers near the output can be more representative of the category.

Spiking Convolutional Neural Network. There are convolutional kernels that use V1-like properties for the CNN closest to the input level, which can extract salient features for images. For example, SR Kheradpisheh et al. used a Difference-of-Gaussian kernel[34] for the input image, followed by unsupervised STDP-based training of the convolutional layer as well as the pooling layer, and finally, the extracted features are passed into the classifier[35](**Figure 5.e**).The performance of directly trained SNNs is often inferior to that of traditional DNNs, while training on non-neuromorphic hardware is time-consuming, and ANN conversion to SNNs can solve this problem. Many studies have shown that converted SCNNs work well and perform close to CNN and that SCNNs can perform inference tasks on neuromorphic hardware[36] and consume less energy.

6.3. Recurrent Neural Network

Artificial Recurrent Neural Network.RNN is a class of neural networks used to process sequential data or time-series data, often used to solve ordinal or temporal problems, such as language translation, speech recognition[37], etc. RNNs, like FNNs and CNNs, are also composed

of the difference that lies in the learning process. Instead of memorizing the overall sequence information, the RNN uses the representational information in the hidden layer to memorize the information in the most recent time step in the learning process based on time series, and the RNN combines the representational information of the previous time step in the hidden layer with the input of the current step to infer the output of the current time step (**Figure 5.c**). Currently, RNN-based Gated recurrent unit (GRU)[38] as well as Long short-term memory (LSTM)[39] have been used to powerful effect in real-life applications.

Spiking Recurrent Neural Network. Neural circuits in the brain display the remarkable dynamic richness and high variability in the form of recurrent connections. Excitatory and inhibitory neurons interconnect to form a neural network that is in a chaotic as well as an equilibrium state transition. This recurrent neural network has complex nonlinear dynamics and can be used to study biological neural networks in specific microcircuits of the brain. People often use Liquid state machines (LSM) for computational modeling, the essence of LSM is related to its own naming, the idea is to throw a stone into a lake, the lake indicates that ripples will be generated, based on the current activity of the lake, it is possible to evaluate what happened previously in the system, such as how long ago the stone entered the lake and thus caused the ripples[40]. Essentially, the lake is the LSM, the rocks are the input, and the ripples are the cluster response of the LSM. the LSM usually consists of three layers, the input layer, the reservoir or liquid layer, and the memoryless readout layer (**Figure 5.f**). this recurrent neural network transforms the time-varying input information into a higher dimensional space by it can exhibit rich temporal as well as spatial properties of neuronal dynamics, and thus can memorize past input information.

The emergence of these different network architectures actually comes from the understanding of the neural system. How to implement different tasks in a generic architecture cannot yet be done on ANNs as well as SNNs. Nevertheless, in many ways, SNNs and ANNs can be complementary and do not replace each other, and for brain science and brain-inspired research, SNNs are of great importance. Faced with computationally oriented tasks, ANNs have unparalleled advantages. Currently, a class of network models combining SNN and ANN has been born that can exploit the advantages of each. With the understanding of biological neural networks, it should be possible to inspire new neural network architectures, and through the conditioning of large-scale neural information as well as recording, it is possible to further understand the patterns of neural information within the network, providing a basis for neural network architectures with biological explanations.

7. CONCLUSION

In this paper, we present the biological background of neurons and then describe mathematical models of biological neurons that simulate the changes in membrane potential of neurons with different computational complexity. An overview of neural circuits in biology is also given, with different circuits implementing different information processing functions. This is followed by an application of the mainstream learning mechanisms in different neural networks. Finally, a review of the mainstream network architectures is presented.

The review shows that the way neurons are connected and the learning mechanisms are the basis of brain learning, and that for different network architectures and different learning mechanisms, neural networks will exhibit different information processing. For SNNs, the use of a bio-interpretive STDP learning mechanism and a recurrent connection network structure has excellent advantages for processing spatial-temporal information. For ANNs, this artificially designed structure with set learning mechanisms and high-quality big data can lead to models that explain the hidden structural patterns of the data.

However, current neural networks face some potential challenges, at least for both SNNs and ANNs. For example, it is unclear whether neural circuits simulated using SNNs can explain biological neural circuits, and the use of different parameters is likely to yield different results. Also, although ANNs can work well in principle and have been used very successfully in engineering, in order to achieve specific functions, ANNs need to receive specific data as well as learning method constraints, which are often different from biological processes. Although, more and more ANN models have recently applied new learning paradigms, ANNs currently fail to achieve better results for multimodal tasks, as they are still inherently data-driven. How to learn effectively with small-scale data will be the key to the progress of ANN or SNN, because they can effectively reveal the nature of learning directly based on the inherent prior knowledge of the network and can evaluate the effectiveness of data learning.

Perhaps, in the future, neural microcircuits in the brain and large-scale neuronal cluster acquisition and analysis can provide new tools that will play a crucial role in discovering neural network architectures as well as learning mechanisms. There may also be a need to integrate advances in different fields, such as neuromorphic chips and hybrid chips, which will facilitate the development of AI as well as brain-inspired intelligence at different levels.

Finally, the neural networks based on general AI and brain-inspired intelligence are far from complete. For example, what computations are done by the dendrites of individual neurons on the inputs[41], what is the internal information flow process of the neural circuits of biological neural networks, and the uninterpretability of back propagation algorithms in biological neural networks. Whether ANN can be combined with current biological laws to achieve similar effects of biological neural networks.

If the neural network gets further breakthrough, it may be possible to expose the nature of neural network for information encoding from another perspective and can effectively solve some problems such as learning and memory, motor planning, pattern recognition, etc. In a new theoretical architecture, either ANN or SNN, we may be able to look at data with a new perspective and perhaps explain the theory of artificial intelligence or brain-Inspired intelligence in another way.

ACKNOWLEDGEMENTS

This work was supported by Key Area R&D Program of Guangdong Province with grant No. 2018B030338001.

REFERENCES

- [1] S. Herculano-Houzel, "The remarkable, yet not extraordinary, human brain as a scaled-up primate brain and its associated cost," *Proceedings of the National Academy of Sciences*, vol. 109, no. Supplement 1, pp. 10661-10668, 2012, doi: 10.1073/pnas.1201895109.
- [2] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J Physiol*, vol. 117, no. 4, pp. 500-44, Aug 1952, doi: 10.1113/jphysiol.1952.sp004764.
- [3] L. Lapique, "Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarization," *Journal of Physiology and Pathology*, vol. 9, pp. 620-635, 1907.
- [4] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans Neural Netw*, vol. 14, no. 6, pp. 1569-72, 2003, doi: 10.1109/TNN.2003.820440.
- [5] F. Rosenblatt, "The Perceptron - a Probabilistic Model for Information-Storage and Organization in the Brain," (in English), *Psychological Review*, vol. 65, no. 6, pp. 386-408, 1958, doi: DOI 10.1037/h0042519.
- [6] S. R. y Cajal, *Estructura de los centros nerviosos de las aves*. 1888.

- [7] L. Luo, "Architectures of neuronal circuits," *Science*, vol. 373, no. 6559, p. eabg7285, 2021, doi: doi:10.1126/science.abg7285.
- [8] D. Durstewitz, J. K. Seamans, and T. J. Sejnowski, "Neurocomputational models of working memory," *Nat Neurosci*, vol. 3 Suppl, pp. 1184-91, Nov 2000.
- [9] S. W. Kuffler, "Discharge patterns and functional organization of mammalian retina," *Journal of neurophysiology*, vol. 16, no. 1, pp. 37-68, 1953.
- [10] H. B. Barlow, "Summation and inhibition in the frog's retina," *The Journal of physiology*, vol. 119, no. 1, pp. 69-88, 1953.
- [11] P. A. Guertin, "The mammalian central pattern generator for locomotion," *Brain research reviews*, vol. 62, no. 1, pp. 45-56, 2009.
- [12] M. H. Hastings, A. B. Reddy, and E. S. Maywood, "A clockwork web: circadian timing in brain and periphery, in health and disease," (in eng), *Nat Rev Neurosci*, vol. 4, no. 8, pp. 649-61, Aug 2003, doi: 10.1038/nrn1177.
- [13] V. K. Ojha, A. Abraham, and V. Snasel, "Metaheuristic design of feedforward neural networks: A review of two decades of research," (in English), *Eng Appl Artif Intel*, vol. 60, pp. 97-116, Apr 2017, doi: 10.1016/j.engappai.2017.01.013.
- [14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-Propagating Errors," (in English), *Nature*, vol. 323, no. 6088, pp. 533-536, Oct 9 1986, doi: DOI 10.1038/323533a0.
- [15] S. M. Bohte, J. N. Kok, and H. La Poutre, "Error-backpropagation in temporally encoded networks of spiking neurons," (in English), *Neurocomputing*, vol. 48, pp. 17-37, Oct 2002, doi: Pii S0925-2312(01)00658-0.
- [16] A. Mohemmed, S. Schliebs, S. Matsuda, and N. Kasabov, "Span: Spike Pattern Association Neuron for Learning Spatio-Temporal Spike Patterns," (in English), *Int J Neural Syst*, vol. 22, no. 4, Aug 2012, doi: Artn 1250012
- [17] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-based optimization to spiking neural networks," (in English), *Ieee Signal Proc Mag*, vol. 36, no. 6, pp. 51-63, Nov 2019.
- [18] M. A. Kramer, "Nonlinear Principal Component Analysis Using Autoassociative Neural Networks," (in English), *Aiche J*, vol. 37, no. 2, pp. 233-243, Feb 1991.
- [19] I. J. Goodfellow *et al.*, "Generative Adversarial Nets," (in English), *Advances in Neural Information Processing Systems 27 (Nips 2014)*, vol. 27, pp. 2672-2680, 2014. [Online]. Available: <Go to ISI>://WOS:000452647101094.
- [20] T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps," (in English), *Biological Cybernetics*, vol. 43, no. 1, pp. 59-69, 1982, doi: Doi 10.1007/Bf00337288.
- [21] D. O. Hebb, *The organization of behavior; a neuropsychological theory* (The organization of behavior; a neuropsychological theory.). Oxford, England: Wiley, 1949, pp. xix, 335-xix, 335.
- [22] G. Q. Bi and M. M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," (in English), *Journal of Neuroscience*, vol. 18, no. 24, pp. 10464-10472, Dec 15 1998.
- [23] J. P. Pfister and W. Gerstner, "Triplets of spikes in a model of spike timing-dependent plasticity," *J Neurosci*, vol. 26, no. 38, pp. 9673-82, Sep 20 2006.
- [24] C. J. Watkins and P. Dayan, "Q-learning," *Mach Learn*, vol. 8, no. 3, pp. 279-292, 1992.
- [25] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," (in English), *Nature*, vol. 518, no. 7540, pp. 529-533, Feb 26 2015, doi: 10.1038/nature14236.
- [26] R. J. Williams, "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning," (in English), *Mach Learn*, vol. 8, no. 3-4, pp. 229-256, May 1992, doi: Doi 10.1023/A:1022672621406.
- [27] N. Fremaux and W. Gerstner, "Neuromodulated Spike-Timing-Dependent Plasticity, and Theory of Three-Factor Learning Rules," *Front Neural Circuits*, vol. 9, p. 85, 2015, doi: 10.3389/fncir.2015.00085.
- [28] B. Rueckauer, I. A. Lungu, Y. Hu, M. Pfeiffer, and S. C. Liu, "Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification," *Front Neurosci*, vol. 11, p. 682, 2017, doi: 10.3389/fnins.2017.00682.
- [29] Y. Niv, M. O. Duff, and P. Dayan, "Dopamine, uncertainty and TD learning," *Behav Brain Funct*, vol. 1, p. 6, May 4 2005, doi: 10.1186/1744-9081-1-6.

- [30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," (in English), *Nature*, vol. 521, no. 7553, pp. 436-444, May 28 2015, doi: 10.1038/nature14539.
- [31] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," (in English), *Frontiers in Computational Neuroscience*, vol. 9, Aug 3 2015, doi: ARTN 99
- [32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," (in English), *P IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov 1998.
- [33] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.
- [34] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, no. 1167, pp. 187-217, 1980.
- [35] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "STDP-based spiking deep convolutional neural networks for object recognition," (in English), *Neural Networks*, vol. 99, pp. 56-67, Mar 2018, doi: 10.1016/j.neunet.2017.12.005.
- [36] C. Mead, "Neuromorphic electronic systems," *P IEEE*, vol. 78, no. 10, pp. 1629-1636, 1990.
- [37] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," (in English), *Interspeech*, pp. 338-342, 2014. [Online]. Available: <Go to ISI>://WOS:000395050100069.
- [38] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv:1409.1259*, 2014.
- [39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput*, vol. 9, no. 8, pp. 1735-80, Nov 15 1997, doi: 10.1162/neco.1997.9.8.1735.
- [40] W. Maass and H. Markram, "On the computational power of circuits of spiking neurons," *Journal of computer and system sciences*, vol. 69, no. 4, pp. 593-616, 2004.
- [41] R. Yuste, "From the neuron doctrine to neural networks," (in English), *Nature Reviews Neuroscience*, vol. 16, no. 8, pp. 487-497, Aug 2015, doi: 10.1038/nrn3962.

AN INTELLIGENT NEWS-BASED STOCK PRICING PREDICTION USING AI AND NATURAL LANGUAGE PROCESSING

Sirui Liu¹ and Yu Sun²

¹Orange Lutheran High School, 2222 N Santiago Blvd, Orange, CA 92867

²California State Polytechnic University, Pomona, CA, 91768, Irvine, CA 92620

ABSTRACT

How do you know which stock is the right stock to invest in and have no risk of losing their money [1]? Even though there are analysis specialists out there to collect data to calculate which stock is good to be invested in, ultimately people could not afford the cost of specialists and specialists are not able to be there every minute that you want to find them. Therefore, the app Stock Recommendation is created to solve this problem where stock investment suggestions are available in touch anywhere and anytime [2]. This application helps us with what we want to invest in and gather information from recent news to show us about the public opinions towards the stock that we are looking for. Investors will no longer struggle with the problem that is the stock that they want to invest in, a good stock or a bad stock, so no money will be lost from the investor's pocket and rather, they will gain my money [4].

KEYWORDS

Stock, machine learning, AI.

1. INTRODUCTION

The Great Depression led to the first time that the stock market officially went into everyone's sight with how much the stock market changed society and people's life [9].

statistic #1: On any given day, stocks have roughly a 53 percent chance of rising and a 47 percent chance of falling. Over any given 3-month period, stocks rise 68 percent of the time, dropping the other 32 percent of the time.

statistic #2: A year after the Covid pandemic shut down the economy, stocks have gained 79% from the lows and the market is in a solid position to continue to rally. It's now being led by sectors that had been very unlikely leaders — like energy and industrials.

Some of the people had discovered and studied multiple ways to predict and proposed how the stock system will go while calculating based on the articles and statics that websites like CNN and Yahoo gives on News that allowed the stock buyer to get an understanding of which stock they should be invested in bases on the news articles that they have published about the recent stock movement that was happening [3]. However, a huge percentage of people who buy stocks do not go to a professional stock adviser but instead, they watch news to see the numbers of the stock. Their sources of information are very limited in the limited source of information that they observed from, with samples given of CNN and Yahoo News being mostly the only two sources

David C. Wyld et al. (Eds): CONEDU, CSITA, MLCL, ISPR, NATAP, ARIN - 2022

pp. 147-155, 2022. CS & IT - CSCP 2022

DOI: 10.5121/csit.2022.121011

that normally people who buy stocks get information from. The limitations of only two sources are there to provide information and create a lot of limitations by the source preferences and their subjective opinions about a stock. Other techniques to calculate for should a person invest in a stock, for example, stock analysts [5]. They not only take a lot of time to analyze and give advice, but also charge a lot of money to do the stock investigation suggestions. However, plenty of time, stocks are not able to be calculated (eg. GameStop), and often results in losing huge amounts of money with investigation. A second practical problem is that giant amount of stock buyer do not relate the recent activities of one company to its stock, which forms the problem that they might keep in or sell of a stock because it shows that it is losing money or gaining money based on what the curve shows right now, and not looking forward to the future possible incomes.

Yahoo finance and CNN are two resources of stock market movement that a huge amount of people look up to for the purpose of seeking information and carefully think about their investigation towards a stock [6]. Yahoo finance and CNN are both news resources and search engines that provide information about daily stock's curve and statistics about how much a stock increases or decreases.

The app that is being built is an app that shows either positivity or negativity about whether you should invest in a stock or not. The app uses Google API to find, point out, and analyze the positive or negative words that are being found in the news resources linked in the app, and rate how positive and how negative the news is about the stock and the company that you want to invest it in.

The app gives actual suggestions about should you invest in the stock compared to Yahoo finance that just gives you a lot of statistics about the stock but not real suggestions about should you invest in the stock or not. Also the app links to more search engines compared to Yahoo finance that only has its single source of statistics.

In the application of the stock predictor, we will have two ways to demonstrate the usage. First, we show the validity of the prediction results by separating the training set and validation set. By different ways of partitioning data into training set and validation set, we can validate the accuracy of the stock prediction at each given period from the training set. By comparing the result of each training set with the validation set, a validation matrix can be computed. Through the validation matrix we can analyze the potential fitting of the machine learning function. Second, we analyze the usability of the application through a user likeability survey. Different users will try out the application and provide a subjective response based on their interaction experience. It will be analyzed with whether they agree with the trending or not. They will also rate the application based on its aesthetic value.

Introduction of the background, open problem, solution and special contribution, and paper structureThe rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Understanding the reports

For most people yahoo finance and CNN data reports are too complicated to understand which lead to a point that they start to buy the “wrong” stock that will make them lose their money in the stock market [7]. By 2018 this dropped 8 percentage points to 34%. More alarming, less than one-third of adults understand three basic financial literacy topics by age 40, although many important financial decisions are made decades earlier. This becomes a very important problem since people start to take risks in the stock market and put plenty of their money into stocks, which they end up buying the wrong stock that makes them lose all of their money.

2.2. Choosing a method to enter the stock market

There are too many resources for stock, so it is overwhelming for people to try to enter into the stock market [8]. Thousands of websites, books, magazines, and it can be very overwhelming when people want to find out one single piece of information that they need. Most times too many news articles and news information resources have identical but conflicting sources, which about one thing, each website might have the same information but each website has a different opinion. As a result of causing confusion that makes people not understand and know which stock they should have invested in or which stock is the correct stock to invest in.

2.3. Understanding machine learning websites

A lot of websites already use machine learning, but it's complicated and hard to understand. Some already use linear regression, etc [11]. We use sentiment analysis which is very good at classifying whether pages are happy or sad, good or bad news, so it's easy for people to understand how it works and what it does and it makes people feel more comfortable using our app.

3. SOLUTION

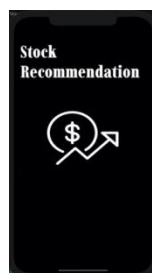


Figure 1. Stock recommendation

The application has been implemented using python and flutter, and it is carefully developed to serve as a multi-functional platform to support the visually-impaired population in navigation, during natural disasters, and in the midst of the COVID-19 pandemic [13]. The application intends to take all aspects into consideration when it provides features like QR code login, locative marker placement, vibration when detected obstacles, alert in face of disasters, GPS-frequency database, and sanitation reminder [14].

The result shows the company's movements that end up reflecting whether you should invest in or not invest into a company based on their recent news articles. By the calculations that API do, they are able to catch emotion words and rate the emotions inside of the words.

The blue links are clickable to actually browsing the website that the API gets information from since the API is still just robots, people might end up having different feelings towards the same word. So blue links that direct to the actual website are provided to let users read it themselves and think about it if they do not trust the result the app gave.

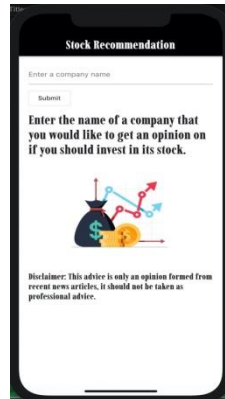


Figure 2. Screenshot of using page

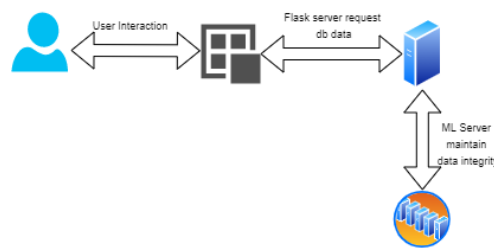


Figure 3. An overview of the project



Figure 4. Screenshot of related articles

The application is designed with two main components, the front-end display and the back-end server that provides the data to be displayed. For the front end we used flutter to design the front end through the graphical interface. In the front end a user input box is provided for the user to enter the name of the company for the stock they wish to gain more information with. The user

input will be saved and sent to the server through GET request. The server is written in flask with several API that provide necessary information for the front end such as articles through json.

```

35
36 class MyHomePage extends StatefulWidget {
37   MyHomePage({Key key, this.title}) : super(key: key);
38
39   // This widget is the home page of your application. It is stateful, meaning
40   // that it has a State object (defined below) that contains fields that affect
41   // how it looks.
42
43   // This class is the configuration for the state. It holds the values (in this
44   // case the title) provided by the parent (in this case the App widget) and
45   // used by the build method of the State. Fields in a Widget subclass are
46   // always marked "final".
47
48   final String title;
49
50   @override
51   _MyHomePageState createState() => _MyHomePageState();
52 }
53
54 class _MyHomePageState extends State<MyHomePage> {
55   Widget build(BuildContext context) {
56     Timer(
57       Duration(seconds: 5),
58       () => Navigator.of(context).pushReplacement(
59         MaterialPageRoute(builder: (BuildContext context) => InfoPage())
60       );
61     return Scaffold(
62       backgroundColor: Colors.black,
63       body: Container(
64         margin: EdgeInsets.fromLTRB(20, 90, 20, 0),
65         child: Column(
66           mainAxisAlignment: MainAxisAlignment.start,
67           crossAxisAlignment: CrossAxisAlignment.center,
68           children: <Widget>[
69             FittedBox(
70               fit: BoxFit.contain,
71               child: Text('Stock\nRecommendation',
72                 style: TextStyle(color: Colors.white, fontSize: 60, fontFamily: 'Imbue', fontWeight: FontWeight.w600),
73             ),
74           ),
75           Padding(
76             padding: EdgeInsets.fromLTRB(0, 40, 0, 40),
77           ),
78           Image(
79             image: AssetImage('assets/images/stockicon.png'),
80             height: 250,
81           ),
82           Padding(
83             padding: EdgeInsets.fromLTRB(0, 40, 0, 40),
84           ),
85         ],
86       ),
87     );
88   },
89 }

```

Figure 5. Screenshot of code 1

The design of the front end is shown above. There are three main components in the front end, which are the main page, info page, and the results page. The main page displays the question to the user to ask for the company they wish for more information about. After the user enters the information, it will parse the information, send to the information page then redirect to the request to the server (Served at [https://Stock-thing.oxxxm.repl.co/results/\\$company](https://Stock-thing.oxxxm.repl.co/results/$company)). The API will then return a json list to the result page, where the returned information will be split into a list and displayed to the front end.

```

10
17 def getArticles(company):
18     # Init
19     global articles, topic
20     topic = company
21
22     newsapi = NewsApiClient(api_key='5ab7a52681914e49813c2ee13f4141e4')
23
24     d = datetime.datetime.strptime(str(date.today()), "%Y-%m-%d")
25     d2 = str(d - dateutil.relativedelta.relativedelta(days=7))
26
27     # /v2/everything
28     all_articles = newsapi.get_everything(
29         q = topic,
30         sources = 'ars-technica, business-
31         insider, the-verge, bloomberg, engadget, fortune, techcrunch, techradar, the-wall-
32         street-journal, wired',
33         domains = 'marketwatch.com, fool.com, finance.yahoo.com, morningstar.com,
34         seekingalpha.com, investopedia.com, zacks.com, aaii.com, barrons.com,
35         kiplinger.com, cnbc.com, thestreet.com',
36         from_param = str(date.today()),
37         to = d2[:10],
38         language = 'en',
39         sort_by = 'popularity',
40         page_size = 100,
41         page = 1)
42
43     articles = all_articles['articles']

```

Figure 6. Screenshot of code 2

The servers are written in Python using the flask library, where the application is created using `app=Flask(app)` command. The server hosts one API which is `retrieveJson`. In its parameter a company is entered which is passed from the front end. Based on this company name it calls the `getArticles(Company)` function where we use news api to search for related information. Once the results are fetched from the api, the information will be saved in a global variable called `links`. The function named `getScored` will then be used to parse each link's information semantically and generate a score for each returned article. If the article has a score between -15 to 15, the article will be used and displayed at the front end. Lastly, once this information is correctly scored and a result list has been finalized, the `toJson` function will jsonify the results package into json format, and the json results will be returned to the front end.

4. EXPERIMENT

4.1. Experiment 1

A good user experience is as important as a good product. So a perfect solution should have excellent user experience feedback. In order to prove that our solution has the best user feedback, we specially designed a user experience questionnaire base on the US system usability questionnaire rules. We statistics the feedback result from 100 users, Show the user our app for 1-5 minutes, let them explore freely on the functionality. We divide those users into Five different groups. The first group of users ages from 10 - 20, the second group of users ages from 20 - 30, the third group of users ages from 30 - 40, the fourth group of users ages from 40 - 50, the fifth group of users ages from 50 - 60. The goal of the first experiment is to verify high feedback scores shows high performance. We collect the feedback scores form these 5 different group of users and analyze it. Experiments have shown that users who ages from 30 - 40 give the highest result feedback to our app. Which may because of the age between those range are more likely to put their money in stock market [10]. The experiment graph shows below:

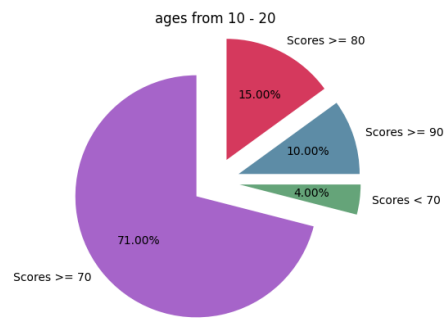


Figure 11. Results of age 10-20

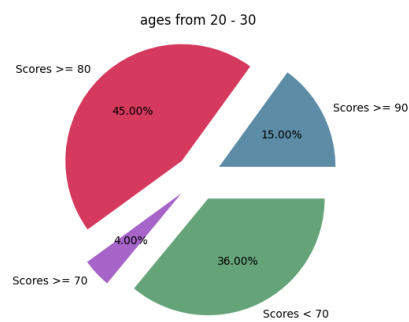


Figure 12. Results of age 20-30

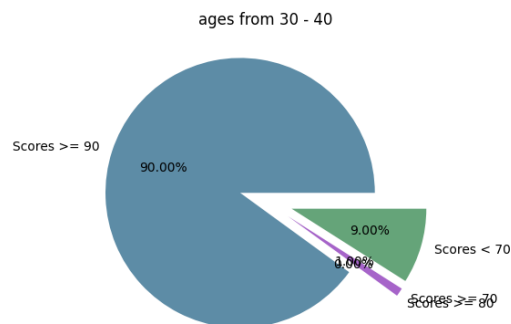


Figure 13. Results of age 30-40

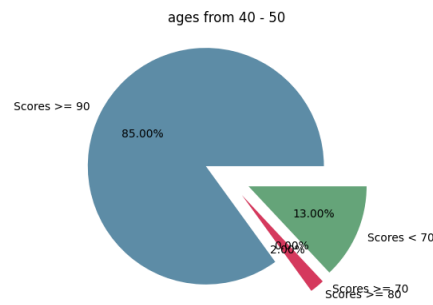


Figure 14. Results of age 40-50

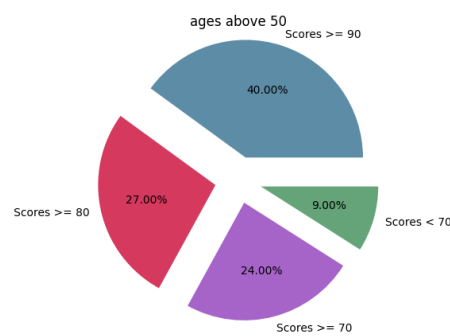


Figure 15. Results of age above 50

5. RELATED WORK

The main contribution is that the data and the api reflect the recent news of a company, and show investors that if the company was positive or negative on the recent news reports and websites.

The application was done by using repl.it and flutter which repl.it was used to write down the code to run the program of the app and the flutter was used to build the app. For example, the font, the color and different pages of the app were built in flutter. Android Studios was used to test out the app on the phone to show the errors and what needs to be improved.

6. CONCLUSIONS

In this paper, we propose a stock trading information collection system to help stock traders to acquire the information they need. We designed a user-end application using flutter,

- Propose a method/an application

In this paper, we proposed a stock information collection system based on a flutter platform using machine learning and front-back end development [12]. Through this application users will be able to enter the name of the company they wish to find more related information and the application will search through the yahoo database then display the results [15].

- Apply the method/application to experiment

The application was then tested through a usability test with participants number of xxx. Each of the users used the application for five minutes and rated the application through a systematic usability test survey.

- Experiment results indicate its effectiveness and solve challenges

The result indicated that the overall usability is above average according to the usability organization, with a score of xx it indicates that the application has an above average usability score and is easy to adapt as a system.

The application has several limitations. First it only allows you to search the keywords for the company only. Yet the application does not have ways to provide more interactive searching methods.

REFERENCES

- [1] Cutler, David M., James M. Poterba, and Lawrence H. Summers. "What moves stock prices?." (1988).
- [2] Walker Z, McMahon DD, Rosenblatt K, Arner T. Beyond Pokémon: Augmented Reality Is a Universal Design for Learning Tool. SAGE Open. October 2017. doi:10.1177/2158244017737815
- [3] Gidofalvi, Gyozo, and Charles Elkan. "Using news articles to predict stock price movements." Department of Computer Science and Engineering, University of California, San Diego (2001): 17.
- [4] Barber, Brad M., and Terrance Odean. "The behavior of individual investors." Handbook of the Economics of Finance. Vol. 2. Elsevier, 2013. 1533-1570.
- [5] Moshirian, Fariborz, David Ng, and Eliza Wu. "The value of stock analysts' recommendations: Evidence from emerging markets." International Review of Financial Analysis 18.1-2 (2009): 74-83.
- [6] Xu, Selene Yue, and C. U. Berkely. "Stock price forecasting using information from Yahoo finance and Google trend." UC Berkley (2014).
- [7] De Bondt, Werner FM, and Richard Thaler. "Does the stock market overreact?." The Journal of finance 40.3 (1985): 793-805.
- [8] Barro, Robert J. "The stock market and investment." The review of financial studies 3.1 (1990): 115-131.
- [9] Barsky, Robert B., and J. Bradford De Long. "Why does the stock market fluctuate?." The Quarterly Journal of Economics 108.2 (1993): 291-311.
- [10] Aggarwal, Rajesh K., and Guojun Wu. "Stock market manipulations." The Journal of Business 79.4 (2006): 1915-1953.
- [11] Su, Xiaogang, Xin Yan, and Chih-Ling Tsai. "Linear regression." Wiley Interdisciplinary Reviews: Computational Statistics 4.3 (2012): 275-294.
- [12] Jordan, Michael I., and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." Science 349.6245 (2015): 255-260.
- [13] Ciotti, Marco, et al. "The COVID-19 pandemic." Critical reviews in clinical laboratory sciences 57.6 (2020): 365-388.
- [14] Tiwari, Sumit. "An introduction to QR code technology." 2016 international conference on information technology (ICIT). IEEE, 2016..
- [15] Callery, Anne, and Deb Tracy Proulx. "Yahoo! cataloging the web." Journal of internet cataloging 1.1 (1997): 57-64.

IDENTIFYING A DEFAULT OF CREDIT CARD CLIENTS BY USING A LSTM METHOD: A CASE STUDY

Jui-Yu Wu and Pei-Ci, Liu

Department of Business Administration,
Lunghwa University of Science and Technology, Taiwan

ABSTRACT

Detecting fraudulent transactions is critical and challenging for financial banks and institutes. This study used a deep learning technique, which is a long short-term memory (LSTM) method, for identifying a default of credit card clients (an imbalanced dataset). To evaluate the performance of optimizers for the LSTM approach, this study employed three optimizers based on gradient methods, such as adaptive moment estimation (Adam), stochastic gradient descent with momentum (Sgdm) and root mean square propagation (Rmsprop). This study used 10-fold cross-validation. Moreover, this study compared the best numerical results of the LSTM method with those of supervised machine learning classifiers, which are back-propagation neural network (BPNN) with a gradient descent algorithm (GDA) and a scaled conjugate gradient algorithm (SCGA). Numerical results indicate that the LSTM-Adam and the BPNN-SCGA classifiers have identical performance, and that selecting an appropriate classification threshold value is important for an imbalanced dataset. Based on the numerical results, the LSTM-Adam classifier can be considered for dealing with credit scoring problems, which are binary classification problems.

KEYWORDS

Deep Learning, Machine Learning, Long Short-Term Memory, Back-Propagation Neural Network, Credit Scoring .

1. INTRODUCTION

For financial banks and institutes, credit card default prediction, credit approval and bankruptcy prediction are significant tasks and challenges. The frauds of credit card can be divided into many activities, such as lost card, card holder not present and counterfeit card [1]. Hence, these tasks of monitoring credit card data and transactions are essential. These issues relate to credit risk management. In credit risk management, many methods have been used, such as judgmental methods, expert systems (e.g. lending committees), statistical models (e.g. credit scoring) and behavioural models [2]. These tasks can be considered as binary classification problems. For solving these classification problems, supervised machine learning (ML) approaches can be considered, such as back-propagation neural networks (BPNN), support vector machines (SVMs), K Nearest Neighbor (KNN) algorithms and random forests. These supervised ML methods have been applied in many fields. For instance, Sehgal [3] used a BPNN classifier for human activity recognition. Lawi and Aziz [4] employed a Least Square SVM (LS-SVM) ensemble classifier for classification of credit card default clients, and indicating that the performance of LS-SVM ensemble classifier is superior to that of a SVM classifier. Vaishnave et. al., [5] applied a KNN classifier for detection and classification of groundnut leaf diseases.

Deep learning (DL) is a subfield of ML and is the multi-layer NNs (more than three layers) that can perform ML algorithms. DL algorithms can learn and extract features from data representation. Many supervised and unsupervised deep learning networks have been developed, such as deep multi-layer perceptrons, convolutional NNs (CNNs), recurrent NNs (RNNs), autoencoder and restricted Boltzmann machine [6]. The CNNs have the capabilities that deal with signals of multi-dimensional arrays and efficiently process imaging problems. The RNNs that contain the information at the previous timesteps are specifically presented to handle sequential signals to capture temporal features. Moreover, the RNNs update the weights of network topology by using an error back-propagation (EBP) algorithm, which causes the limitations of gradient vanishing and exploding. To overcome these drawbacks, a long short-term memory (LSTM) approach with advanced RNN cells has been developed. The LSTM methods have been applied to prediction and classification problems [7, 8].

BPNNs are well-known NNs and have been widely applied to various fields. A conventional BPNN updates the weights of a network topology by using an EBP method, which is a gradient descent algorithm (GDA). The GDA has the limitation that is easily to trap into local optima. To overcome this drawback, a scaled conjugate gradient algorithm (SCGA) has been developed by Moller [9]. Therefore, the SCGA is employed in this study.

To evaluate the performance the LSTM classifier that is a deep learning scheme for binary classification problems, this study used the LSTM approach with three three optimizers (training algorithms), such as adaptive moment estimation (Adam), stochastic gradient descent with momentum (Sgdm) and root mean square propagation (Rmsprop) algorithms to identify a default of credit card clients. The dataset was taken from UCI machine learning repository [10, 11] and is an imbalanced dataset. For the imbalanced dataset, this study employed different *CT* (classification threshold) values, such as 0.5 and 0.3. Furthermore, this study compared the best numerical results obtained by using the LSTM method and with those yielded from BPNN-GDA and BPNN-SCGA classifier.

The rest of this study is organized as follows. Section 2 describes the concept of ML and deep DL, a LSTM method, BPNN scheme and performance evaluation factors of a classifier. Section 3 then introduces the implementation of the LSTM and the BPNN classifier. Next, Section 4 compares the numerical results. Conclusions are finally drawn in Section 5.

2. RELATED WORK

2.1. Machine Learning and Deep Learning

ML algorithms, which consist of supervised, unsupervised and reinforcement learning, are that computers can simulate a human learning, identify and acquire knowledge from the real world and can enhance its performance based on the obtained knowledge on some tasks [12].

1. ML approaches with a supervised learning algorithm can be used to solve forecasting (times series and regression) and classification tasks.
2. ML methods with an unsupervised learning algorithm can be applied to deal with clustering problems.
3. ML systems with a reinforcement learning algorithm that learns the optimal behavior in an environment to yield a maximum reward. The conception of reinforcement learning algorithm is composed of agent, action, discount factor, environment, state, reward, penalty and policy [13].

The difference between the ML and the DL methods are that the DL approaches can extract high-level features from a huge amount of raw data by using a general-purpose learning procedure [6, 14].

2.2. LSTM Method

To improve the limitations of gradient vanishing and exploding of an RNN, a specific cell is introduced into a network topology of a LSTM method. The cell executes a mission of decision making by considering the values of previous memory cell, current input and previous output. The information of the memory cell is then updated by creating a new output value. The network topology of a LSTM approach is shown in Figure 1.

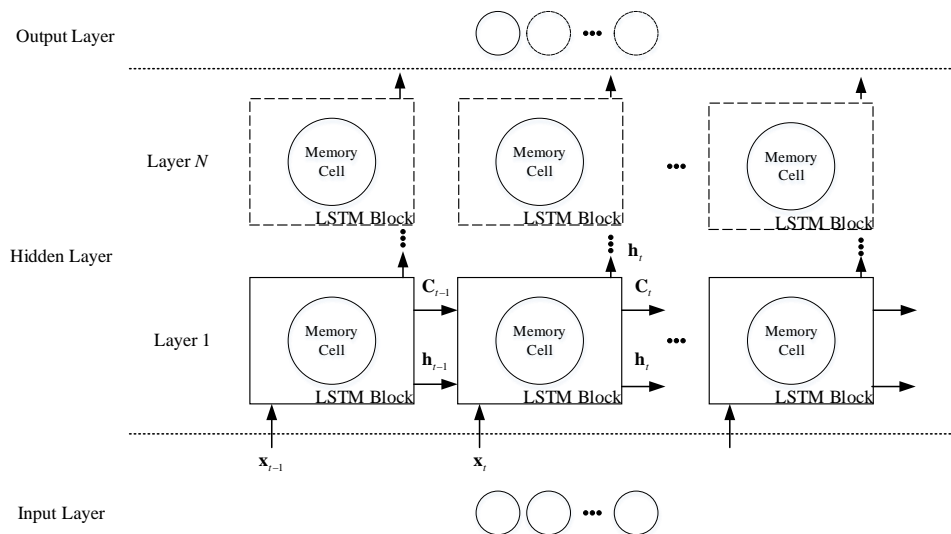


Figure 1. A network topology of a LSTM method

The implementation of a forget gate, an input gate, an output gate and update of a memory cell can be found in the literature [8, 15]. Many optimizers have been used, such as the Adam, the Sgdm and the Rmsprop methods. Chang et. al., [16] presented that a LSTM-Adam works well and is superior to existed optimizers for electricity price forecasting. Hence, this study compared the performance of the LSTM-Adam, the LSTM-Sgdm and the LSTM-Rmsprop classifiers for solving a credit scoring problem.

2.3. BPNN Scheme

A BPNN is a multi-layer NN, which consists of an input layer, some hidden layers and an output layer. Activation functions in input-hidden and hidden-output layers are responsible for biasing the neurons. Moreover, parameter settings for a BPNN include the number of hidden layers, number of hidden neurons, learning rate and momentum term. To evaluate the performance of optimizers based on gradient methods, this study used the BPNN classifiers with the GDA and the SCGA to identify a default of credit card clients and compared the numerical results with those obtained using the LSTM-Adam, the LSTM-Sgdm and the LSTM-Rmsprop classifiers.

2.4. Performance Evaluation Factors of a Classifier

For a binary classification problem, a confusion matrix for visualizing the classification performance can be defined, as shown in Table 1. The confusion matrix is composed of *TP* (true positives), *FN* (false negatives), *FP* (false positives) and *TN* (true negatives). To evaluate the performance of classification models, this study used four factors, such as *Acc* (accuracy), *Pre* (precision), *Rec* (recall) and *F1-Score*.

Table 1. Confusion matrix

Ture condition				factor
Predicted condition		1 (fraudulent)	0 (non-fraudulent)	
	1 (fraudulent)	TP	FP	<i>Pre</i>
	0 (non-fraudulent)	FN	TN	
factor		<i>Rec</i>		

The factor *Acc* represents a percentage of the number of correct predictions and total predictions, as defined by using Eq. (1). The factor can be used to evaluate the effectiveness of a classifier. For balanced dataset, the factor *Acc* can be considered as a main factor.

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

The factor *Pre* is a proportion of the number of *TP* with the number of *TP* and *FP*, as represented by using Eq. (2). The factor *Pre* can be applied to estimate the correctness of positive outcome of a classifier.

$$Pre = \frac{TP}{TP + FP} \quad (2)$$

The factor *Rec* represents a percentage of the number of *TP* with the number of *TP* and *FN*, as defined by using Eq. (3). The factor *Rec* can be employed to evaluate the rate that a classifier correctly identifies actual positives.

$$Rec = \frac{TP}{TP + FN} \quad (3)$$

Increased factor *pre* associates with the decreased factor *Rec*. The factor *F1-Score* balances the factors *Pre* and *Rec*, as expressed by using Eq. (4). This study employed the factor *F1-Score* to evaluate mainly the performance of a classifier.

$$F1-Score = \frac{2 \times Rec \times Pre}{Rec + Pre} \quad (4)$$

3. METHODS

This study used the LSTM-Adam, the LSTM-Sgdm, the LSTM- Rmsprop, the BPNN-GDA and the BPNN-SCGA classifiers to identify a default of credit card clients taken from the literature [10, 11]. This study selected the best numerical results from the LSTM methods and compared the results with those of the BPNN-GDA and the BPNN-SCGA classifiers. A one-way ANOVA is performed. The Fisher test is then executed when the one-way ANOVA is statistically significant.

3.1. Dataset Description

The dataset consists of 23 features and 30,000 instances, which include default instances (fraudulent transactions) of 6,636. The description of features is listed in Table 2. This study employed 10-fold cross-validation. This study used 90% of the dataset to train the LSTM and the BPNN models and employed other 10% to test models.

Table 2. The description of each feature in the dataset

Variables	Features	Description	Type
x_1	LIMIT_BAL	amount of the given credit	numerical
x_2	SEX	gender (1 = male; 2 = female)	categorical
x_3	EDUCATION	education (1 = graduate school; 2 = university; 3 = high school; 4 = others)	categorical
x_4	MARRIAGE	marital status (1 = married; 2 = single; 3 = others)	categorical
x_5	AGE	age (year)	categorical
x_6	PAY_0	history of past payment, the repayment status in September, 2005	numerical
x_7	PAY_2	history of past payment, the repayment status in August, 2005	numerical
x_8	PAY_3	history of past payment, the repayment status in July, 2005	numerical
x_9	PAY_4	history of past payment, the repayment status in June, 2005	numerical
x_{10}	PAY_5	history of past payment, the repayment status in May, 2005	numerical
x_{11}	PAY_6	history of past payment, the repayment status in April, 2005	numerical
x_{12}	BILL_AMT1	amount of bill statement in September, 2005	numerical
x_{13}	BILL_AMT2	amount of bill statement in August, 2005	numerical
x_{14}	BILL_AMT3	amount of bill statement in July, 2005	numerical
x_{15}	BILL_AMT4	amount of bill statement in June, 2005	numerical
x_{16}	BILL_AMT5	amount of bill statement in May, 2005	numerical
x_{17}	BILL_AMT6	amount of bill statement in April, 2005	numerical
x_{18}	PAY_AMT1	amount paid in September, 2005	numerical
x_{19}	PAY_AMT2	amount paid in August, 2005	numerical
x_{20}	PAY_AMT3	amount paid in July, 2005	numerical

Table 2. The description of each feature in the dataset (count.)

Variables	Features	Description	Type
x_{21}	PAY_AMT4	amount paid in June, 2005	numerical
x_{22}	PAY_AMT5	amount paid in May, 2005	numerical
x_{23}	PAY_AMT6	amount paid in April, 2005	numerical
y	Default	1 for fraudulent transactions, 0 otherwise	categorical

3.2. Data Pre-Processing

A data normalization is used to rescale the feature values to create the expected inputs, as defined by using Eq. (5).

$$x'_{ij} = \frac{x_{ij} - \mathbf{x}_i^{\min}}{\mathbf{x}_i^{\max} - \mathbf{x}_i^{\min}} (E_{\max} - E_{\min}) + E_{\min}, \quad (5)$$

$$i = 1, 2, \dots, i_{\max} \quad j = 1, 2, \dots, n_{\text{total}}$$

Where

x'_{ij} = normalized desired output j of input i

\mathbf{x}_i^{\min} = minimum value of input vector i

\mathbf{x}_i^{\max} = maximum value of input vector i

E_{\min} = minimum value of expected output

E_{\max} = maximum value of expected output

n_{total} = total number of a dataset

These values $[E_{\min}, E_{\max}]$ are generally set to $[0.2, 0.8]$.

3.3. Parameter Settings

The parameter settings of the LSTM and the BPNN classifiers are listed in Tables 3. and 4. For each parameter settings, this study used 10-fold cross-validation.

Table 3. The parameter settings of the proposed LSTM classifier

Parameter Settings	Values
Training function	Adam, Sgdm, Rmsprop
Gradient threshold	1
Maximum epoch	1000
Number of hidden units	10, 20, 30, 40, 50
Initial learning rate	0.1
Drop period of a learning rate	500
Drop factor of a learning rate	0.2

Table 4. The parameter settings of the BPNN classifier

Parameter Settings	Values
Training function	trainscg
Learning function	learngdm
Number of hidden neurons	10, 20, 30, 40, 50
Learning rate	0.1
Transfer function in a hidden layer	tansig
Transfer function in an output layer	purelin
Terminal conditions	maximum epoch = 1000 or reaching the error goal = 0.000001

3.4. Classification Threshold Value

For an imbalanced dataset, a *CT* (classification threshold) value must be carefully defined. This study employed the *CT* values 0.3 and 0.5, as expressed by using Eq. (6).

$$\begin{cases} \text{class 1, if network output} \geq CT \\ \text{class 0, if network output} < CT \end{cases} \quad (6)$$

4. NUMERICAL RESULTS

The DL and parallel computing toolboxes in the MATLAB 2020b software were executed on a notebook that has an Intel Core (TM) i9-1190H 2.50 GHz and 64 GB RAM. The LSTM (GPU mode) and BPNN classifiers were performed based parameter settings described in subsection 3.3. Several numerical results were summarized, such as mean training *Acc* (%), mean testing *Acc* (%), mean training *Pre* (%), mean training *Rec* (%), mean training *F1-score* (%), mean testing *Pre* (%), mean testing *Rec* (%), mean testing *F1-score* (%) and mean computation time (MCT).

4.1. Numerical results obtained from the LSTM classifier

This study used the LSTM with the Adam, the Sgdm, and the Rmsprop optimizers for the number of neurons {10, 20, 30, 40, 50} by using the *CT* value = 0.3. The best numerical results are shown in Table 5., indicating that the LSTM-Adam classifier can obtain the best mean testing *F1-score* (%).

Table 6. shows the best numerical results obtained from the LSTM-Adam, the LSTM-Sgdm and the LSTM-Rmsprop methods for the number of neurons {10, 20, 30, 40, 50} by using the *CT* = 0.5, and showing that the LSTM-Adam classifier can find the best mean testing *F1-score* (%).

According to Tables 5. – 6., the LSTM-Adam classifier can obtain the best mean testing *F1-score* (%) by using the *CT* = 0.3 for the imbalanced dataset.

Table 5. The best numerical results obtained from the LSTM-Adam, the LSTM-Sgdm and the LSTM-Rmsprop methods with the $CT = 0.3$

Methods	the number of neurons	mean training Acc (%)	mean testing Acc (%)	mean training Pre (%)	mean training Rec (%)	mean training <i>FI-score</i> (%)	mean testing Pre (%)	mean testing Rec (%)	mean testing <i>FI-score</i> (%)	MCT (sec.)
LSTM-Adam	10	80.67	80.63	57.30	49.41	53.05	57.41	49.36	53.03	561.27
LSTM-Sgdm	20	80.36	80.58	56.95	47.04	51.45	57.19	46.68	51.26	541.89
LSTM-Rmsprop	10	80.55	80.26	56.91	49.90	53.15	56.28	48.95	52.31	558.08

Table 6. The best numerical results yielded from the LSTM-Adam, the LSTM-Sgdm and the LSTM-Rmsprop methods with the $CT = 0.5$

Methods	the number of neurons	mean training Acc (%)	mean testing Acc (%)	mean training Pre (%)	mean training Rec (%)	mean training <i>FI-score</i> (%)	mean testing Pre (%)	mean testing Rec (%)	mean testing <i>FI-score</i> (%)	MCT (sec.)
LSTM-Adam	10	81.39	81.30	67.68	30.35	41.90	66.87	30.10	41.43	562.27
LSTM-Sgdm	10	81.12	81.04	69.63	26.03	37.83	69.13	25.42	37.11	561.27
LSTM-Rmsprop	10	81.38	81.17	68.31	29.50	41.19	66.84	28.94	40.33	561.02

4.2. Numerical results obtained from the BPNN classifier

This study employed the BPNN classifiers with the SCGA and GDA for the number of neurons {10, 20, 30, 40, 50} by using the CT value = 0.3. The best numerical results are shown in Table 7., indicating that the BPNN-SCGA classifier can yield the best mean testing *FI-score* (%).

Table 7. The best numerical results obtained from the BPNN-SCGA and the BPNN-GDA classifiers with $CT = 0.3$

Methods	the number of neurons	mean training Acc (%)	mean testing Acc (%)	mean training Pre (%)	mean training Rec (%)	mean training <i>FI-score</i> (%)	mean testing Pre (%)	mean testing Rec (%)	mean testing <i>FI-score</i> (%)	MCT (sec.)
BPNN-SCGA	10	80.69	80.57	57.62	48.38	52.56	57.20	48.04	52.14	8.75
BPNN-GDA	50	80.03	80.11	56.34	43.58	49.10	56.37	43.39	48.98	12.42

Table 8. shows the best numerical results obtained from the BPNN-SCGA and the BPNN-GDA classifiers for the number of neurons {10, 20, 30, 40, 50} by using the $CT = 0.5$, showing that the best mean testing *FI-score* (%) obtained from the BPNN-SCGA classifier is strongly superior to that of the BPNN-GDA classifier.

Referring to Tables 7. – 8., the numerical results obtained from the BPNN-SCGA and the BPNN-GDA classifiers by using the $CT = 0.3$ are superior to those yielded from the BPNN-SCGA and the BPNN-GDA classifiers by using the $CT = 0.5$.

Table 8. The best numerical results yielded from the BPNN-SCGA and the BPNN-GDA classifiers with the $CT = 0.5$

Methods	the number of neurons	mean training <i>Acc</i> (%)	mean testing <i>Acc</i> (%)	mean training <i>Pre</i> (%)	mean training <i>Rec</i> (%)	mean training <i>F1-score</i> (%)	mean testing <i>Pre</i> (%)	mean testing <i>Rec</i> (%)	mean testing <i>F1-score</i> (%)	MCT (sec.)
BPNN-SCGA	10	82.02	81.97	67.42	36.19	47.09	66.76	35.89	46.64	8.65
BPNN-GDA	50	78.45	78.47	70.73	4.32	8.08	71.12	4.51	8.40	12.22

4.3. Comparison

Table 9. Lists the comparison of the best numerical results obtained from the LSTM-Adam, the BPNN-SCGA and the BPNN-GDA classifiers. For fraud issues, a classifier can correctly detect the factor *Rec* is important, miss fraud may cause a critical risk. Therefore, this study focuses on the mean testing *Recs*. A one-way ANOVA was performed, and indicating that the *P* value (0.026) is smaller than or equals to a significant level 0.05, and that at least two mean testing *Recs* are statistically different. The Fisher test was then executed, showing that the mean testing *Rec* obtained from the LSTM-Adam and the BPNN-SCGA classifiers are identical and are superior to that of the BPNN-GDA classifier. As shown in Table 9., the performance of the LSTM-Adam and the BPNN-SCGA classifiers is similar and spent MCT of the LSTM-Adam classifier is more than that of the BPNN-SCGA classifier.

Learning algorithms of the LSTM-Adam, the BPNN-SCGA and the BPNN-GDA classifiers are EBP algorithms based on gradient methods. The LSTM-Adam and the BPNN-SCGA methods can overcome the drawback that traps into local optima of the standard BPNN-GDA approach.

Table 9. Comparison of the best numerical results obtained from the LSTM-Adam, the BPNN-SCGA and the BPNN-GDA classifiers

Methods	mean testing <i>Pre</i> (%)	mean testing <i>Rec</i> (%)	mean testing <i>F1-score</i> (%)	MCT (sec.)
LSTM-Adam	57.41	49.36	53.03	561.27
BPNN-SCGA	57.20	48.04	52.14	8.75
BPNN-GDA	56.37	43.39	48.98	12.42

5. CONCLUSION

This study used the LSTM-Adam, the LSTM-Sgdm, the LSTM-Rmsprop, the BPNN-SCGA and the BPNN-GDA classifiers for identifying a default of credit card clients, which is an imbalanced dataset. This study employed 10-fold cross-validation. Many remarks are given. For the imbalanced dataset, a *CT* value must be carefully selected. For the LSTM method, the performance of the Adam optimizer is superior to the Sgdm and the Rmsprop algorithms. The LSTM-Adam and the BPNN-SCGA classifier can achieve identical performance. Therefore, the LSTM-Adam classifier has the potential to deal with credit scoring problems, which are binary classification problems.

6. FUTURE WORK

Future work will compare the performance of the LSTM-Adam classifier with those of supervised ML algorithms, such as an SVM classifier (statistical based algorithm), a KNN classifier (instance-based learners) and a Logistical regression method. Moreover, this study will use the resampling methods (such as the random undersampling and the random oversampling methods) for the imbalanced dataset.

REFERENCES

- [1] Khan, F. N., Khan, A. H. & Israt, L., (2020) "Credit card fraud prediction and classification using deep neural network and ensemble learning," in 2020 IEEE Region 10 Symposium (TENSYP), pp 114-119.
- [2] Brown, K. & Moles, P., (2014) Credit Risk Management. Edinburgh, United Kingdom: Edinburgh Business School.
- [3] Sehgal, S., (2018) "Human activity recognition using BPNN classifier on HOG features," in 2018 International Conference on Intelligent Circuits and Systems (ICICS), pp 286-289.
- [4] Lawi, A. & Aziz, F., (2018) "Classification of credit card default clients using LS-SVM ensemble," in 2018 Third International Conference on Informatics and Computing (ICIC), pp. 1-4.
- [5] Vaishnnave, M. P., Devi, K. S., Srinivasan, P. & Jothi, G. A. P., (2019) "Detection and classification of groundnut leaf diseases using KNN classifier," in 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), pp 1-5.
- [6] Zhu, T., Li, K., Herrero, P. & Georgiou, P., (2021) "Deep learning for diabetes: A Systematic review," IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 7, pp 2744-2757.
- [7] Chaabane, F., Réjichi, S. & Tupin, F., (2020) "Comparison between multitemporal graph based classical learning and LSTM model classifications for sits analysis," in 2020 IEEE International Geoscience and Remote Sensing Symposium, pp 144-147.
- [8] Chen, D., Zhang, J. & Jiang, S., (2020) "Forecasting the short-term metro ridership with seasonal and trend decomposition using loess and LSTM neural networks" IEEE Access, vol. 8, pp 91181-91187.
- [9] Moller, A. F., (1993) "A scaled conjugate gradient algorithm for fast supervised learning," Neural Networks, vol. 6, no. 4, pp 525-533.
- [10] Dua, D. & Graff, C. (2019). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.
- [11] Yeh, I. C. & Lien, C.H., (2009) "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," Expert Systems with Applications, vol. 36, no. 2, Part 1, pp 2473-2480.
- [12] Portugal, I., Alencar, P. & Cowan, D., (2018) "The use of machine learning algorithms in recommender systems: A systematic review," Expert Systems with Applications, vol. 97, pp 205-227.
- [13] Gupta, G. & Katarya, R., (2021) "A study of deep reinforcement learning based recommender systems," in 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC), pp. 218-220.
- [14] Hsu, T. C., Liou, S. T., Wang, Y. P., Huang, Y. S. & Che, L., (2019) "Enhanced recurrent neural network for combining static and dynamic features for credit card default prediction," in 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 1572-1576.
- [15] Olah, C. (2021). Understanding LSTM networks. Available: <http://colah.github.io/posts/2015-08-understanding-lstms/>
- [16] Chang, Z., Zhang, Y. & Chen, W., (2018) "Effective Adam-optimized LSTM neural network for electricity price forecasting," in IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), pp. 245-248.

AN INTELLIGENT SENSOR MOBILE PHONE ASSISTING SYSTEM USING AI AND MACHINE LEARNING

Ruilang Liang¹ and Yu Sun²

¹Brea Olinda High School, 789 N Wildcat Way, Brea, CA 92821

²California State Polytechnic University, Pomona,
CA, 91768, Irvine, CA 92620

ABSTRACT

Technology is taking over the world [7]. Thus, how elderly people can request for help when they use a mobile device if there is anybody around them? In this paper, we address this issue by providing a system that can share and remote control a mobile device in real time [8]. An Android mobile app has been developed as an assistant tool. Thus, when a user needs help, she/he uses an unique ID, sends a request and shares the mobile screen, so the helper sees the sharing screen in his/her device and assists the person who needs help. We applied our application to data analysis and accurate measurements. For the accurate measurement, we conducted diverse experiments to observe the stability of use in different devices, and the influence of geographic, environmental, and network factors. The result shows there are no interrupts during the 30 experiments, which means that the system is stable for use and the network speed is the main factor which affects the average connection delay. For the data analysis, we advertised the Mobile App in communities and schools and received a total of 20 feedback questionnaires. We observe that users from 66 - 70 yield the highest positive score.

KEYWORDS

Machine Learning, Screen Remote Sharing, Mobile APP.

1. INTRODUCTION

Working and living outside all year round, not around their parents. In addition to the phone, we also want to communicate with our parents in time, so that they can understand our state and life. The rich and colorful social software is easy for us to operate [9]. For parents who are not good at playing with mobile phones, the difficulty is sometimes no less than doing a high number of problems without solutions. For this reason, we came up with the idea of remote screen control software that would allow us to use another phone to control the original phone over the network. This is a function to support remote assistance of mobile phones [10]. With a tap on the mobile phone, you can ask for help or help others to solve their mobile phone problems remotely, just as easy and convenient as operating your own mobile phone. Ask a contact in the address book for help. After the contact accepts the request, the contact can control your mobile phone. Accept the help of the other party, remotely view and control the other party's mobile phone, help the other party to modify mobile phone Settings, download and install applications, or remotely doodle on the other party's mobile phone, direct operation. It also supports voice calls between two mobile phones.

There exists multiple software that help to share the screen with other users and might or not control other mobile devices. Some of the software are Skype, TeamViewer and Inkwire Screen Share + Assist [3]. Skype is a very popular social software that people use to communicate and share their screen. Skype users can use Skype through the internet; however it doesn't provide the feature to remote control other mobile devices [11].

The other software is TeamViewer. TeamViewer is a software for personal use that can be used in Android devices. Similarly to Skype it can be used through the internet and share the screen. It allows others to control mobile devices after a user enters the unique ID for the device that needs help. However, to use this remote control feature the helper device and the "help" device need to install 2 different software, TeamViewer Quick Support and TeamViewer for Remote Control. Also, if a user who needs help has an Android device, he/she needs to install an add-on depending on your Android device manufacturer.

Finally, Inkwire Screen Share + Assist is software that is only for Android devices. Like Skype and TeamViewer, it allows users to share their screen. The app does not have the ability to control other devices, but it has a feature to draw on other screens. To use this app the user who shares the screen needs to share his/her unique ID. Even though this app is easy to use, there exists significant lag during the communication.

In this paper, we follow similar approaches that Skype, TeamViewer and Inkwire Screen Share + Assist do in their apps. Our goal is to allow users to share their screen and ask for help when they need it by allowing other devices to control their device screen. We provide an Android app that users can utilize to share their screen and use through the internet.. As different from Skype, our app has the ability to remote control other devices which is helpful when others need assistance. There are good features in TeamViewer and Inkwire Screen Share + Assist like users can remote control other devices to assist others when they need it. Secondly, these apps use a unique ID feature to make the sharing screen and remote control easier and secure between 2 or more devices. Therefore we believe that using a unique ID adds more security when they share their screen since only the users that know the unique ID will be allowed to see the screen of the mobile device that needs help and at the same time make the app easy to use. In our app as different from TeamViewer, users do not need to install 2 different soft wares and additional Add- on to allow the remote control on Android devices.

In two application scenarios, we demonstrate how the above combination of techniques increases the convenience of screen sharing:

1. Experiment 1: In order to have sufficient experimental data, we pick up 10 different groups of phone screens to test if the remote sharing function works well. To prove that the program can run stably, we conducted 3 experiments on each experimental group. In order to detect the influence of geographic, environmental, and network factors on APP performance, We have adopted different control groups for the above factors. Experiments results prove that our application can run stably, The influence of the network environment and geographic location is not obvious.

2. Experiment 2: To investigate our user experience and user satisfaction in UI design, we promoted our products to communities and schools. A total of more than 1,000 students and parents have used our app, We received a total of 20 feedback questionnaires. The questionnaire shows that families with parents in the age group 66 - 70 years of age have the most significant effect and give the highest feedback scores.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Lacking knowledge in Java

At the beginning, I found that my knowledge of Java was far from enough for me to complete the program. As a result, I had to learn more Java while planning my software in order to keep the project going. However, exceptions are my unavoidable problem. This includes Error, Runtime Exception, Exception, throw custom Exception, and so on. Exception encountered before will be hurry-scurry, most anomaly can be solved through mode, there are also many exceptions are due to the developers coding errors caused by, therefore encounters abnormal must first to analyze the causes of abnormal, step by step to the position of the mode for throwing an exception, and then constantly sums up the various reasons of throwing an exception, In the study and work to continuously improve their ability to solve problems. There are two ways to learn exceptions. One is to systematically understand the types of exceptions and understand the causes of the exceptions, apply the methods to the actual problems, and then look for different solutions. Another method is to do a lot of practice in learning. After encountering abnormalities in the process of practice, check the causes of abnormalities and summarize them according to the actual situation.

2.2. Platform problem

Then I realized that the app I wanted to create had to be tested on Android [12]. Since I needed to create an Android app, the best program for me was Android Studio. But when I needed to install virtual machines on Android Studio, I had problems [13]. The two virtual Android phones I needed couldn't exist on my Android Studio at the same time. In the initial learning stage of Android, there are usually many problems. In the learning process of Android, there are many knowledge points and it is difficult to skillfully apply them. It is difficult to skillfully apply what you have learned without a long period of time. There are two ways to use ragmen: statically loaded and dynamically loaded. I usually use the dynamic approach. There are three ways to write fragments. List Fragment Dialog fragment Load the layout object in the on Create View method and set the value in the on View Created method. Finally, in the main method you can get the Fragment Manager submission and load the fragment Layout from mian.xml and fragment into the layout file.

2.3. Mastering Java programming ability

To sum up, in order to develop this APP, I need to master some JAVA programming ability and be able to make use of JAVA programming smoothly. In addition, you must learn to master some basic knowledge of Linux, which is based on the android system design foundation. In addition, we should also learn some basic knowledge of database and network protocol, which will be involved in the design of mobile app. Of course, it is also important to master the operation of some development platforms, such as AppmakrAppMakr, AppCanAppCan and Ling. The whole

app may be simple or complex, and the difference of application functions of different apps also leads to different technical implementation or algorithm model. Generally speaking, I need to know the following essential aspects from design to final implementation of this app:

1. Preliminary requirements planning and information, interaction design -- you need to develop a complete requirements document, function document, flow chart, sequence diagram.
2. Interaction design and UI design -- Design a basic and perfect prototype diagram and the basic interaction design effect of app, then design a complete UI interface based on these and learn to cut diagrams. Some adaptive material pictures need to be made with 9patch. You also need to understand the conversion between PX, PT and DP, screen density conversion and coefficients between each other, so that your app can adapt to different resolutions. Interaction design requires you to know a lot of man-machine operation skills and experience, master the use of Axure and other interactive tools, UI design requires you to master Photoshop and Illustrator and other operations.
3. Using the DEVELOPMENT environment such as ADT for APP development, you have to master the Java language, familiar with the Android environment and mechanism, which involves a wide range of areas, please learn relevant knowledge according to the project [15].
4. if it is not a stand-alone version of the app, you need to use the server, then you have to master Web Service related knowledge and development language, commonly used ASP.Net, PHP, JSP and so on.
5. Familiar with and able to develop databases.
6. some functions need to do algorithms, which also need certain professional knowledge, especially mathematical basis.
7. Be familiar with API development, including your ability to develop your own API and experience in calling third-party apis.
8. Familiar with TCP/IP, socket and other network protocols and related knowledge.
9. Proficient in App release process, real machine debugging skills, certificates, packaging and shelves.

3. SOLUTION

Mobile Phone Remote Control Software is an application for one phone to control the other phone. We first need to download the app on two different Android phones, and then open it on both phones. Once in the app, click the "Give Help" button on one phone, click "Accept Help" on the other, and enter the same numeric key on both phones (you can use any number as your key, but both phones must enter the same numeric key). At this time click "accept help" mobile phone will jump to your mobile phone desktop, you can operate the phone normally. After clicking the "Help" button, the phone will see the screen of the "Help" phone. The operator can not only see the screen of the "help" phone, but also guide the user of the "help" phone to use the phone by clicking the screen of the "help" phone. Mobile phone screen clicks on the "help", "accept help" to see "help" on the phone's screen mobile phone operation, then users only need to apply these operations themselves also to "accept help" on the phone, give the helper can be accomplished by a mobile phone to guide the other mobile phone users of mobile phone use this action.

We also provide a set of navigation techniques for our system. The following sections describe these components in detail.

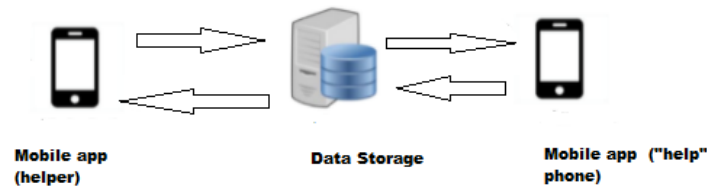


Figure 1. Overview of the solution

The frontend mobile application is developed using Android Studio. Android Studio is an Android integrated development environment (IDE) that is developed specially for Android App[1]. It is written in Java, Kotlin and C++.

To keep tracking the screen of the “help” phone, we implement a foreground service, so that the app can continue to perform a task in the foreground while the user navigates to the screen that he/she needs help. As shown in Fig.2, we developed a function that pops up a notification to the user, so that he/she knows that the app is running in the foreground. In this notification, we created a NotificationManager object to create a channel. Then we use a Notification Builder to add all the widgets that are needed for the notification.

```

private void foregroundify() {
    NotificationManager mgr=
        (NotificationManager)getSystemService(NOTIFICATION_SERVICE);

    if (Build.VERSION.SDK_INT>=Build.VERSION_CODES.O &&
        mgr.getNotificationChannel(CHANNEL_WHATEVER)==null) {
        mgr.createNotificationChannel(new NotificationChannel(CHANNEL_WHATEVER,
            "Default", NotificationManager.IMPORTANCE_DEFAULT));
    }

    NotificationCompat.Builder b=
        new NotificationCompat.Builder(this, CHANNEL_WHATEVER);

    b.setAutoCancel(true)
        .setDefaults(Notification.DEFAULT_ALL);

    b.setContentView(getString(R.string.app_name))
        .setSmallIcon(R.mipmap.ic_launcher)
        .setTicker(getString(R.string.app_name));

    b.addAction(android.R.drawable.ic_menu_add, "notify_record",
        buildPendingIntent(ACTION_RECORD));

    b.addAction(android.R.drawable.ic_menu_save,
        "notify_shutdown",
        buildPendingIntent(ACTION_SHUTDOWN));

    startForeground(NOTIFY_ID, b.build());
}
  
```

Figure 2. Foreground Feature Code

For the database storage, we use Firebase. Firebase is a platform developed by Google that allows you to store files and text information in real time[2]. It provides different features that help to build applications.

In order to share the screen of the “help” device, we implement ImageReader to get the screenshot and send it to Firebase. As shown in Fig. 3, we use FirebaseStorage to store the screenshot of the “help” device. After the screenshot is stored in Firebase Storage, we fetch the

url of FirebaseStorage location of the screenshot and send it to FirebaseDatabase; thus the helper device can get the screenshot of the “help” device.

```
private void uploadFile(File imageFile) {
    FirebaseStorage storage = FirebaseStorage.getInstance("████████████████████");
    // Create a storage reference from our app
    StorageReference storageRef = storage.getReference();

    Uri file = Uri.fromFile(imageFile);
    StorageReference riversRef = storageRef.child("████████" + file.getLastPathSegment());
    UploadTask uploadTask = riversRef.putFile(file);

    // Register observers to listen for when the download is done or if it fails
    uploadTask.addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception exception) {
            // Handle unsuccessful uploads
            Log.e("TEST", "Failed to upload the image");
        }
    }).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
        @Override
        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
            // taskSnapshot.getMetadata() contains file metadata such as size, content-type, etc.
            // ...
            Log.i("TEST", "Upload: " + taskSnapshot);
            riversRef.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                @Override
                public void onSuccess(Uri uri) {
                    Log.i("TEST", "URL: " + uri.getPath());
                    Log.i("TEST", "URL: " + uri);
                    ScreenRecord sr = new ScreenRecord(uri.toString(), System.currentTimeMillis());
                    final FirebaseDatabase database = FirebaseDatabase.getInstance();
                    DatabaseReference ref = database.getReference("remotetutor");
                    ref.child("/") + deviceId).setValue(sr);
                }
            });
        }
    });
}
```

Figure 3. Capture Device Screenshot

To develop the screen for the helper, we show the screen of the other device and track the event when the screen is touched. Thus, when the user requests help, the helper sees the screen of the other device. As shown in Fig. 4 the helper device can see the screenshot of the “help” device, so that the helper instructs the other user where she/he needs to click on the screen. In order to assist the “help” device, the helper clicks on the screen; thus the coordinates of the screen are recorded and sent to the database. In Fig. 5 we fetch the coordinate of the helper screen and send the coordinates to the Firebase.

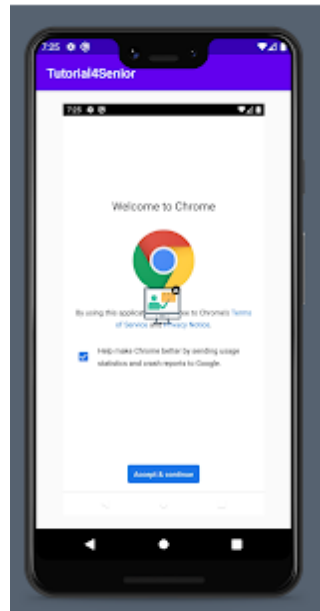


Figure 4. Helper Screen

```
private void post_coords(double x, double y, double xMargin, double yMargin){
    if (x < width / 2 && x - xMargin > 0 )
        x= x-xMargin;
    else if (x + xMargin > width)
        x = x + xMargin;
    if (y < height / 2 && y - yMargin < 0 )
        y= y-yMargin;
    else if (y + yMargin > height)
        y = y + yMargin;
    x = (x / width *100);
    y = (y / height *100);
    Log.i("TEST", "post: x:" + x + ", y: " + y);
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference ref = database.getReference("remotecontrol");
    ClickRecord cr = new ClickRecord((int)x, (int) y);
    ref.child("/") + deviceId).setValue(cr);
}
```

Figure 5. Coordinate of Helper Screen

Fig. 6 shows the “help” device screen. We can observe an icon with a red target and arrow that indicates where the user needs to click in order to perform the target task. To implement this feature, we use a listener that notify when there is a change in the Firebase Database [14]. Thus, when there is any change in the Firebase Database coordinates of the icon with the red and target and arrow are updated and placed in the corresponding position (See Fig. 7)



Figure 6. “Help” Device Screen

```

public boolean onTouch(View v, MotionEvent event) {

    switch (event.getAction()) {
        case MotionEvent.ACTION_DOWN:
            initialX = params.x;
            initialY = params.y;
            initialTouchX = event.getRawX();
            initialTouchY = event.getRawY();
            return true;

        case MotionEvent.ACTION_UP:
            Log.i("TEST", "Action Up!! isCapturing: " + isCapturing);
            //when the drag is ended switching the state of the widget
            collapsedView.setVisibility(View.GONE);
            if (!isCapturing) {
                isCapturing = true;
                mStatusChecker.run();

                new Handler().postDelayed(new Runnable() {
                    @Override
                    public void run() {
                        expandedView.setVisibility(View.VISIBLE);
                    }
                }, 2000);

                canvasLayout = expandedView.findViewById(R.id.tutorialCanvas);
                canvasLayout.removeAllViews();

                final FirebaseDatabase database = FirebaseDatabase.getInstance();
                DatabaseReference ref = database.getReference("remotecontrol");
                if (ref != null) {
                    ref.child("/") + deviceId).addValueEventListener(new ValueEventListener() {
                        @Override
                        public void onDataChange(@NonNull DataSnapshot snapshot) {
                            if (snapshot.getKey() != null && snapshot.getValue() != null) {
                                Log.i("TEST", "changed: " + snapshot.getKey());
                                Log.i("TEST", "changed: " + snapshot.getValue());
                                ControlRecord cr = snapshot.getValue(ControlRecord.class);
                                Log.i("TEST", "ControlRecord: x: " + cr.getX() + ", y: " + cr.getY());
                                Log.i("TEST", "coord : " + "w: " + width * cr.getX() / 100 + ", h: " + height * cr.getY() / 100);
                                moveClickedView(width * cr.getX() / 100, height * cr.getY() / 100);
                                clickedView.setVisibility(View.VISIBLE);
                            }
                        }
                    });
                }
            }
    }
}

```

Figure 7. Update Coordinates of the target Icon

4. EXPERIMENT

4.1. Experiment 1

To evaluate the accuracy of our approach, we have collected 30 real dataset from 10 Sensor and Students Group. In order to compare the approaches, we conducted experiments to verify two aspects: the stability of use in different devices, and the influence of geographic, environmental, and network factors. To test the stability of use in different devices, we ask the 10 different groups to test the sharing function 3 times. The sharing duration of each time is 1 minute, 5 minutes and 10 minutes. The data table shows below:

Group Index	Interrupt Times During 1 minute sharing	Interrupt Times During 5 minutes sharing	Interrupt Times During 10 minutes sharing
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0

Figure 8. Result of experiment 1 (1)

The result shows there are no any interrupts during the 30 experiments, which means that the system is stable for use.

To find the influence of geographic, environmental, and network factors, we collect the connecting time and the average delay of each connection, the result shows below:

Group Index	Distance Between Master Device and Controlled Device	Network Load of Master Device	Network Load of Controlled Device	Delay
1	1km	10 mb/s	10 mb/s	1.1s
2	50km	10 mb/s	10 mb/s	1.1s
3	500km	10 mb/s	10 mb/s	1.3s
4	3000km	10 mb/s	10 mb/s	1.6s
5	10000km	10 mb/s	10 mb/s	2.7s
6	1km	5 mb/s	10 mb/s	1.7s
7	1km	1 mb/s	10 mb/s	2.3s
8	1km	10 mb/s	5 mb/s	1.8s
9	1km	10 mb/s	1 mb/s	2.3 s
10	1km	1 mb/s	1 mb/s	3 s

Figure 9. Result of experiment 1 (2)

The result shows that the network speed is the main factor which affect the average connection delay.

4.2. Experiment 2

To know if our user experience is good and if the user is satisfied with the UI design, we publish our app in the market and advertise the production in communities and schools. A total of more than 1,000 students used our app to help their parents, and we received a total of 20 feedback questionnaires. We collect the data and make a diagram to show the feedback result.

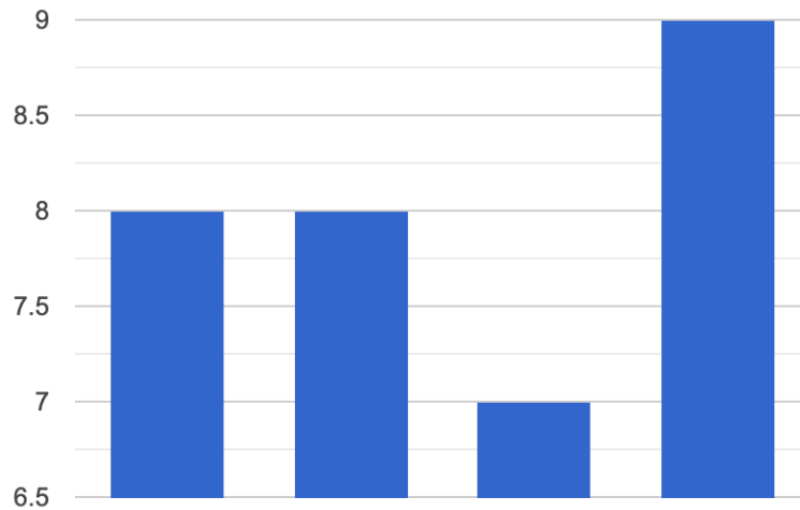


Figure 10. Result of experiment 2

The first from left bar shows the average score while the parents age is between 50 - 55, the second from left bar shows the average score while the parents age is between 56 - 60, the third from left bar shows the average score while the parents age is between 60 - 65, the third from left bar shows the average score while the parents age is between 66 - 70. The result shows the age with 66 - 70 has the highest score.

The first experimental results show that network speed is the main factor affecting the average connection latency. We can solve this problem by increasing the speed of the network. The second experimental result shows that when people aged 66-70 get the highest average score in using this software, we can know our main target group through this result. The results of these two experiments are the same as what I guess, which also meets my expectation for this software.

5. RELATED WORK

Yuanyi Chen presented a system to remote control Android Mobile Phone by using a computer[4]. In the paper, the author explained that developers need to understand and identify the relationship between the four components, active page, service, content provider and broadcast receiver, of the Android system in order to create a remote control application. Also, the author described how the remote control system works from PC device to the server and mobile device. Our application has a similar approach. We use Wifi to send the information from one device to the other. However, our app uses another mobile device as a controller instead of PC, which makes the controller device be more efficient at the time to help another user since most of the people carry the mobile device.

Sørensen H. et al presented a wireless system to share screens in video calls [5]. They proposed a system that can share both digital content as well as physical artifacts in a video call. Our app is similar to this system, our system mirroring the screen in realtime. However, our system is not for a video call and not only for screen share; it also provides remote control.

Bi L. et al proposed a system to remote control power point play in computers without installing any program in mobile devices. It uses Java Native Interface (JNI) technology to control the windows system's function [6]. In our research, we use Android Studio that uses JNI in order to

compile our code. As different from this paper, we control the screen of other mobile devices to provide help instead of remote control power point play.

6. CONCLUSIONS

What if older people need our help to use their phones, but we're not around? That way, when the aforementioned emergency actually happens, they just need to open a simple app on their phone, and the person on the other side can use the app to control the older person's phone with their phone and guide them step by step through the phone to use the phone remotely. So I created a mobile remote control app that allowed us to display a page on a mobile phone on another mobile phone. Through experiments, we know that this method needs to run under good network conditions. And through experiments, we can know that this software is very suitable for the elderly, which is also what I hope this software can solve.

Currently, the app is only available on Android phones, which is its limitation. Then there's the latency of running the software on both phones, so when one phone enters a command, the other phone takes a long time to sync the command and has to manually refresh the screen. Finally, we cannot directly control all operations of another mobile phone through one mobile phone, but can only use one mobile phone to guide another mobile phone, and then the operator needs to operate the mobile phone according to the instructions.

In the future, I will optimize the program to reduce latency when using the software. Updates will be pushed to upgrade the functionality of the software to directly control all operations from one phone to the other. Finally, the compatibility of software on the Apple system is solved.

REFERENCES

- [1] Developers, Android. "Download Android Studio and SDK tools." linha]. Disponível em: <https://developer.android.com/studio/index.html>. [Acedido: 03-Mai-2019] (2015).
- [2] Ayewah, Nathaniel, and William Pugh. "The google findbugs fixit." Proceedings of the 19th international symposium on Software testing and analysis. 2010.
- [3] Machiry, Aravind, Rohan Tahiliani, and Mayur Naik. "Dynodroid: An input generation system for android apps." Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering. 2013.
- [4] Chen, Yuanyi. "Research on Application System of Remote-Control Computer of Android Mobile Phone." Journal of Physics: Conference Series. Vol. 1992. No. 2. IOP Publishing, 2021.
- [5] Sørensen, Henrik, et al. "Wireless smartphone mirroring in video calls." IFIP Conference on Human-Computer Interaction. Springer, Cham, 2015.
- [6] Tan, Gang, et al. "Safe Java native interface." Proceedings of IEEE International Symposium on Secure Software Engineering. Vol. 97. 2006.
- [7] Burke, Andrew. "Ultracapacitors: why, how, and where is the technology." Journal of power sources 91.1 (2000): 37-50.
- [8] Punja, Shafik G., and Richard P. Mislán. "Mobile device analysis." Small scale digital device forensics journal 2.1 (2008): 1-16.
- [9] Osterweil, Leon. "Software processes are software too." Engineering of Software. Springer, Berlin, Heidelberg, 2011. 323-344.
- [10] Fakourfar, Omid, et al. "Stabilized annotations for mobile remote assistance." Proceedings of the 2016 CHI conference on human factors in computing systems. 2016.
- [11] Chen, Kuan-Ta, et al. "Quantifying skype user satisfaction." ACM SIGCOMM Computer Communication Review 36.4 (2006): 399-410.
- [12] Developers, Android. "What is android?." Dosegljivo: <http://www.academia.edu/download/30551848/andoid--tech.pdf> (2011).
- [13] Enck, William, Machigar Ongtang, and Patrick McDaniel. "Understanding android security." IEEE security & privacy 7.1 (2009): 50-57.

- [14] Moroney, Laurence. "The firebase realtime database." *The Definitive Guide to Firebase*. Apress, Berkeley, CA, 2017. 51-71.
- [15] Joorabchi, Mona Erfani, Ali Mesbah, and Philippe Kruchten. "Real challenges in mobile app development." *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. IEEE, 2013.

© 2022 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.

TRANSFORMER BASED ENSEMBLE LEARNING TO HATE SPEECH DETECTION LEVERAGING SENTIMENT AND EMOTION KNOWLEDGE SHARING

Prashant Kapil and Asif Ekbal

Department of Computer Science and Engineering, IIT Patna, India

ABSTRACT

In recent years, the increasing propagation of hate speech on social media has encouraged researchers to address the problem of hateful content identification. To build an efficient hate speech detection model, a large number of annotated data is needed to train the model. To solve this approach we utilized eleven datasets from the hate speech domain and compared different transformer encoder-based approaches such as BERT, and ALBERT in single-task learning and multi-task learning (MTL) framework. We also leveraged the eight sentiment and emotion analysis datasets in the training to enrich the features in the MTL setting. The stacking based ensemble of BERT-MTL and ALBERT-MTL is utilized to combine the features from best two models. The experiments demonstrate the efficacy of the approach by attaining state-of-the-art results in all the datasets. The qualitative and quantitative error analysis was done to figure out the misclassified tweets and the effect of models on the different data sets.

KEYWORDS

BERT, Multi-task learning, Hate speech, Transformer, Ensemble.

1. INTRODUCTION

The majority of the post on the social media platform are harmless but some express hatred towards a targeted individual or any group based on some attributes such as religion, nationality, colour, gender, nationality, ethnicity, etc. These posts have detrimental effects on their victims, e.g., victims are more likely to have lower self-esteem and a tendency of suicidal thoughts [1]. The violence due to hate speech has increased worldwide. The USA has seen an increase in hate speech and related violence following the Presidential election. Therefore Governments and social media platforms must build an efficient tool to combat this issue. To detect online hate speech a large number of scientific studies have been done leveraging Machine learning and Deep learning methods. The trend has been shifted to deep learning architectures for feature extraction and training of the classifiers to enhance the performance but they still lack a sufficient number of labelled data. Recently pre-trained language model BERT has shown substantial and consistent improvement in solving the task. Therefore in this paper, we investigated the effects of transferring knowledge from BERT, and ALBERT to distinguish different hate posts trained in single-task learning, multi-task learning paradigm, and stacking of MTL. The semantics of hate speech often contains negative sentiment that is correlated to hate. The effective features from other sources can be used to enhance the performance [2][3]. We are also providing the different definitions of hate used in the existing literatures to collect the data in Table 1. The laws by

different countries regarding the hate speech is in Table 2. The significant contributions of this work are as follows:

Dataset: We utilized eleven bench mark datasets related to hate domain, harassment, aggressiveness, offensiveness, abusive, spam, racist, sexist, etc. Due to high correlation with the sentiment and emotion data we also utilized three sentiment analysis and five emotion data sets that are publicly available.

Model: We investigated the various state of the art models such as BERT,ALBERT in single task learning and multi task learning framework. The sentiment data is also leveraged in multi task training framework. The stack based ensemble of MTL with BERT and ALBERT as the shared encoder trained on the hate, sentiment, and emotion data is utilized.

Error Analysis: The results and errors on the experimented models were analyzed by presenting qualitative and quantitative analysis to highlight some of the errors that need to be rectified to improve the system performance.

The remaining structure of this paper is as follows. A brief overview of the related background literature is presented in Section 2. In Section 3, the datasets used for the experiments is described. Section 4 discusses in detail the proposed methodology and Experimental setup. Section 5 reports the evaluation results and comparisons to the state-of-the-art. Error analysis containing qualitative and quantitative analysis of the obtained results is presented in Section 6. Finally, the conclusion and directions for future research are presented in Section 7.

Table1. Definitions of hate to collect data

Authors	Definition
[4]	a language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate or insult the members of the group
[5]	a speech that denigrates any person or any group based on characteristics like race, color, gender, religion, ethnicity, nationality, sexual preferences etc.
[6]	A tweet is offensive if it contains racist or sexist slur, intention to attack, promote violent crimes, threatening minorities, and stereotyping genders.
[7]	It is a bias-motivated hostile speech aimed at a person or group of people with intentions to injure, dehumanize, harass, degrade and victimizing targeted groups based on some innate characteristics.
[8]	It is defined as abusive speech containing a high frequency of stereotypical words.

Table 2. Laws of different country on hate speech.

Country	Law
USA	Hate speech is legally protected free speech under the First Amendment. However, speech that include obscenity, speech integral to illegal conduct, speech that incites lawless action or likely to produce such activity are given lesser or no protection.
Brazil	According to the 1988 Brazilian constitution racism is an offense with no statute of limitations and no right to bail for the defendant.
Germany	Section 130 of Germany criminal code states incitement to hatred is a punishable offense leading up to 5 years imprisonment. It also states that publicly inciting hate against some parts of population or using insulting malicious slur or defaming to violate their human dignity is a crime.
India	Article 19(1) of the constitution of India protects the freedom of speech and expression. However, article 19(2) states that to protect sovereignty, integrity, and security of the state, to protect decency and morality, defamation and incitement to an event, some restriction can be imposed

Japan	The Hate speech act of 2016 does not apply to groups of people but covers threats and slander to protect.
New Zealand	Their Hate speech act follows Section 61 of the Human Rights Act 1993 that asserts that threatening, abusive contents in any form, words that are likely to create hostility against a group of people on the basis of race, color, ethnicity is unlawful.

2. RELATED WORK

The state-of-the-art approaches try to solve this problem by supervised learning. These methods can be divided into two parts.

A. Classical methods: [4] created unigram, bigram, and trigram weighted by TF-IDF. The syntactic structure is captured by the Part of speech (POS) tag. The sentiment score and readability of all the tweets along with surface-level features were merged to fed into a logistic regression, naive bayes, support vector machine, decision trees, and random forests. In [5] the features included were unigram, bigram, trigram, and four grams for each tweet, and user-based features such as location, and gender is fed into a logistic regression to solve the task. [6] leveraged characters 3-5 grams, unigram, and bigrams as the n-gram features. The linguistic features along with syntactic features such as POS and dependency relations to detect hate speech.

B. Deep neural networks: In the last 5 years the neural network-based approaches outperformed the traditional classical methods as the former can capture more abstract features helpful in the classification. [7] proposed a CNN_GRU where the first layer is a word embedding layer. The features from the embedding layer followed by the dropout are fed into 1D Convolutions with 100 filters and a window size of 4. The extracted features from the CNN are fed into the GRU for the final classification. [8] proposed a deep learning architecture that utilizes the user features and network features combined with automatically extracted hidden patterns within the text of tweets. [9] investigated deep neural networks, namely Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) by initializing word embeddings with random embedding, FastText word embeddings [10] and GloVe word embeddings [11] using data by [5]. [12] constrained their work to binary classification between *abusive* and *not abusive*. Their character-based approach outperformed token-based and distributional-based features on the dataset by [6]. [13] trained four CNN models, based on character 4-grams, word vectors based on semantic information built using word2vec, randomly generated word vectors, and word vectors combined with character n-grams utilizing the dataset by [5].

[14] considered *SentiWordNet*, *Affinn*, *Bing Liu*, *General Inquirer*, *Subjectivity clues* and *NRC* to explore the relationship between sentiment and toxicity in social media messages from 3 domains, namely Reddit, Wikipedia talk labels and Toxic comment classification. The toxicity detector is a Bi-GRU layer with words represented by 300d FastText pre-trained word embeddings [10] characters represented by 60 dimensions one-hot vector and 3 sentiment values obtained from 3 best lexicons based on their study. These input values are then concatenated together into a vector of 363 dimensions. [15] proposed a transfer learning approach advantaging the pre-trained language model BERT to enhance the performance of the hate speech detection system and generalize it to new datasets. [16] proposed a hybrid methodology to infuse external knowledge into a supervised model for abusive language detection. The external knowledge is lexical features with BERT at the sentence or term level. Transformer-based BERT outperforms traditional deep neural networks in all the tasks. In the semeval task6, 7 out of the top 10 teams used BERT with variations in the parameters and the pre-processing steps. [17] described the architecture of BERT-CNN which utilizes the merged output of the last four layers of BERT to pass into the convolution layer.

Table 3. Statistics of hate data used in the experiment

Datasets	Training	Testing	Inter Annotator Agreement score
D1	Hate: 1430, Offensive:19190 Neutral: 4163	Cross Validation	0.92
D2	Racism:1923, Sexism:2871 Neutral:10682	Cross Validation	0.84
D3	OAG: 3419, CAG: 5297 NAG: 6285	Cross Validation	0.72
D4	Offensive:4400, Non-Offensive:8840	Cross Validation	0.83
D5	Harassment: 5285, Neutral: 15075	Cross Validation	0.84
D6	HOF: 2261, NOT: 3591	HOF: 288, NOT: 865	0.61
D7	Hate: 4210, Neutral: 5790	Hate: 1260 Neutral: 1740	0.62
D8	Hate: 1097, Neutral: 8571	Cross Validation	0.36
D9	OAG: 548, CAG: 570 NAG: 4211	OAG: 286, CAG: 224 NAG: 690	0.69
D10	HOF: 2501, NOT: 1342	HOF: 483, NOT: 798	--
D11	Hate: 4965, Normal: 53851 Spam : 14030, Abusive: 27150	Cross Validation	0.70

3. DATA SETS

In this section we will be briefly describe all the 11 datasets related to the hate domain used in this paper. The statistics of all the hate related data is in Table 3.

3.1. Hate Domain Data

Data 1(D1) [4]: They begin with a hate speech lexicon containing words and phrases identified by internet users as hate speech, compiled by hatebase.org. Using Twitter API, 33458 user data were crawled. A random sample of 25K tweets was manually coded by Crowdsourcing workers. The tweet was categorized into hate, offensive, and neutral. The intercoder-agreement score provided by the CF is 92%. Only 5% of tweets were tagged as hate by the majority of coders and only 1.3% were coded unanimously, demonstrating the imprecision of the hate lexicon. While f*g, b***h, n**ga are used in both offensive and hate speech the terms f**got and n**ger is generally associated with hate speech. Many of the tweets considered most hateful contain multiple racial and homophobic slurs.

Data 2(D2) [5] : The data consists of tweets collected over 2 months. In total 16914 tweets were annotated into racism, sexism, and neutral out of 136052 tweets. The corpus is collected by performing an initial manual search of slurs and terms used about religious, sexual, ethnic minorities, and gender. They presented a list of criteria based on critical race theory to identify racist and sexist slurs. The inter-annotator agreement score is 0.84. 85% of all disagreements occur in the annotation of sexism.

Data 3(D3) [18]: The data is crawled from the public Facebook pages and Twitter. For Facebook, more than 40 pages were crawled which included news websites, web-based forums, political parties, student organizations, etc. For Twitter, the data was collected using some of the popular hashtags such as beef ban, election results, etc. The complete dataset contains 18K tweets and 21K facebook comments annotated with aggression and discursive effects. The inter-annotator agreement for the top level is 72%.

Data 4(D4) [19] : They compiled the Offensive Language Dataset(OLID), where the tweets were annotated using a fine-grained three-layer annotation schema. They retrieved the examples in OLID from Twitter using API and searched for the keywords and constructions often included in 'she is or 'to"Breitbart news. Some of the keywords leveraged are ANTIFA, MAGA, liberals, conservatives, etc. The full datasets consist of 50\% tweets from political and 50\% tweets from non-political keywords. The Fleiss Kappa score is 83\% for the first layer.

Data 5(D5) [20]: It introduces a hand-coded corpus of online harassment data of 35K tweets. It has 15% harassment and 85% non-harassment tweets. They collected a sample of tweets from the blocked user in the Block together. The list terms such as #white genocides, #fuckniggers, #whitepower, #whitelivesmatter, #fucking faggot, #the Jews, etc were searched. Each tweet was labeled by 2 annotators, where the third coder is to break the tie of 2711 tweets. The cohen kappa score is 0.84.

Data 6(D6) [21] : The content was scrapped from Storm front using web-scraping techniques. The extracted forum content was published between 2002 and 2017. A subset of 22 sub-forums covering diverse topics and nationalities is randomly sampled to gather individual posts uniformly distributed among sub-forums and users. The average percentage agreement, Cohen's kappa coefficient, and Fleiss kappa coefficient are 91.03%, 61.4%, and 60.7% respectively. The most occurring hateful words were ape, scum, savages, filthy, mud, homosexuals, etc.

Data 7(D7) [22] : The data have been collected using different gathering strategies in the period from July to September 2018. The different approaches to collecting the tweets are (1) monitoring potential victims of hate accounts, (2) downloading the history of identified haters, and (3) filtering Twitter streams with keywords, i.e. words, hashtags, and stems. The frequent occurring keywords were migrants, refugee, #buildthatwall, bitch, hoe, and women.

Data 8(D8) [23]: the authors searched with heuristics for hate speech in an online forum by identifying the topics for which hate speech can be expected. Different hashtags and keywords were used to sample the posts from Twitter and Facebook. The inter-annotator agreement score obtained is 0.36.

Data 9(D9) [24] : The sampling of the datasets was planned during the extremely hard COVID-19 second wave in India. Therefore during the sampling process, major topics in social media are influenced by COVID-19. To obtain potential hateful tweets, a weak classifier based on an SVM classifier with n-grams features to predict weak labels on the unlabeled corpus. The trending hashtags used to sample the tweets were #resignmodi, #TMCTerror, #chinesevirus, #islamophobia, #covidvaccine, #IndiaCovidcrisis, etc. The inter-annotator agreement score is 69%.

Data 10(D10) [25]: The dataset is collected from various social media platforms namely Facebook, Twitter, and Youtube. The actual sources of information ranged from public posts, tweets, videos, news coverage, etc. The annotation of data involves multiple human interventions and constant deliberations over the justification of assigned tags.

Data 11(D11) [26]: The first step is to collect random tweets by utilizing Twitter API. They collected all the tweets provided by the API over 10 days, consisting of 32 million in total. They store the data in elastic search and basic filtering techniques. They also applied simple text analysis and machine learning to create a boosted set of tweets that will be used to improve the coverage of the minority classes. Finally, they randomly sampled a small data D1 for the exploratory analysis and the remaining D2 for the large scale annotation.

Table 4. Statistics of emotion and sentiment data used in the experiment

Authors	Labels	Total
Kaggle Airline data	Positive, Negative, Neutral	14640
[27]	Positive, Negative, Neutral	20632
[28]	Ekman's Emotion	21051
[29]	Ekman's Emotion	7665
[30]	Ekman's Emotion	13118
[31]	Ekman's Emotion	7303
[32]	Sadness, joy	2585
[33]	Positive, Negative, Neutral	63192

3.2. Sentiment and Emotion Data

Table 4 consists of the sentiment and emotion datasets used for our experiment. We have utilized three sentiment data tagged into positive, negative, and neutral whereas five emotion data is being used tagged based on Ekman's emotion fear, anger, joy, sadness, disgust, and surprise.

4. METHODOLOGY

4.1. Pre-processing

Social media posts contain a lot of noisy texts which are not considered as useful features for the classification. We perform the following steps to remove the noise, and make it ready for machine learning experiments:

1. All the characters like |,;,? were removed along with the numbers and URLs.
2. Words are reduced to lower case so that words such as "BI**H", "bi**h" and "Bi**h" will have the same syntax and will utilize the same pre-trained embedding values.
3. Word segmentation is being done using the Python based word segment to preserve the important features present in hashtag mentions.
4. All the emoticons were categorized into 5 categories, namely *love*, *sad*, *happy*, *shocking* and *anger*. The unicode character of emoticon in text is substituted with one category.
5. All the @ (ex. @abc) mentions were replaced with the common token, i.e *user*.
6. The stop words were not removed due to the risk of losing some useful information, and this was also empirically found to be of little or no impact on the classification performance after removing them.
7. The maximum sequence length is set to 40. Post padding is done if any sentence is less than 40 and pruning is performed from the last if the sentence is greater than 40.

We experimented on 7 transformer based approaches which are discussed in this section.

4.2. Models

1.Model 1(M1):BERT [34] : It stands for Bidirectional Encoder representations from Transformer is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on on both left and right context in all layers. There are two steps in this framework: pre-training and fine-tuning. During pre-training, the model is trained on unlabeled data over pre-training tasks. For the fine-tuning, the BERT model is initialized with the pre-trained parameters and all of the parameters are fine-tuned using labeled data from the downstream tasks. The pre-training of BERT is done by two unsupervised tasks.

Masked LM: This method masks 15% of wordpiece tokens in each sequence at random. The final hidden vectors corresponding to the masked token are fed into an output softmax over the vocabulary. The objective is to predict the masked words rather than reconstructing the entire sentence.

Next Sentence Prediction: In this, the model is trained to understand sentence relationship by pretraining for a binarized next sentence prediction task. When choosing the sentences A and B for each training example 50% of the time B is the actual sentence that follows A and 50% of the time it is the random sentence from the corpus.

In the fine-tuning the task specific inputs and outputs are fed into BERT and all the parameters are fine-tuned end to end. At the output, the CLS representation is fed into an output layer for classification.

2. Model 2(M2) ALBERT [35] :: The design choice of ALBERT uses three new techniques over BERT.

(i) Factorized embedding parameterization: In BERT and RoBERTa the word piece embedding size E is equal to hidden layer size $E=H$. The word piece embeddings learn context independent representations whereas the hidden layers capture context dependent representations. ALBERT in order to make more efficient usage of the total model parameters dictate the $H \gg E$.

(ii) Cross layer parameter sharing: There are multiple ways to share the parameters (i) sharing the feed forward network across layers or only sharing attention parameters. The ALBERT share all parameters across the layers to improve parameter efficiency.

(iii) Inter sentence coherence loss: They propose a loss based on coherence which is sentence order prediction loss that avoids topic prediction and focuses on modelling inter sentence coherence.

The general architecture of transformer encoder block is in Figure 1.

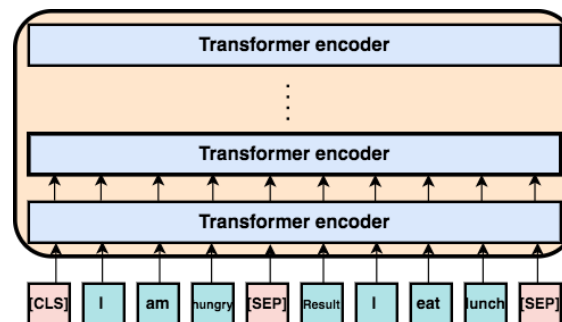


Figure 1. Transformer Encoder

4.3. Multi-Task Learning

Multi-tasking learning aims at solving more than one problem simultaneously. The end-to-end deep multi-task learning has been recently employed in solving various problems of Natural Language Processing (NLP). It enables the model by sharing representations between the related tasks and generalize better by achieving better performance for the individual tasks.

[36] developed two forms of MTL, namely Symmetric multi-task learning (SMTL) and Asymmetric multi-task learning (AMTL). The former is joint learning of multiple classification tasks, which may differ in data distribution due to temporal, geographical, or other variations, and the latter refers to the transfer of learned features to a new task for the purpose of improving the new task's learning performance.

[37] discussed the two most commonly used ways to perform multi-task in deep neural networks.

(i) **Hard Parameter Sharing:** Sharing the hidden layers between all tasks with several task-specific output layers.

(ii) **Soft Parameter Sharing:** Each task has its own specific layers with some sharable part.

In this paper, we leverage a deep multi-task learning framework to leverage the useful information of multiple related tasks. To deal with the data scarcity problem we utilize a multi-task learning approach that enables the model by sharing representations between the related tasks and generalize better by achieving better performance for the individual tasks. Detailed empirical evaluation shows that the proposed multi-task learning framework achieves statistically significant performance improvement over the single-task setting

The architecture of the MTL-DNN is shown in Figure 2. The lower layers are shared across all the tasks, while the top layers represent task-specific outputs. In our experiment all the tasks are classification. The input X is a word sequence (either a sentence or a pair of sentences packed together) represented as a sequence of embedding vectors, one for each word in l_1 . Then the transformer encoder captures the contextual information for each word via self attention, and generates a sequence of contextual embedding in l_2 . In the following, we will describe the model in detail.

Lexicon Encoder (l_1): The input $X = \{x_1, x_2, \dots, x_m\}$ is a sequence of tokens of length m . Following Devlin et al the first token x_1 is always the {CLS} token. If X is packed by a sentence pair (X_1, X_2), we separate the two sentences with a special token [SEP]. The lexicon encoder maps X into a sequence of input embedding vectors, one for each token, constructed by summing the corresponding word, segment, and positional embeddings.

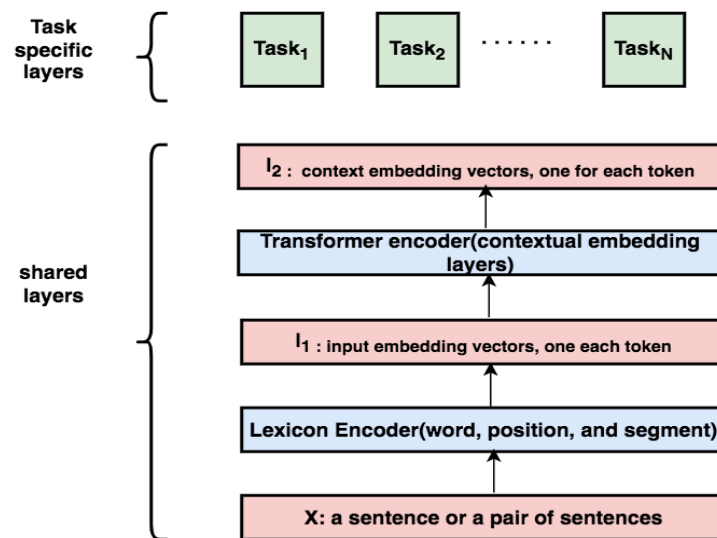


Figure 2. Multi task learning architecture with BERT/ALBERT as shared encoder

Transformer Encoder (I2): It consists of multi-layer bidirectional Transformer encoder (Vaswani et al) to map the input representation vectors(I1) into a sequence of contextual embedding vectors C belongs to $R(d*m)$. This will be the shared representation across different tasks. MT-DNN learns the representation using multi-task objectives, in addition to pre-training.

Single-Sentence Classification Output: Suppose that x is the contextual embedding (I2) of the token [CLS] that can be viewed as the semantic representation of input sentence X . The probability that X is labelled as class c is predicted with softmax:

$$P_r(c|X) = \text{softmax}(W_{SST}^T \cdot x), \quad (1)$$

In the multi-task learning stage, mini-batch based stochastic gradient descent (SGD) is used to learn the parameters of our model. In each epoch, a mini-batch b_i is selected among all the tasks. For the classification tasks the loss function used is categorical cross entropy loss.

$$-\sum_c \mathbb{1}(X, c) \log(P_r(c|X)), \quad (2)$$

Where $\mathbb{1}(X, c)$ is the binary indicator (0 or 1) if class label c is the correct classification for X

We experimented with BERT, and ALBERT as shared encoder in MTL which we termed as **(iii)Model 3(M3):MTL with BERT** and **(iv)Model 4(M4):MTL with ALBERT** as the shared encoder.

4.4. Sentiment and Emotion knowledge

High-quality annotation data is scarce in hate speech detection, which makes the task stereotype words and hence suffer from inherently biased training. Sentiment analysis research has been carried out for many years, and there are abundant high-quality labelled datasets. There is a high degree of correlation between two tasks,

Negative sentiment can be an indicator of hate as reported in the previous research. Therefore, we adopt a multi-task learning method for sentiment knowledge sharing, so as to better extract sentiment features and apply them to hate speech detection.

Model 5(M5): The BERT is used as shared encoder with eleven hate and eight sentiment task trained jointly.

Model 6(M6): The ALBERT is used as shared encoder with eleven hate and eight sentiment task trained jointly.

The same architecture as in Figure 2 is used for M5 and M6.

4.5. Ensemble learning

The Ensemble learning strategy have been proposed to effectively generalize machine learning techniques in several domains including text classification. The existing approaches on ensemble learning have outperformed the baseline classifiers by reducing the variance of predictions. In our experiments we utilized stacking based approach that combines multiple machine learning

algorithm via meta learning. In this, the base level algorithms are trained on complete datasets, the meta-model is trained on the final outcomes of all the base model as the feature. This model is termed as **Model 7(M7): Stack based Ensemble** and figure 3 explains the idea.

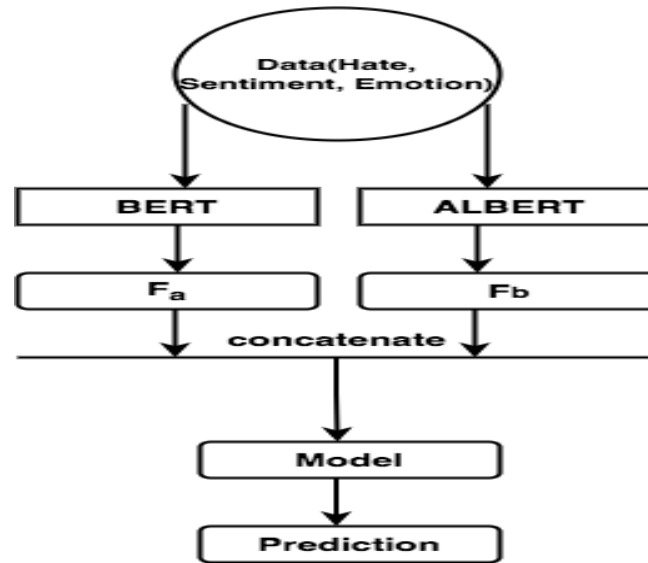


Figure 3. Stack based Ensemble

4.6. Experimental Setup

All the deep learning models were implemented using Keras, a neural network package [38] with Tensorflow [39] as backend. Each dataset is split into an 80:20 ratio to use 80% in grid-search to tune the batch size and learning epochs using 5-fold cross-validation experiments and test the optimized model on 20% held-out data. For some data with the separate test set, model is trained on train data and performance is evaluated using test data. Categorical cross-entropy is used as a loss function, and Adam [40] optimizer is used for optimizing the network.

We use Adam optimizer and $2e-5$ for the transformer models. The batch size of 30 is used to train the shared encoder and epoch of 2 is found to be optimal. The value for bias is randomly initialized to all zeros, Relu activation function is employed at the intermediate layer, and Softmax is utilized at the last dense layer. The transformers library is loaded from Hugging Face. It is a python library providing a pre-trained and configurable transformer model useful for a various NLP tasks.

5. RESULTS, COMPARISON AND ANALYSIS

We report the accuracy and weighted-F1 score of all the eleven datasets in Table 5 and Table 6. Table 7 enlists comparison with the state-of-the-art approaches and the proposed approach over the weighted-F1 score. From the results it can be seen that Ensemble of BERT and ALBERT trained in MTL with sentiment and emotion features outperformed the other methods. The ensemble based approach obtained the best results for all the eleven hate domain data. We are also presenting the qualitative and quantitative analysis on the obtained results to highlight some of the errors that need to be rectified to improve the system performance.

Table 5. Weighted-F1 of eleven datasets.

Datasets	M1	M2	M3	M4	M5	M6	M7
D1	91.10	89.50	92.73	90.81	93.13	90.93	93.63
D2	85.50	84.40	89.66	87.03	89.98	87.36	90.10
D3	78.80	77.80	83.32	82.52	83.51	82.78	83.78
D4	79.70	80.10	83.18	84.57	83.48	84.71	83.92
D5	75.90	77.40	82.54	83.37	82.92	83.63	83.89
D6	80.50	79.10	82.48	82.58	82.89	82.96	83.54
D7	56.40	55.80	59.80	56.62	59.91	56.74	60.14
D8	83.80	83.80	86.58	86.56	87.13	86.82	87.52
D9	79.20	79.98	80.82	81.22	81.46	81.69	81.96
D10	72.30	71.50	76.32	74.23	77.18	74.80	77.78
D11	80.70	80.90	81.93	82.32	82.28	82.46	82.89

Table 6. Accuracy of eleven datasets

Datasets	M1	M2	M3	M4	M5	M6	M7
D1	91.90	90.80	93.60	92.70	94.17	93.10	94.72
D2	86.40	85.41	90.69	87.34	91.93	87.72	91.99
D3	78.90	78.10	85.14	84.43	92.18	90.98	92.63
D4	79.80	80.10	83.93	84.84	84.63	85.34	85.78
D5	76.90	78.50	84.51	84.38	84.98	84.65	85.67
D6	80.93	78.12	87.23	86.43	87.76	83.32	87.99
D7	57.82	59.23	62.64	59.98	63.12	60.78	63.93
D8	85.10	85.10	87.38	90.51	89.10	91.78	92.13
D9	79.84	80.62	81.32	82.54	81.98	84.34	84.96
D10	76.70	74.10	79.83	77.23	81.34	79.32	81.54
D11	80.90	81.50	81.91	82.12	82.67	83.32	83.72

Table 7. Comparison to the state-of-the-art systems and the proposed approach

Best Model (Weighted-F1)	Comparison (Weighted-F1)
D1 (93.63)	[4]: (90), [41]: (91.10)
D2 (90.10)	[42]: (83), [43]: (86), [8]: (87)
D3 (83.78)	[44]: (58.72)
D4 (83.92)	[45]: (72.85), [46]: (78.3)
D5 (83.89)	[41]: (72.75), [47]: (73.6)
D6 (83.54)	[48]:(74.65), [49]:(74.31),
D7 (60.14)	[50]:(54.60), [51]:(51.90),
D8 (87.52)	[52]: (82.01), [53]: (78.40)
D9 (81.96)	[54]:(80.20), [55]:(75.90)
D10 (77.78)	[56]: (80.89), [57]:(81.99)
D11 (82.89)	[58]:(78.40), [58]:(80.10)

5.1. Quantitative Analysis

The sentences in Neutral class play a very crucial role in determining the annotators' global knowledge about any specific topic and how much they can distinguish between free speech or any subtypes of harmful speech. We analyzed the misclassification rate of 5 datasets from one class into other over the baseline BERT model and the best performing stacked MTL using BERT and ALBERT in Table 8. In the M1 for D1 in Table 8 ,9.2% of hate tweets were misclassified to neutral showing the model's ability to distinguish the hateful text. The addition of

emotion and sentiment features in the MTL setting with ALBERT for D1 improved with only 27.42% and 6.6% misclassification to offensive and neutral. The error rate for all the other 4 datasets were improved with MTL based approach. The most notable improvement is 61% improvement in case of harassment class.

5.2. Qualitative Analysis

Table 9 and Table 10 consists of True positive of hate sub class and false positive of hate sub class. We show seven different types of hate speech that were correctly classified by all the models. Some of the misclassified non-hate into hate is shown in Table 9. The lack of adequate contextual information is one of the factor involved due to which model is not able to distinguish non-hate from hate.

Table 8. Misclassification comparison between Model 1(M1) and Model 7(M7)

Model	Class	Misclassification
M1	Hate (D1)	Offensive (63.76%), Neutral (9.2)
M7	Hate (D1)	Offensive (27.42%), Neutral (6.6%)
M1	Offensive (D1)	Hate (3.9%) , Neutral (3.5%)
M7	Offensive (D1)	Hate (2.1%), Neutral (1.2%)
M1	Racism (D2)	Sexism (0.8%), Neutral (23%)
M7	Racism (D2)	Sexism (0.4%), Neutral (8.78%)
M1	Sexism (D2)	Racism (1.34%), Neutral (35.66%)
M7	Sexism (D2)	Racism (0.76%), Neutral (14.03%)
M1	OAG (D3)	CAG (58.07%), NAG (16.61%)
M7	OAG (D3)	CAG (14.73%), NAG (6.8%)
M1	Offensive (D4)	NOT (40.56%)
M7	Offensive (D4)	NOT (15.79%)
M1	Harassment (D5)	Non-Harassment (79.18%)
M7	Harassment (D5)	Non-Harassment (18.65%)

6. CONCLUSION AND FUTURE WORK

Our study is based on the assumption that discourse of hate speech detection involves other affective components such as sentiment and emotion. We have leveraged the labeled corpora for each tasks and experimented on single task learning and multi-task learning paradigm. Our results demonstrates that stack based multi-task architectures are the best performing model and emotion and sentiment knowledge sharing improves system performance and advances hate speech detection. The plausible extensions include the inclusion of more affective phenomenon correlated to hate speech such as sarcasm/irony [59], "big five" personality traits [60], and emotion roe labeling [61].

Table 9. True Postives

Sentence No.	Type	Tweets
1.	Toxic	bitch please whatever
2.	Non-Toxic	@user your sadness is exactly what the terrorist want
3.	Direct Attack	@user the jew are te mastermind idiot
4.	Indirect Attack	@user he is a dumb and dumber
5.	Doubtful	shame on icc now please stop it
6.	References	What an idiot. \#buildthatwall
7.	Annotators Bias	ball is in our court not yours

Table 10. False Positives

1.	Toxic	@user a woman you should not complain about cleaning up your house a man should always take the trash out
2.	External Knowledge	user eebo who want to get there nose in these bad bois then scally chav sock fetish game of basketball hoe
3.	Hate is subjective	user @user love frat boy with soft long

REFERENCES

- [1] Vazsonyi, A. T., Machackova, H., Sevcikova, A., Smahel, D., & Cerna, A. (2012). Cyberbullying in context: Direct and indirect effects by low self-control across 25 European countries. *European Journal of Developmental Psychology*, 9(2), 210-227.
- [2] Zhou, X., Yong, Y., Fan, X., Ren, G., Song, Y., Diao, Y., ... & Lin, H. (2021, August). Hate speech detection based on sentiment knowledge sharing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 7158-7166).
- [3]. del Arco, F. M. P., Halat, S., Padó, S., & Klinger, R. (2021). Multi-task learning with sentiment, emotion, and target detection to recognize hate speech and offensive language.
- [4] Davidson, T., Warmesley, D., Macy, M., & Weber, I. (2017, May). Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 11, No. 1, pp. 512-515).
- [5] Waseem, Z., & Hovy, D. (2016, June). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop* (pp. 88-93).
- [6] Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016, April). Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web* (pp. 145-153).
- [7] Zhang, Z., Robinson, D., & Tepper, J. (2018, June). Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European semantic web conference* (pp. 745-760). Springer, Cham.
- [8] Founta, A. M., Chatzakou, D., Kourtellis, N., Blackburn, J., Vakali, A., & Leontiadis, I. (2019, June). A unified deep learning architecture for abuse detection. In *Proceedings of the 10th ACM conference on web science* (pp. 105-114).
- [9] Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017, April). Deep learning for hate speech detection in tweets. In *Proceedings of the 26th international conference on World Wide Web companion* (pp. 759-760).
- [10] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5, 135-146.
- [11] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).
- [12] Mehdad, Y., & Tetreault, J. (2016, September). Do characters abuse more than words?. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue* (pp. 299-303).
- [13] Gambäck, B., & Sikdar, U. K. (2017, August). Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online* (pp. 85-90).
- [14] Brassard-Gourdeau, E., & Khoury, R. (2019, August). Subversive toxicity detection using sentiment information. In *Proceedings of the Third Workshop on Abusive Language Online* (pp. 1-10).
- [15] Mozafari, M., Farahbakhsh, R., & Crespi, N. (2019, December). A BERT-based transfer learning approach for hate speech detection in online social media. In *International Conference on Complex Networks and Their Applications* (pp. 928-940). Springer, Cham.
- [16] Koufakou, A., Pamungkas, E. W., Basile, V., & Patti, V. (2020). HurtBERT: incorporating lexical features with BERT for the detection of abusive language. In *Fourth Workshop on Online Abuse and Harms* (pp. 34-43). Association for Computational Linguistics.

- [17] Safaya, A., Abdullatif, M., & Yuret, D. (2020, December). Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation* (pp. 2054-2059).
- [18] Kumar, R., Reganti, A. N., Bhatia, A., & Maheshwari, T. (2018). Aggression-annotated corpus of hindi-english code-mixed data. *arXiv preprint arXiv:1803.09402*.
- [19] Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.
- [20] Golbeck, J., Ashktorab, Z., Banjo, R. O., Berlinger, A., Bhagwan, S., Buntain, C., ... & Wu, D. M. (2017, June). A large labeled corpus for online harassment research. In *Proceedings of the 2017 ACM on web science conference* (pp. 229-233).
- [21] De Gibert, O., Perez, N., García-Pablos, A., & Cuadros, M. (2018). Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*.
- [22] Basile, V., Bosco, C., Fersini, E., Debora, N., Patti, V., Pardo, F. M. R., ... & Sanguinetti, M. (2019). Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *13th International Workshop on Semantic Evaluation* (pp. 54-63). Association for Computational Linguistics.
- [23] Mandl, T., Modha, S., Majumder, P., Patel, D., Dave, M., Mandlia, C., & Patel, A. (2019, December). Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th forum for information retrieval evaluation* (pp. 14-17).
- [24] Mandl, T., Modha, S., Shahi, G. K., Madhu, H., Satapara, S., Majumder, P., ... & Jaiswal, A. K. (2021). Overview of the HASOC subtrack at FIRE 2021: Hate speech and offensive content identification in English and Indo-Aryan languages. *arXiv preprint arXiv:2112.09301*.
- [25] Bhattacharya, S., Singh, S., Kumar, R., Bansal, A., Bhagat, A., Dawer, Y., ... & Ojha, A. K. (2020). Developing a multilingual annotated corpus of misogyny and aggression. *arXiv preprint arXiv:2003.07428*.
- [26] Founta, A. M., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., ... & Kourtellis, N. (2018, June). Large scale crowdsourcing and characterization of twitter abusive behavior. In *Twelfth International AAAI Conference on Web and Social Media*.
- [27] Rosenthal, S., Farra, N., & Nakov, P. (2017, August). SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)* (pp. 502-518).
- [28] Mohammad, S. (2012). # Emotional tweets. In * *SEM 2012: The First Joint Conference on Lexical and Computational Semantics—Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)* (pp. 246-255).
- [29] Scherer, K., & Wallbott, H. (1997). The ISEAR questionnaire and codebook. *Geneva Emotion Research Group*.
- [30] Li, Y., Su, H., Shen, X., Li, W., Cao, Z., & Niu, S. (2017). Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.
- [31] Plaza-del-Arco, F. M., Strapparava, C., Lopez, L. A. U., & Martín-Valdivia, M. T. (2020, May). EmoEvent: A multilingual emotion corpus based on different events. In *Proceedings of the 12th Language Resources and Evaluation Conference* (pp. 1492-1498).
- [32] Zhao, J., Liu, K., & Xu, L. (2016). Sentiment analysis: mining opinions, sentiments, and emotions.
- [33] Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., & Cherry, C. (2016, June). Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)* (pp. 31-41).
- [34] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- [35] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.
- [36] Xue, Y., Liao, X., Carin, L., & Krishnapuram, B. (2007). Multi-Task Learning for Classification with Dirichlet Process Priors. *Journal of Machine Learning Research*, 8(1).
- [37] Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.
- [38] Chollet, F. (2018). Keras: The python deep learning library. *Astrophysics source code library*, ascl-1806.

- [39] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.
- [40] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [41] Chakrabarty, T., Gupta, K., & Muresan, S. (2019, August). Pay “attention” to your context when classifying abusive language. In *Proceedings of the Third Workshop on Abusive Language Online* (pp. 70-79).
- [42] Park, J. H., & Fung, P. (2017). One-step and two-step classification for abusive language detection on twitter. *arXiv preprint arXiv:1706.01206*.
- [43] Kshirsagar, R., Cukuvac, T., McKeown, K., & McGregor, S. (2018). Predictive embeddings for hate speech detection on twitter. *arXiv preprint arXiv:1809.10644*.
- [44] Kapil, P., Ekbal, A., & Das, D. (2020). Investigating deep learning approaches for hate speech detection in social media. *arXiv preprint arXiv:2005.14690*.
- [45] Cambray, A., & Podsadowski, N. (2019). Bidirectional recurrent models for offensive tweet classification. *arXiv preprint arXiv:1903.08808*.
- [46] Liu, P., Li, W., & Zou, L. (2019, June). NULI at SemEval-2019 Task 6: Transfer Learning for Offensive Language Detection using Bidirectional Transformers. In *SemEval@ NAACL-HLT* (pp. 87-91).
- [47] Naseem, U., Razzak, I., & Hameed, I. A. (2019). Deep Context-Aware Embedding for Abusive and Hate Speech detection on Twitter. *Aust. J. Intell. Inf. Process. Syst.*, 15(3), 69-76.
- [48] Mishra, S., & Mishra, S. (2019). 3Idiots at HASOC 2019: Fine-tuning Transformer Neural Networks for Hate Speech Identification in Indo-European Languages. In *FIRE (Working Notes)* (pp. 208-213).
- [49] Baruah, A., Barbhuiya, F., & Dey, K. (2019, June). Abaruah at semeval-2019 task 5: Bi-directional lstm for hate speech detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 371-376).
- [50] Ding, Y., Zhou, X., & Zhang, X. (2019, June). Ynu_dyx at semeval-2019 task 5: A stacked bigru model based on capsule network in detection of hate. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 535-539).
- [51] Montejo-Ráez, A., Jiménez-Zafra, S. M., García-Cumbreras, M. A., & Díaz-Galiano, M. C. (2019, June). SINAI-DL at SemEval-2019 Task 5: Recurrent networks and data augmentation by paraphrasing. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 480-483).
- [52] MacAvaney, S., Yao, H. R., Yang, E., Russell, K., Goharian, N., & Frieder, O. (2019). Hate speech detection: Challenges and solutions. *PloS one*, 14(8), e0221152.
- [53] Berglind, T., Pelzer, B., & Kaati, L. (2019, August). Levels of hate in online environments. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 842-847).
- [54] Risch, J., & Krestel, R. (2020, May). Bagging BERT models for robust aggression identification. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying* (pp. 55-61).
- [55] Mishra, S., Prasad, S., & Mishra, S. (2020, May). Multilingual Joint Fine-tuning of Transformer models for identifying Trolling, Aggression and Cyberbullying at TRAC 2020. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying* (pp. 120-125).
- [56] Mitra, A., & Sankhala, P. (2022). Multilingual Hate Speech and Offensive Content Detection using Modified Cross-entropy Loss. *arXiv preprint arXiv:2202.02635*.
- [57] Glazkova, A., Kadantsev, M., & Glazkov, M. (2021). Fine-tuning of Pre-trained Transformers for Hate, Offensive, and Profane Content Detection in English and Marathi. *arXiv preprint arXiv:2110.12687*.
- [58] Lee, Y., Yoon, S., & Jung, K. (2018). Comparative studies of detecting abusive language on twitter. *arXiv preprint arXiv:1808.10245*.
- [59] Reyes, A., Rosso, P., & Buscaldi, D. (2012). From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74, 1-12.
- [60] Flek, L. (2020, July). Returning the N to NLP: Towards contextually personalized classification models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7828-7838).

- [61] Mohammad, S., Zhu, X., & Martin, J. (2014, June). Semantic role labeling of emotions in tweets. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis* (pp. 32-41).

AUTHORS

1. **Prashant Kapil** is a PhD scholar at the Department of CSE at IIT Patna. The author would like to acknowledge the funding agency, the University Grant Commission (UGC) of the Government of India, for providing financial support in the form of UGC NET-JRF/SRF. Research interests: AI, NLP, and ML
2. **Asif Ekbal** is an Associate Professor in the Department of CSE, IIT Patna, India. Research interests: AI, NLP and ML.



AUTHOR INDEX

<i>Asif Ekbal</i>	179
<i>Charles Tian</i>	55
<i>Chenggang He</i>	129
<i>Jarrett Yeo Shan Wei</i>	113
<i>John E. Ortega</i>	43
<i>Jui-Yu Wu</i>	157
<i>Lang Qian</i>	129
<i>Miss Pitchsinee Oimpitiwong</i>	19
<i>Nathan Ji</i>	31
<i>Pei-Ci Liu</i>	157
<i>Pingsheng Li</i>	29
<i>Prashant Kapil</i>	179
<i>Ruilang Liang</i>	157, 167
<i>Ryan Yan</i>	67
<i>Shengjie Zheng</i>	129
<i>Sirui Liu</i>	147
<i>Siu Ming YIU</i>	95
<i>Uranchimeg Tudevdaeva</i>	01
<i>Xiaojian Li</i>	29
<i>Xiaoqi Qin</i>	129
<i>Yeo Chai Kiat</i>	113
<i>Yiqi DENG</i>	95
<i>Yu Sun</i>	31, 55, 67, 83, 147, 167
<i>Zhanhao Cao</i>	83
<i>Zhishuo Zhang</i>	67
<i>Zolzaya Badamjav</i>	01