Computer Science & Information Technology 188

Computer Science, Engineering and Applications

David C. Wyld
Dhinaharan Nagamalai (Eds)

Computer Science & Information Technology

- 13th International Conference on Computer Science, Engineering and Applications (CCSEA 2023)
- 9th International Conference on Artificial Intelligence and Applications (AIFU 2023)
- 12th International Conference on Embedded Systems and Applications (EMSA 2023)
- 4th International Conference on Natural Language Computing and AI (NLCAI 2023)
- 9th International Conference on Networks & Communication (NCOM 2023)
- 9th International Conference on Signal and Image Processing (SIPRO 2023)
- 10th International Conference on Software Engineering and Applications (SEA 2023)
- 11th International Conference on Data Mining & Knowledge Management Process (DKMP 2023)
- 4th International Conference on Big Data and Machine Learning (BDML 2023)
- 4th International Conference on Blockchain and Internet of Things (BIoT 2023)
- 12th International Conference on Cloud Computing: Services and Architecture (CLOUD 2023)

Published By

AIRCC Publishing Corporation

Volume Editors

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

Dhinaharan Nagamalai (Eds)
Wireilla, Australia
E-mail: dhinaharann@gmail.com

ISSN: 2231 - 5403
ISBN: 978-1-925953-90-9
DOI: 10.5121/csit.2023.130501 - 10.5121/csit.2023.130522

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

Preface

13th International Conference on Computer Science, Engineering and Applications (CCSEA 2023), March 18 ~ 19, 2023, Vienna, Austria, 9th International Conference on Artificial Intelligence and Applications (AIFU 2023), 12th International Conference on Embedded Systems and Applications (EMSA 2023), 4th International Conference on Natural Language Computing and AI (NLCAI 2023), 9th International Conference on Networks & Communication (NCOM 2023), 9th International Conference on Signal and Image Processing (SIPRO 2023), 10th International Conference on Software Engineering and Applications (SEA 2023), 11th International Conference on Data Mining & Knowledge Management Process (DKMP 2023), 4th International Conference on Big Data and Machine Learning (BDML 2023), 4th International Conference on Blockchain and Internet of Things (BIoT 2023) was collocated with 12th International Conference on Cloud Computing: Services and Architecture (CLOUD 2023). 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 CCSEA 2023, AIFU 2023, EMSA 2023, NLCAI 2023, NCOM 2023, SIPRO 2023, SEA 2023, DKMP 2023, BDML 2023, BIoT 2023, CLOUD 2023. 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, CCSEA 2023, AIFU 2023, EMSA 2023, NLCAI 2023, NCOM 2023, SIPRO 2023, SEA 2023, DKMP 2023, BDML 2023, BIoT 2023, CLOUD 2023 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 CCSEA 2023, AIFU 2023, EMSA 2023, NLCAI 2023, NCOM 2023, SIPRO 2023, SEA 2023, DKMP 2023, BDML 2023, BIoT 2023, CLOUD 2023.

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

Organization

David C. Wyld, Southeastern Louisiana University, USA
Dhinaharan Nagamalai (Eds) Wireilla Net Solutions, Australia

Program Committee Members

K L University, India

A. Raja Basha,
Abdel-Badeeh M. Salem, Ain Shams University, Egypt
Abdelhadi Assir, Hassan 1st University, Morocco
Abderrahmane ez-zahout, Mohammed V University, Morocco
Addisson Salazar, Universitat Politècnica de València, Spain
Afaq Ahmad, Sultan Qaboos University, Oman
Agnes Vathy-Fogarassy, University of Pannonia, Hungary
Aiyudubie Uyi, Air force Institute of Technology, Nigeria
Akhil Gupta, Lovely Professional University, India
Alexander Gelbukh, Instituto Politécnico Nacional, Mexico
Ali A. Shukur, University of Kufa, Iraq
Alia karim Abdul Hassan, University of Technology, Iraq
Alireza Valipour Baboli, University Technical and Vocational, Iran
Amal Azeroual, Mohammed V University, Morocco
Amanul Islam, University of Malaya, Malaysia
Ammar A. Aldair, University of Basrah, Iraq
Ana Luísa Varani Leal, University of Macau, China
Anouar Abtoy, Abdelmalek Essaadi University, Morocco
Aridj Mohamed Hassiba, Benbouali University, Algeria
Ashraf Darwish, Helwan University, Egypt
Ashraf Elnagar, Uinversity of Sharjah, UAE
Assem Abdel Hamied Moussa, Asdf-Srca Africa President, Egypt
Atul Garg, Chitkara University, India
Ayush Dogra, Ronin Institute, USA
Azeddine Wahbi, Hassan II University, Morocco
B Nandini, Telangana University, Nizamabad
Benyamin Ahmadnia, California State University, USA
Bhagyashree S R, ATME College of Engineering, India
Bhay Kumar Agarwal, Kamla Nehru Institute of Technology, India
Bibhu Dash, University of the Cumberlands, USA
Brahim Lejdel, University of El-Oued, Algeria
Bratin Ghosh, Indian Institute of Technology, India
Cheng Siong Chin, Newcastle University, Singapore
Christian Mancas, DATASIS ProSoft srl, Romania
Daniel Hunyadi, Lucian Blaga University of Sibiu, Romania
Dário Ferreira, University of Beira interior, Portugal
Dariusz Jacek Jakóbczak, Koszalin University of Technology, Poland
Debjani Chakraborty, Indian Institute of Technology, India
Ekbal Rashid, RTC Institute of Technology, India
Elżbieta Macioszek, Silesian University of Technology, Poland
F. Abbasi, Islamic Azad University, Iran
Felix J. Garcia Clemente, University of Murcia, Spain
Firmenich, Diego, UNPSJB, Argentina
Fzlollah Abbasi, Islamic Azad University, Iran

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Artificial Intelligence Community (AIC)

Soft Computing Community (SCC)

Digital Signal & Image Processing Community (DSIPC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

13th International Conference on Computer Science, Engineering and Applications (CCSEA 2023)

9th International Conference on Artificial Intelligence and Applications (AIFU 2023)

12th International Conference on Embedded Systems and Applications (EMSA 2023)

4th International Conference on Natural Language Computing and AI (NLCAI 2023)

9th International Conference on Networks & Communication (NCOM 2023)

9th International Conference on Signal and Image Processing (SIPRO 2023)

10th International Conference on Software Engineering and Applications (SEA 2023)

11th International Conference on Data Mining & Knowledge Management Process (DKMP 2023)

4th International Conference on Big Data and Machine Learning (BDML 2023)

4th International Conference on Blockchain and Internet of Things (BIoT 2023)

FUN WRITER: A CONTEXT-BASED INTELLIGENT WRITING PLATFORM TO ASSIST AND MOTIVATE WRITING ACTIVITIES USING ARTIFICIAL INTELLIGENCE AND NATURAL LANGUAGE PROCESSING

Jay Pang[1], Marisabel Chang[2]

[1]The Webb Schools, Front Entrance, 1175 W Baseline Rd,
Claremont, CA 91711
[2]Computer Science Department, California State Polytechnic University,
Pomona, CA 91768

ABSTRACT

The aim of this paper is to apply computer techniques to assist high school students' essay writing. A python program named Fun Writer is developed to assist users in paraphrasing, summarizing, and keywording. We applied it to real text sentences and paragraphs and conducted a qualitative evaluation of the approach. The results show that Fun Writer can automatically provide users with different versions of sentence paraphrases, Text or paragraph summaries, and keyword retrieval, thus being flexible to the user's needs. The program will be a website where the user can upload his own input text and freely select to either paraphrase, summarize, or extract keywords.

KEYWORDS

Writing, application, paraphrase, summary, keyword

1. INTRODUCTION

Essay writing can be difficult, especially if students need to solve problems with redundancy and understanding [1]. High school students particularly find it difficult to properly paraphrase or analyze texts when writing. What many end up doing is switching a couple of words for synonyms or swapping and flipping the sentence around to paraphrase, which often sounds unnatural or obvious that it is taken from elsewhere. They may summarize a paragraph or a text in a shorter or simple version but fail to keep the key ideas of the original. They even find analyzing other texts containing many complex subjects and terminology can seem impossible [2]. Therefore, we wanted to create an online tool to deal with this situation.

Currently, many writing assistance programs already exist, including Grammarly and google spell check, but these programs generally focus on the grammar portions of an essay [3][4]. However, the changes that these programs provide appear as suggestions which in many cases is not what the user is trying to find. Our goal was to create a python program where the user can choose exactly a specific part of a text they want analyzed and how [5]. There are many libraries that, if implemented correctly, can be purposed into analyzing the content of a sentence or

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD- 2023
pp. 01-06, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130501

paragraph. The three libraries we used for this program are Pegasus, Bart Transformer, and NLTK Rake for paraphrasing, summarizing, and finding keywords, respectively [6].

Our method took inspiration from prompt generators and programs such as Grammarly. Many useful features we noticed in Grammarly include the ability to spot redundancy within text and find paraphrased solutions to replace them. Unlike Grammarly which can be attached to a browser and be used in many websites, we decided to start off our program on its own website, which can be expanded upon in the future. Also unlike Grammarly, the user selects whether they want to paraphrase, summarize, or find keywords for the text.

To prove the effectiveness of the program, each function's results can be analyzed using different prompts. Paraphrasing solutions can be compared to the original text, checking if the subject remains consistent. Summary solutions are considered successful if they retain the same meaning but within much fewer words. Keyword solutions can be analyzed by checking whether they contain mainly subjects and actions rather than transition or conjunction words such as “because” or “and”.

The rest of the paper is organized as follows: Section 2 describes the challenges we faced throughout the development of this project. Section 3 explains our methods and solutions for our program and the challenges we met, while Section 4 presents two specific experiments we conducted to test our program. Section 5 compares our program to other text editing/analyzing programs. Section 6 contains our conclusion and plans for future development.

2. CHALLENGES

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

2.1. Finding a Way to Solve the Issues

The first challenge we faced was finding a way to solve the issue regarding essay writing beyond grammar or word spelling level. I spent a while observing how student essays can be improved and eventually settled on paraphrasing, summarizing, and finding keywords. Actions on these three aspects can not only help students to understand it better but also enhance the quality.

2.2. Paraphrased Sentences in Different Ways

The second challenge faced while developing this project was that sentences can be paraphrased in different ways and keywords can also differ depending on the context [7]. We noticed that the previous program could paraphrase sentences in ways the user may not prefer. To solve this issue, we designed the program to give multiple outputs for both paraphrasing and keywords. In the paraphrasing and keyword functions, several solutions are taken from the libraries and saved into variables, which are then displayed and chosen by the user themselves.

2.3. Unpredictability of the Libraries we Used

One more challenge we encountered was the unpredictability of the libraries we used. The outputs they created would often vary even with the same inputs, and there would often be mistakes regarding punctuation and other grammar issues as well. Therefore, we needed to test several different libraries for each action with many inputs until we found the 3 most consistent ones to use.

3. SOLUTION

Our program must recognize certain elements within the language in order to assist in writing. Coding each behavior for the program is impossible because of the number of rules in English, so as mentioned above, we implemented three different libraries for paraphrasing, summarizing, and finding keywords. Shown in Figure 1, The user first inputs the text they want to analyze into a text box, which is saved into a variable when the program is activated. We avoided making the program to save after every change in the text to allow the user to write long texts in the box without constantly running the program. The user can now give the second input, which is to select their desired function for the program. There are three buttons located beneath the text box labeled "Paraphrase", "Summarize", and "Keyword". The program will perform the corresponding function on the text within the box when a button is clicked. Each library can create multiple different solutions, which the algorithm will output in a column.

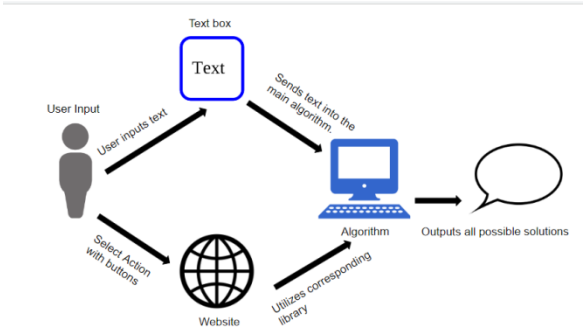


Figure 1. Project Layout

To implement this system, we imported all three libraries at the top of the program [8]. The program begins running after the user selects an action, in which the text within the text box are called into a function. The function outputs are saved within the variable "message" as seen above, which is returned and displayed on the website. The function for paraphrasing is defined above, while the summarizing and keyword actions are performed within this function.

4. EXPERIMENT

4.1. Experiment 1

The most effective way to experiment with this program is to simply test different inputs and compare them with the results. The Paraphrase function is designed to help students reword certain sentences, so the result maintains the same meaning and subjects as the original text. The Keywords function is meant to assist the user in understanding the text and to help them with summary writing, while the Summary function gives them an example of what it could look like. The keyword outputs should consist of mainly subject nouns and verbs from the text, as well as related words, but not function words such as "to", "the", or "for" etc. While the Summary outputs must be shorter than the original, it must also retain adequate amounts of important information. For the first experiment, we tested the program using the prompts as follows:

The input text: "A tractor is an engineering vehicle specifically designed to deliver a high tractive effort (or torque) at slow speeds, for the purposes of hauling a trailer or machinery such as that used in agriculture, mining or construction. Most commonly, the term is used to describe a farm

vehicle that provides the power and traction to mechanize agricultural tasks, especially (and originally) tillage, and now many more."(Segment with 2sentences, 66words)

The results of the three functions of Fun Writer are as follows:

Paraphrase: The last sentence is paraphrased into three versions:

The term refers to a farm vehicle that provides the power and traction to mechanize agricultural tasks.（17w）
The term is used to describe a farm vehicle that provides the power and traction to mechanize agricultural tasks.(19w)

The term is often used to describe a farm vehicle that provides the power and traction to mechanize agricultural tasks.(20w)

Keywords: engineering vehicle specifically designed', 'mechanize agricultural tasks', 'high tractive effort', 'farm vehicle', 'slow speeds', 'used', 'used', 'trailer', 'tractor', 'traction', 'torque', 'tillage', 'term', 'purposes', 'provides', 'power', 'originally', 'mining', 'many', 'machinery', 'hauling', 'especially', 'describe', 'deliver', 'construction', 'commonly', 'agriculture$^{5(27\ item)}$

'Summary: "A tractor is an engineering vehicle specifically designed to deliver a high tractive effort (or torque) at slow speeds. Most commonly, the term is used to describe a farm vehicle that provides the power and traction to mechanize agricultural tasks, especially (and originally) tillage. (Segment with 2 sentences but only 44 words)

4.2. Experiment 2

In the second experiment, we chose a random narrative segment rather than an informational text. After testing, we looked for the same conditions presented above. The input was:

"There were two things that were important to Tracey. The first was her dog. Anyone that had ever met Tracey knew how much she loved her dog. Most would say that she treated it as her child. The dog went everywhere with her, and it had been her best friend for the past five years. The second thing that was important to Tracey, however, would be a lot more surprising to most people" (segment with 6 sentences,73words in total)

The results of the three functions of Fun Writer are as follows:

Paraphrase: The last sentence is paraphrased as:

Most people would be surprised by the second thing that was important to Tracey.

Keyword: "'ever met tracey knew', 'past five years', 'dog went everywhere', 'two things', 'second thing', 'best friend', 'would say', 'tracey', 'tracey', 'dog', 'dog', 'would', 'treated', 'surprising', 'people', 'much', 'loved', 'lot', 'important', 'important', 'however', 'first', 'child', 'anyone'

Summary: "There were two things that were important to Tracey. The first was her dog. Most would say that she treated it as her child. The dog went everywhere with her and it had been her best friend for the past five years. The second thing that was important to her would be a lot more surprising."(Summary in 5 sentences, 56 words in total)

Results from both experiments show mostly accurate predictions. The paraphrase function works well at sentence level by providing 3 versions of paraphrasing for users to select, so that they can express the same idea in various grammatical and lexical ways. The summary function can shorten the original text but maintain its main ideas, so that users can understand or write about the topic more concentratedly. The Keywords function can display the content-type of key words that mainly consist of subject nouns and action verbs to help the user figure out the most useful information of the text or use them in writing.

5. RELATED WORK

Fitria, T. N. analyzes Quillbot, an online paraphrasing app that focuses on paraphrasing English texts [9]. Both Quillbot and our program use synonyms, passive/active voices, word forms, and word reorganization to paraphrase texts, and requires the user to input text into the website. However, unlike ours, Quillbot also performs grammar and sentence fixing functions that are not yet implemented in our program.

Litvak, M. et al. describes MUSEEC, a tool that uses three different methods of summarizing texts, MUSE, POLY, and WECOM, which creates multiple different summaries and compares them to find the best [10]. This is similar to the paraphrasing algorithm in our project, where we create a few solutions for users to choose.

Zhang, C. describes CRF, a method of finding keywords, and further explains the process of keyword extraction [11]. Unlike previous works, they focus on the use of functions and the processes behind keyword finding algorithms. Despite our use of libraries and functions for the project, I took note of many aspects used within these algorithms from this work.

6. CONCLUSIONS

In this paper, we report on how we use artificial intelligence, natural language processing and python coding to create a context-based intelligent writing platform to assist and motivate writing activities. The website platform utilizes the libraries Pegasus, Bart Transformer, and NTLK rake. The users can paste or type text into a text box on the website, then select the functions of paraphrase, summarize, or keywords by clicking three buttons below the box. To test the effectiveness, we gave several unique inputs varying in length and content density, and found that each function works well, as is illustrated in Experiment 1 and 2 in above.

One limitation of this program is that the accuracy is constantly varying, and there are many small grammar related issues within the library that occasionally appear. Another limitation is usage itself, as the website can be better optimized with more actions to create a better writing environment.

In the future, we can solve these issues by including with extra conditions and logic placed within the algorithm itself to check for mistakes in the solutions. The website can be reorganized into a more practical state with more precise and common actions such as grammar checks and passage tone.

REFERENCES

[1] Hounsell, Dai. "Essay writing and the quality of feedback." Student learning: Research in education and cognitive psychology (1987): 109-119.
[2] Maddela, Mounica, Fernando Alva-Manchego, and Wei Xu. "Controllable text simplification with explicit paraphrasing." arXiv preprint arXiv:2010.11004 (2020).
[3] Ghufron, M. Ali, and Fathia Rosyida. "The role of Grammarly in assessing English as a Foreign Language (EFL) writing." Lingua Cultura 12.4 (2018): 395-403.
[4] De Amorim, Renato Cordeiro, and Marcos Zampieri. "Effective spell checking methods using clustering algorithms." Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013. 2013.
[5] Sanner, Michel F. "Python: a programming language for software integration and development." J Mol Graph Model 17.1 (1999): 57-61.
[6] Deelman, Ewa, et al. "Pegasus, a workflow management system for science automation." Future Generation Computer Systems 46 (2015): 17-35.
[7] Bhagat, Rahul, and Eduard Hovy. "What is a paraphrase?." Computational Linguistics 39.3 (2013): 463-472.
[8] Robey, Daniel, Jeanne W. Ross, and Marie-Claude Boudreau. "Learning to implement enterprise systems: An exploratory study of the dialectics of change." Journal of management information systems 19.1 (2002): 17-46.
[9] Fitria, Tira Nur. "QuillBot as an online tool: Students' alternative in paraphrasing and rewriting of English writing." Englisia: Journal of Language, Education, and Humanities 9.1 (2021): 183-196.
[10] Litvak, Marina, et al. "Museec: A multilingual text summarization tool." Proceedings of ACL-2016 System Demonstrations. 2016.
[11] Bale, Tracy L., and Wylie W. Vale. "CRF and CRF receptors: role in stress responsivity and other behaviors." Annu. Rev. Pharmacol. Toxicol. 44 (2004): 525-557.

BUSINESS VALUE IMPACT OF AI-POWERED SERVICE OPERATIONS (AISERVICEOPS)

Harsha Vijayakumar

Researcher, S.P. Jain School of Global Management

ABSTRACT

Artificial Intelligence (AI) has been significant technology of the 21st century. This technology is changing every aspect of modern enterprise technology tooling, from strategies to selecting and implementing to adopting digital AI transformation. The rapid development of Artificial Intelligence has prompted many changes in the field of Information Technology (IT) Service Operations. IT Service Operations are driven by AI, i.e., AIServiceOps. AI has empowered new vitality and addressed many challenges in IT Service Operations. However, there is a literature gap on the Business Value Impact of Artificial intelligence (AI) Powered IT Service Operations. It can help IT build optimized business resilience by creating value in complex and ever-changing environments as product organizations move faster than IT can handle. So, this research paper examines how AIServiceOps creates business value and sustainability, basically how AIServiceOps makes the IT staff liberation from a low-level, repetitive workout and traditional IT practices for a continuously optimized process. One of the research objectives is to compare Traditional IT Service Operations with AIServiceOPs. This paper provides the basis for how enterprises can evaluate AIServiceOps and consider it a digital transformation tool.

KEYWORDS

AI-Powered Service Operations (AIServiceOps), Business Value Assessment, IT Service Management, IT Operations Management, and Digital Transformation.

1. INTRODUCTION

Digital transformation (accelerated due to COVID-19) impacts many changes across multiple industries. Part of this revolution is increasing IT spending on managing and maintaining IT Services to provide uninterrupted access to services from anywhere. Many organizations accelerated the adoption of digital technologies to build, deploy, monitor, remediate and manage IT Services; even mom-and-pop shop companies, e.g., Pet Clinics, have started websites (hosting services on the internet) to facilitate their customers' services. Such changes are known as “digital transformation” [23]. Another standard definition of digital transformation is as follows: a process that aims to improve an entity by triggering significant changes to its properties through a combination of information, computing, communication, and connectivity technologies [12][26][21].Many studies have shown that digital transformation increases businesses' ability to absorb effectively, adjust to situations and capitalize on surprises (e.g., COVID-19) that potentially threaten their existence. Digital technology will be, tomorrow, an increasingly crucial aspect of business resilience, with every company having to rely on data analytics, digital tools, and automation [7][21]. Manage and operate services—is the foundation of digital transformation—organizations have implemented IT solutions to build, deploy, monitor, remediate, serve and manage Services. So, to do that, customers need to purchase ITSM (IT Service Management) and ITOM (IT Operations Management) solutions to resolve incoming incidents or issues with their Services timely. With that being said, the emergence of new

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 07-17, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130502

technologies AI, ML, Blockchain, IoT, and many others, have influenced IT to automate, monitor, remediate, operate and manage IT Services. Artificial Intelligence is a simulation of human intelligence and automation capabilities in machines and software programs to think like humans and mimic human behavior. Artificial intelligence (AI) is changing a strategy setting off a wave of AI and automation in every digital transformation in all industry sectors. Artificial intelligence (AI) has been touted as a means for organizations to cut costs and enhance their quality of services, coordination, productivity, and practice efficiencies [4][14]. With the advancement of processing speed and communication bandwidth growth, hardware and storage cost has gone down; AI Transformation has become necessary for organizations to adapt in the new area as they have to stay ahead of the competition and risk managing disruption. There is a clear need for a deeper exploration of AI's impact on organizational activities, boundaries, and goals, including the mechanisms and processes involved in harnessing its power in digital transformation [2][13].

The future of IT Service Operations depends on the advancement of new technologies and intelligent machines. IT teams have to fight an uphill battle managing the massive amounts of data generated by modern IT systems. They are expected to handle more incidents than ever with shorter service-level agreements (SLAs) [16]. Technological advances open new possibilities and challenges for IT Service Operations. AIServiceOps combines Service Management and Operations Management to avoid IT silos. AIServiceOps platforms combine incident and event management into one platform utilizing advanced analytics technologies (AI/ML) to enhance IT Service Operations functions (operations and service desk) with proactive insights and recommendations. With the potential to target automation, potentially self-healing – applications, detecting anomalies, predicting outages, root cause analysis, and provide IT Agents with the capability to see across IT silos. AIOps (AI Operations) and AITSM (AI IT Service Management) have been typical industry terms and mainly focused on individual sections of AI narratives for operations and service management; slowly, platforms are moving towards unified records to avoid IT silos and drive end-to-end automation.

Business Value Impact (BVI) is the justification of an organization's investment in digital transformation. BVI mainly covers the justification with ROI (return on investment) and an in-depth explanation of investment benefits. Business Value Impact not just covers value creation; it covers preservation. AIServiceOps has effectively built value creation for an organization and has been an effective tool for building value as IT services have responded to ever-changing environments and have been able to deliver value with a focus on software delivery, agility, and experience. Consequently, BVI should be regarded as a strategic resource; enterprises are required not only to view business value as a cost to cope with short-term shocks but to promote the shaping of value-added activity in the long term. AIServiceOps is next-generation ITServiceOps that will not make organizations focus on short-term or daily operations but consider the long-term in creating an intelligent and automated digitized workplace and consider the AIServiceOps business sustainability. So, the research in this article aims to compare Traditional IT Service Ops and AIServiceOps benefits and treat AIServiceOps as a catalyst for digital transformation to increase business value. Various research tools will be applied to verify the objective above. This includes creative thinking techniques and semi-structured interviews with expert practitioners and organizations implementing or has adopted AIServiceOps solutions.

2. THEORETICAL BACKGROUND

2.1. Role of Digital Transformation in Creating Business Value

Organizations are vulnerable to many external factors that may impact their business, e.g., COVID-19 has influenced how organizations run their businesses as many companies started focusing on e-commerce than regular retail stores. External factors like these can make the organization's work unproductive both in terms of value creation and value preservation. Hence assessing the business value of digital transformation is very important. Assessment of business value and resilience allows organizations to withstand any impact from external or internal factors or events. Moreover, resilience can be formed consciously, i.e., it can be strengthened or weakened. Such actions change the level of resilience gradually [1][21]. Concurrently, “resilience” is a contextual term and needs to specify what it is in a given context it refers to. This requires answering questions such as “whose?” and “of what's?” resilience we mean; “what?” is the current and, perhaps also, the desired level of resilience; and “against what” type of event is this resilience supposed to be working, or against “what?" event do we want to be resilient [10][21]. “Organization Resilience” is the ability of an organization to create value and preserve the value even when encountering by surprises, this perception of organizational resilience is in line with the dynamic capability theory [21][25], which has been explored very intensely in recent years and explains how companies respond to rapid changes in technology and markets [21][24]. Organizations should start or have started considering digital transformation as a way to enable organizational resilience and create business value; digital transformation is an effective method for enterprises to avoid risks and facilitates the enterprise's ability to comprehend and adapt to changing environmental contexts [21][27].

On the other hand, business value helps organizations understand digital transformation's benefits, costs, and risks. Every organization should evaluate the potential financial impact of digital transformation on their organization. Organizations should perceive digital transformation as creating more value and preserving it. At times it should avoid risks – for example, IT Support teams used to be unaware of service issues, and now with IT tools like service operations management, IT teams can detect anomalies and correct issues 15-20 minutes before it occurs. Organizations need to ask below questions to assess the business value of digital transformation:

- How will digital transformation improve IT and employee productivity?
- What is the impact ofreducing high-priority incidents from digital transformation and, in turn, impacts customer experience?
- What are the impact of digital transformation on project performance and accelerated returns?
- What are the reduced and avoided costs of digital transformation? E.g., payback period?

2.2. IT Service Operation is Driven by Digital Transformation and Artificial Intelligence

Digitization means fundamental changes in how business operations and enterprises' business models are implemented and introduced, thanks to digital technologies and data that are both digitized and natively digital [20][21]. Digital transformation (accelerated due to COVID-19 – an external environment disruptor) impacts many changes across multiple industries. Part of this revolution is increasing IT spending on managing and maintaining IT Services to provide uninterrupted access to services from anywhere, e.g., before covid, employees used to sit in offices (inside the company's infrastructure) to access services, and now IT has to extend VPN(a virtual private network) so that employees can access those IT services.

Furthermore, a business model is a conceptual tool that contains a set of elements and relations that enables the business logic of a given company to be expressed; it includes a description of the value offered by a company to a group or groups of buyers, a description of the enterprise's architecture and a list of its network of partners who co-create, offer, and deliver this value and relational capital, ensuring continuous revenues conducive to profitability [18][21]. Business Value Impact will show how organizations continue to have both quantifiable and flexible value. Quantifiable Value is a value that can be quantified, and Flexible value cannot be quantified but are significant benefits to organizations, e.g., improved productivity, reduced priority incidents, project performance, and avoided costs are examples of quantifiable value and scale without increasing headcount, single platform for many business applications are examples of the flexible value.

Technological changes currently taking place in the market encourage companies to experiment with how new IT solutions will affect their business models, and—based on research conducted among nearly 340 European enterprises, it can very clearly be seen that such an impact exists [3][21]. So from the perspective of technological change, AIServiceOps(AI-Powered Service Operations) is a next-generation ITServiceOps (IT Service and Operations Management) that will not make organizations focus on short-term or daily operations but consider long-term in creating an intelligent and automated digitized workplace and consider the AIServiceOps in creating business value, resilience and sustainability. So, the research in this article aims to compare Traditional IT Service Ops and AIServiceOps benefits, treat AIServiceOps as an instrument for digital transformation, and study its impact on business value and resilience. Various research tools will be applied to verify the objective above.

2.3. AIServiceOps in the Context of Business Value Impact

According to IDC, 95% of organizations report they are implementing a digital-first strategy to support new digital revenue streams. By 2027, the average enterprise will see 41% of its revenue come from digital products and services [17]. Even if organizations were already on a digital-first journey before the pandemic, you know the unnecessary friction and increased costs that come from having hundreds of disparate technology tools across different teams when services and operations teams, tools, and data are siloed – IT services teams are not able to meet the increase in the number of requests from employees resulting in poor experiences and the IT operations teams are not able to predict and prevent service outages resulting in lost productivity and revenue. Both IT Services and operations feel pain from the siloed organization. Also, the key challenge with siloed solutions is lack of agility, difficulty in maintenance, and significant room for improvement in terms of automating processes For Example:

On IT Services:

- Employees are frustrated with IT supportand poor employee experiences due to the burden of routine, repetitive requests to IT staff.
- Unmet decentralized tech needs: Tech teams forming in the business with little to no consistency in managing tech vendors and systems.
- Poor IT productivity: Disparate data and lack of insight into service delivery hinder service and cost improvement.

On IT Operations:

- Significant delays in resolving high-priority incidents: To diagnose and resolve high-priority incidents/ service outages, Ops needs historical, real-time change and incident data.
- Spotty service availability: Erratic service availability, project reliability, and manual escalation process.

IT silos slow innovation, high cost of services, and a slower rate of innovation' from DevOps and IT teams operating in silos as teams will be busy fixing incidents with disparate data and insights. And in reality, both of these teams also feel the other's pain. Achieving digital-first business growth starts by bringing your technology services and operations together, by bringing them together:

- Expand technology services while reducing costs
- deliver extraordinary employee experience, customer experience, and resiliency and drive technology best practices and optimized processes.

The first step is to modernize service and operations on a single platform to help expand and improve the services. Modernize - Stand up the foundation to:

- Gain visibility of services and their dependencies
- Integrate the service experiences with Incident and Event Management
- Integrate the data from across the enterprise

The second step is automating service and operations using AI to empower self-service and predict and prevent incidents before they impact users or your business. AIServiceOps enables: Automate services to:

- Provide employees self-service – identify automation to build playbooks to reduce MTTR (mean time to repair)
- Virtual Agent and AI Search for helping employees to self-help and system to self-heal with auto remediations
- Predictive Intelligence on major incidents with effective AI/ML technology to correlate the historical and create benchmarks

Incident prevention for systems and applications:

- Predict and prevent issues
- Pinpoint root cause using AI/ML
- Reduce noise
- Correlate alerts
- Run automation playbooks

Furthermore, enterprises that have roots in traditional industries can improve the quality of their customer experience, change the company's revenue structure, and transform their distribution channels by introducing digitization into their business models [21][22]. AIServicOps is one of the keyways for the digital transformation of organizations, and its impacts can be viewed by IT/DevOps and its customers, creating a seamless value chain for Organizations.

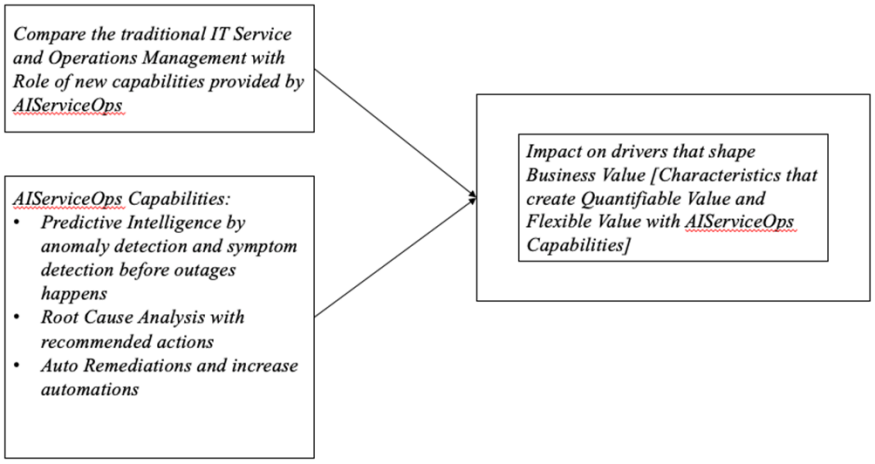


Figure 1: AIServiceOps capabilities influence on Business Value

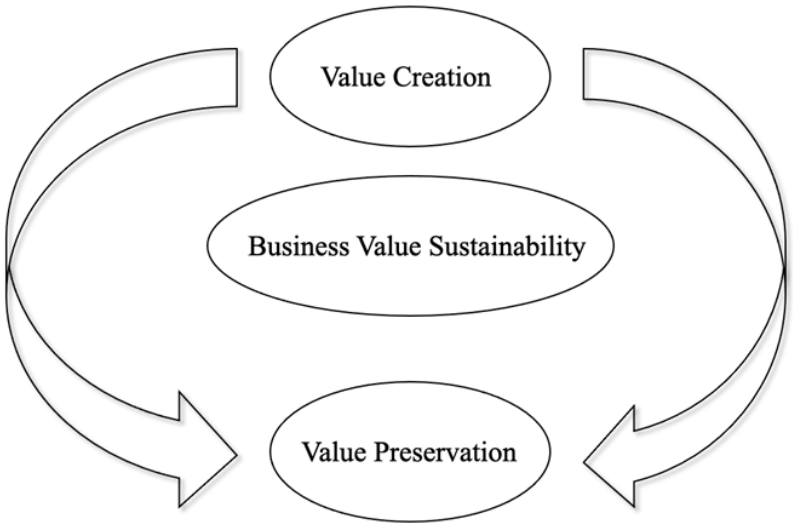


Figure 2: Cycle of Business Value Sustainability

Sustainability is a driver of business value, and value has to be created and preserved for long-term organizational goals.

2.4. Conceptual Model

A conceptual model is a framework initially used in research to outline the possible courses of action or present an idea or thought [6]. When a conceptual model is developed logically, it will rigor the research process. A model is a representational illustration and a heuristic device visually portraying concepts and theory [6]. Models provide a common understanding to the viewers about the knowledge by showing various elements of a system and their interrelationship [8], defining a conceptual model as a diagram of proposed causal linkages among a set of concepts. They highlight that a conceptual model provides a visual picture representing concepts through boxes and processes delineated by arrows. Thus, a model developed using the standard conventions can clearly define the causal, sequential, and logical argument that creates a clear and shared understanding by the habit of mind. So below is the conceptual model that illustrates how the theory of Business Value Impact of AI-Powered Service Operations. The model is

described with various factors of AIServiceOps capabilities impacting the aspects of Business Value.

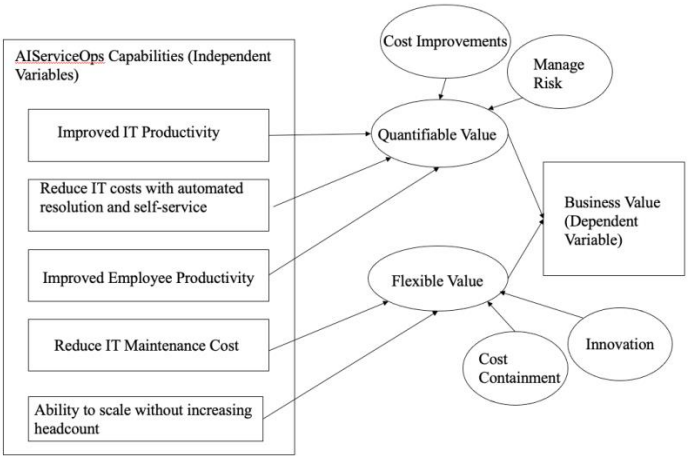


Figure 3: Business Value framework for adopting AIServiceOps

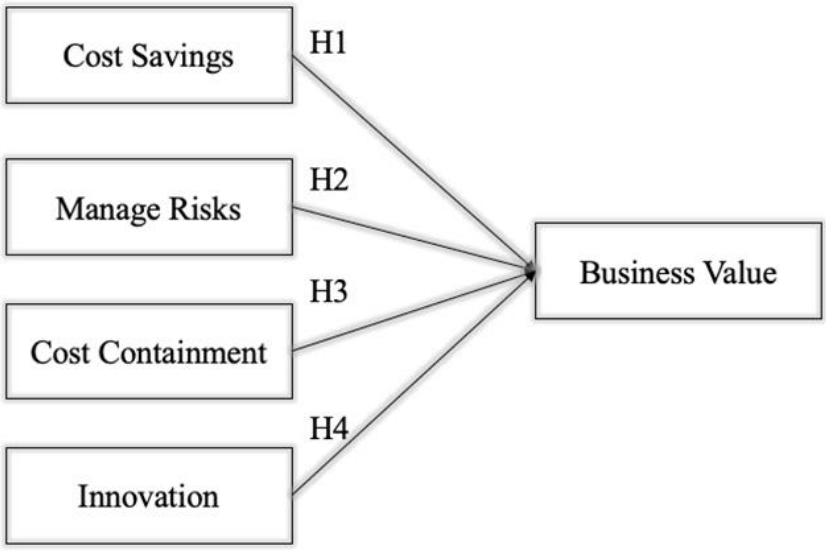


Figure 4: Conceptual Model for Business Value impact of AIServiceOps

There is one dependent variable – Business Value as the outcome measures and benefits for actual adoption AIServiceOps Capabilities with meditation variables – Value Creation and Value Preservation and moderating variables supporting it are – Manage Risk, Cost Containment, Innovativeness and Cost Improvements.

Below are the hypotheses that we need to prove with this research:

- H1: With the adoption of AIServiceOps, cost containment significantly influences the creation of business value
- H2: With the adoption of AIServiceOps, cost improvement significantly influences the creation of business value

- H3: With the adoption of AIServiceOps, innovativeness significantly influences the creation of business value
- H4: With the adoption of AIServiceOps, managing risk significantly influences the creation of business value

3. QUANTITATIVE RESEARCH METHODOLOGY

This research will adopt a quantitative research methodology, focusing on analyzing how AIServiceOps has impacted factors that, in turn, impact business value.

3.1. Data Collection

The research will be regarded as the state of AIServiceOps usage in enterprises and its impact on business value. The study will be comprised of questionnaires to participants and a few interviews. The survey questionnaire was asked to 70 organizations mainly as a paper form in workshops as it reduced the cost of study and gave the right contextual quality and complete answers.

3.2. Questionnaire Description

This research is intended to explore and examine the AIServiceOps factors that influence Business Value. All aspects and questions have been created from existing literature related to the topic and expertise in the field. Some questions determine the overall thinking and outcome of AIServiceOps. Not all questions in the research are related to factors influencing business value, as these are related to scope, preconditions, and implementation associated with AIServiceOps. Some are focused on disruptors that can impact Organizational stability. Survey questionnaires using the Likert Scale of 5 points, designed by Rensis Likert; this is a prevalent rating scale for measuring ordinal data in social science research. This scale includes Likert items that are worded statements to which respondents can indicate their extent of agreement or disagreement on a five or seven-point scale ranging from “strongly disagree” to “strongly agree” 5-point scale will be “1= Strongly disagree,” “2 = Rather disagree,” “3 = Hard to say,” “4 = Rather agree,” and “5 = Strongly Agree”. Survey questions that relate to factors:-

Cost Containment [Value Preservation]

- AIServiceOps has reduced effort on IT maintenance.
- AIServiceOps has avoided development costs and avoided legacy costs.
- AIServiceOps has improved IT Organizations' ability to scale without increasing headcount.

Cost Improvement [Value Creation and Value Preservation]

- AIServiceOps investment improves digital outcomes more than traditional IT Service Operations Management
- Cost per incident/ticket has decreased with adopting AIServiceOps and has “bend the cost curve” for Organizations.
- Agent utilization has improved with the adoption of AIServiceOps.

Innovativeness [Value Preservation]

- AIServiceOps has made your centralized IT team more productive by focusing on critical tasks and incidents, liberated IT staff from low-level, repetitive work, and played a vital role in your organization's digital transformation.
- AIServiceOps has helped you to find automation opportunities and prioritize to potentially deflect and lower the mean time to resolve (MTTR).
- There is a clear gap in value provided by AIServiceOps compared to traditional IT Service Operations Management.

Manage Risk [Value Creation]

- AIServiceOps has improved identifying anomalies 10-15 minutes before the actual incidents or outage of service occur.
- AIServiceOps has improved the quality of managing services, and it provides organizations with end-to-end visibility, focuses on incidents that matter, and fixes them more quickly.
- AIServiceOps has reduced incident volumes and repair time by providing tools and the proper context for the investigation.

Business Value Sustainability [Value Creation and Value Preservation]

- Implementing AIServiceOps has an impact on the value creation for your organization.
- Implementing AIServiceOps has an impact on the value preservations for your organization.

Other interview questions

- What conditions before implementing the AIServiceOps?
- How has AIServiceOps changed your traditional IT practices regarding risk, cost, customer satisfaction, and innovations?
- What are the critical KPIs to measure the success of AIServiceOps?
- What is the current scope of AIServiceOps?
- What are the key disruptors that have impacted IT lately, and what was the strategy to overcome that?
- What would be the impact of AIServiceOps on the business value and resilience of the enterprise?

4. CONTRIBUTION TO PRACTICE

Using AIServiceOps in organizations solves some of the current issues, such as investigating root-cause-analysis (RCA) of incidents and preventing outages of services and visibility into the entire organization's infrastructure estate. It also provides many automation opportunities and innovativeness for IT teams to automate and create value for organizations. Adopting AIServiceOps helps bring efficiencies in IT service and operations management by avoiding IT silos and increasing customer satisfaction. AIServiceOps usage helps transform IT service and operations management by managing risk, increasing customer satisfaction and cost improvements, increasing cost containment, increasing innovativeness, and improving overall organizational resilience.

5. LIMITATIONS AND SCOPE OF FUTURE RESEARCH

Due to lack of time, this study did not expand its scope of impact of AIServiceOps on a specific industry; this research focused on general IT-specific functions related to service and operations management. Also, the existing analysis did not account for disruptors that would impact business value. Understanding the characteristics of each organization's external and internal disruptors is necessary. Furthermore, this research did not focus on the impact of AIServiceOps on value optimization in the value creation and preservation process to maximize the effect of AIServiceOps on business value. The focus of this research paper was to establish the positive impact of AIServiceOps on business value via quantitative research methodology.

6. CONCLUSION

Organizations need to consider AIServicOps solutions as it impacts customer satisfaction, managing the risk of outages and cost savings via deflecting incidents and liberating IT from mundane, repetitive tasks with automated remediations. AIServiceOps enables IT to take on a more customer-centric approach in event and ticket management to improve IT productivity, customer satisfaction, and business well-being. Also, AIServiceOps lets IT teams reduce MTTR across all phases of the incident management process.

REFERENCES

[1] Annarelli, A., Battistella, C., & Nonino, F. (2020). A framework to evaluate the effects of organizational resilience on service quality. Sustainability, 12(3), 958.
[2] Aldrich, H. (1999). Organizations evolving. Sage.
[3] Bouwman, H., Nikou, S., Molina-Castillo, F. J., & de Reuver, M. (2018). The impact of digitalization on business models. Digital Policy, Regulation and Governance, 20(2), 105-124.
[4] Davenport, T. H. (2018). The AI Advantage: How to put the artificial intelligence revolution to work. mit Press.
[5] Denyer, D. (2017). Organizational resilience. UK: BSI and Cranfield University.
[6] Elangovan, N., & Rajendran, R. (2015). Conceptual model: A framework for institutionalizing the vigor in business research. In Proceedings of Third National Conference on Indian Business Management. Coimbatore: Sri Ramakrishna Institute of Technology (pp. 1-32).
[7] Elgazzar, Y., El-Shahawy, R., & Senousy, Y. (2022). The Role of Digital Transformation in Enhancing Business Resilience with Pandemic of COVID-19. In Digital Transformation Technology (pp. 323-333). Springer, Singapore.
[8] Earp, J. A., & Ennett, S. T. (1991). Conceptual models for health education research and practice. Health education research, 6(2), 163-171.
[9] Foss, N. J., & Saebi, T. (2017). Fifteen years of research on business model innovation: How far have we come, and where should we go?. Journal of Management, 43(1), 200-227
[10] Fraccascia, L., Giannoccaro, I., & Albino, V. (2018). Resilience of complex systems: State of the art and directions for future research. Complexity, 2018.
[11] Gefen, D., Straub, D., & Boudreau, M. C. (2000). Structural equation modeling and regression: Guidelines for research practice. Communications of the association for information systems, 4(1), 7.
[12] Hinings, B., Gegenhuber, T., & Greenwood, R. (2018). Digital innovation and transformation: An institutional perspective. Information and Organization, 28(1), 52-61.
[13] Holmström, J. (2022). From AI to digital transformation: The AI readiness framework. Business Horizons, 65(3), 329-339.
[14] Iansiti, M., & Lakhani, K. R. (2020). Competing in the age of AI: strategy and leadership when algorithms and networks run the world. Harvard Business Press.
[15] Mao, H., Zhang, T., & Tang, Q. (2021). Research Framework for Determining How Artificial Intelligence Enables Information Technology Service Management for Business Model Resilience. Sustainability, 13(20), 11496.

[16] Masood, A., Hashmi, A. (2019). AIOps: Predictive Analytics & Machine Learning in Operations. In: Cognitive Computing Recipes. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-4106-6_7 [Original source: https://studycrumb.com/alphabetizer]
[17] Meredith Whalen - Chief Research Officer. (2022, August 3). What will a digital-first world look like in the future? IDC Blog. Retrieved August 5, 2022, from https://blogs.idc.com/2022/05/18/what-will-a-digital-first-world-look-like-in-the-future/
[18] Osterwalder, A., Pigneur, Y., & Tucci, C. L. (2005). Clarifying business models: Origins, present, and future of the concept. Communications of the association for Information Systems, 16(1), 1.
[19] Sarstedt, M., Ringle, C. M., & Hair, J. F. (2021). Partial least squares structural equation modeling. In Handbook of market research (pp. 587-632). Cham: Springer International Publishing.
[20] Schallmo, A., & Daniel, R. (2018). Digital Transformation Now! Guiding the Successful Digitalization of YourBusiness Model. Springer Science+ Business Media, LLC.
[21] Sobczak, A. (2022). Robotic Process Automation as a Digital Transformation Tool for Increasing Organizational Resilience in Polish Enterprises. Sustainability, 14(3), 1333
[22] Sundaram, R., Sharma, D., & Shakya, D. (2020). Digital transformation of business models: A systematic review of the impact on revenue and supply chain. International Journal of Management, 11(5)
[23] Suryono, R. R., Budi, I., & Purwandari, B. (2020). Challenges and trends of financial technology (Fintech): a systematic literature review. Information, 11(12), 590.
[24] Tan, B. C., Pan, S. L., & Hackney, R. (2009). The strategic implications of web technologies: A process model of how web technologies enhance organizational performance. IEEE Transactions on Engineering Management, 57(2), 181-197
[25] Teece, D. J., Pisano, G., & Shuen, A. (1997). Dynamic capabilities and strategic management. Strategic management journal, 18(7), 509-533
[26] Vial, G. (2021). Understanding digital transformation: A review and a research agenda. Managing Digital Transformation, 13-66.
[27] Zhang, J., Long, J., & von Schaewen, A. M. E. (2021). How does digital transformation improve organizational resilience?—findings from PLS-SEM and fsQCA. Sustainability, 13(20), 1148

AUTHOR

Harsha Vijayakumar, Research scholar at S.P. Jain School of GlobalManagement, having more than 12+ years of experience in Product Management and Software Engineering. Currently working as a Sr Principal Product Manager at ServiceNow, Inc. (California, USA)
Personal Website: https://www.linkedin.com/in/harshavijayakumar1/
ORCID:0000-0002-2833-9159
Alternative Email: harsha.dm22dba013@spjain.org

AN INTERACTIVE AND COLLABORATIVE GAMING PLATFORM TO ENGAGE THE AUTISM SPECTRUM IN ART LEARNING USING ARTIFICIAL INTELLIGENCE

Carina Zheng[1], Yu Sun[2] and Yujia Zhang[3]

[1]Orange County School of The Arts, 1010 Main Street N, Stata Ana, 92701
[2]California State Polytechnic University, Pomona, CA, 91768,
Irvine, CA 92620
[3]University of California Irvine, Irvine, CA 92697

ABSTRACT

For decades, mental illness has been a popular topic of discussion that still lingers for effective treatments [1]. While current therapy of mental disorders can achieve success, it is far from enough to prevent their occurrence and impacts on individuals [2]. Because of this, mental illness is an area of study that requires professionals and specialists to take a further step. Additionally, as the use of technology advances in the current society, young children and pre-teens gradually become victims of mental disorders as well: a community that often needs careful attention from adults and caregivers [3]. This paper introduces a method of treating mental disorders in young individuals that is not considered rare, but often overlooked, by many. This application encourages creativity and interests in its users, motivating them to actively engage on their strengths and use it to reflect their struggles.

KEYWORDS

3D Modeling, Unity, Collaborative Gaming Platform

1. INTRODUCTION

Psychology, the study of human mind and behavior, is on the rise to understand the relationship between human emotions and their leading consequences [4]. As our society approaches advancement with developing technology, the romantic side of humans seems to be degraded and bothered. Looking at scientific research, the number of people who suffer from mental disorders is endlessly increasing, While mental illness among teenagers and adults is common and vital, these similar struggles aren't any less significant in the even younger age group. Scientific studies have shown that childhood trauma is one of the main causes for mental illness that follows into adulthood if not discovered and treated by the right time [5]. These mental hardship forces much more pressure on a child than what they are able to manage. Childhood trauma can push individuals to acquire low self-esteem, pessimism, and fanaticism in many aspects of life [6]. These factors can slowly integrate with other struggles that one faces into larger pressure and fear, affecting their lives. Fortunately, the psychology of children's minds is not ignored among scientific scholars, many study the neuroscience behind the topic and conclude their findings. Yet, effective solutions to childhood mental illness are not widely populated as many parents continue to turn to therapy for their children, believing that they will receive fruitful results from it.

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 19-28, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130503

Over the last few decades, some medical and psychological techniques have been proposed to parents and children that allow better connections within families, such as the evidence-based treatment (EBTs). There are various types of children therapy that are applied as a treatment to encourage interactions and healing, like the Trauma-Focused Cognitive Behavioral Therapy(TF-CBT) [7]. The TF-CBT aims to heal children who have experienced sexual, physical, and mental abuse by addressing distorted, pessimistic thinkings in children and incorporating family therapy [8]. Its core treatments include psychoeducation, coping skills, gradual exposure to the child's traumatic memory, cognitive processing, and caregiver involvement [9]. All these features of the TF-CBT are used to develop a sense of safety and confidence in the patients. However, these implementations may not be effective to all. Oftentimes, trauma experiencers tend to buried their horrifying past and memories for them to not be reflected on, which is the main reason that can thrawt the the TF-CBT treatment. Additionally, some victim may not be effective to all. Oftentimes, trauma experiencers tend to buried their horrifying past an struggle with financial and educational resources, preventing them to ask for the help they need.

In this paper, we will address our solution to the previously mentioned struggles among young children, and answer the question regarding the scientific prove behind our method. [transition] Before we help, it is important to know how. Several readings that we had done prior to building this program informed us that young children can easily be misunderstood and pressured by others; therefore, finding a relaxing, suitable way of communication is a crucial factor that determines the efficacy of the treatment. Our goal is to provide children a place where they can express themselves freely, without judgement; this aspiration leads us to develop the program, The Drawing Plan.

There are some useful features of The Drawing Plan. First, hearing the name of the app, it is a drawing app. The Drawing Plan is a simple drawing app designed for young children. We avoided having numerous features and functions in the app to "enrich" our user experience and instead stuck with the fundamental drawing functions with only five different types of drawing utensils. This is to ensure that a child can learn how to use the app quickly and without an adult supervision/help, granting them more time and space to work alone. Second, The Drawing Plan provides drawing prompts, randomly and custom-designed. Every drawing prompt is carefully considered to ensure it has its own value in helping the child reveal their struggles. The custom-designed prompt is a function for the caregiver or guardian of a child who desires dedication on a topic, which helps the users to dig deeper into the problem if they have already discovered some information about the child. Third, the user-experience designs of The Drawing Plan are all original creations. We first began the work on paper, sketching for icons and laying details out, then scanning the hand-drawn designs to the computer for further edit, and finally importing our designs into the app to create the finished look. Therefore, we believe that The Drawing Plan will be able to assist both adults and childrens in healing any sorts of childhood trauma and illness.

In two application scenarios, we demonstrate how the above combination of techniques can provide proper help to our users. First, before we begin promoting and launching the application, we run an experiment ensuring the appropriate functions of the application: what does each button do, does the application give immediate response when a button is pressed, is there anything that malfunctions? Second, we introduce this application to audiences as a real-life experiment, analyzing the evolution of an user's thought as they use the application. This includes hosting in-person workshops to engage with young children and physically experiencing the atmosphere of a classroom. The application scenario is the main experiment of our program that relates to child psychology.

One other scientific proof of the efficacy of The Drawing Plan comes from the reading of the book What Happened To You: Conversations on Trauma, Resilience, and Healing. This book

discusses the perspective of individuals whose lives are heavily impacted by trauma, and the results of experiencing childhood trauma and the process of healing from mental disability.

The rest of the paper is organized in an order that will help readers better understand the rhythm and usage of my app. Section Two will give specific details on the challenges that we faced while creating the app. Section Three focuses on the solutions to the challenges addressed in Section Two and an outline of the program. Section Four gives detailed information on the real-life experiments and community services done regarding this application and its related topic. Following that, a series of articles and books written by professional scientists are summarized and compared to the concept of The Drawing Plan. Finally, Section Six addresses the application's potential future updates and corrections.

2. CHALLENGES

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

2.1. Determine a Topic

Being able to determine what program to build was the first challenge that I encountered. Initially, many paths and ideas were laid in front: creating a color mixer for artists, an automatic phone number recorder, a drawing game. All well-inspired and absorbing. Yet, choosing the topic that has practicality, not too difficult to accomplish, and does good to the society was difficult. I carefully evaluated the necessity and functionality of each concept with many things in mind, such as the intended audience of a concept, the difficulties to reach my end goal—a finished program, and whether or not this program will be useful.

In the end, out of all the other ideas that fascinated me, I chose to design a program for children with special needs and/or mental disability. The reason for this is that I'd like to help those who don't often get enough attention and support from others, and are struggling with their daily life at the same time. Those who are thriving to overcome their struggles, parents or children, need their excruciating pain to be eased; but sometimes this is overlooked by others. Therefore, I had decided that making an app that aids this group of population will help the society towards a greater sense of happiness.

2.2. Designing the Visual Theme

The second challenge that I faced was to create the right functions and designs for children. This may seem like it has an easy solution: use cartoons and children-friendly designs. However, with my program's purpose of helping special-needs children, the usage and visuals of my app have to go beyond ingenious. Because of this, I conducted a vast amount of background readings and research about children. I read books and articles about different types of children's psychological struggles, the impact of childhood trauma on a person, children's psychological needs when they suffer from mental illness, and the process of healing. Additionally, it is known that young children don't have complete knowledge of revealing their thoughts to others with words, therefore I dedicated myself to developing a methodology that children favor. The application must encourage the user to be regulated and relaxed, it must not create tension in any way, because negative emotions affect what a child's creative mind can produce.

2.3. Programming Difficulties

The third challenge that I had to embrace was the constant learning of unfamiliar codes and programming stages. In order to ensure users have a fine experience when using this app, my team had to research suitable programming functions and Unity's, what we used to build the app, features and settings for what we needed to establish. For example, it was a challenge when we tried to import brush textures into my program to create a variety of digital drawing utensils for users. We worked with numerous different line codes and explored Unity's own app settings to try to overcome the struggle. Many attempts took place, but most did not reach what we hoped for.

3. SOLUTION

Component/Platform 1 - Free Draw

This application is a child-psychological help program that allows young users to convey their stories and their supervisors to evaluate necessary actions to be taken based on the child's drawing. To ensure whether the quality of this application meets its intended mission, there are three characteristics that I focused on: the clarity of the application's job, the artistic theme, and the simplicity of each branch inside the application.

With the intended users of this application, children and caregivers, the application will provide users with three platforms: the Free-Draw platform, Get-a-Prompt platform, and Make-a-Propmt platform. Each platform has its specially-designed options to satisfy the users' needs. When entered into the app, all users will see a "Start" page with the application's title and a button to begin using the application. The user will then be led to the Main Menu page, and there, they are presented with the three platforms discussed earlier. Each of these platforms is implemented with its own purpose while sharing the same major functionality, drawing. Below are three visualizations of the the platforms.

Figure 1, shown below, is the page upon clicking on the Free Draw button in the Main Menu. Here, a window will appear in the center of the screen with the prompt, "Free Draw". Below the prompt, there is the "Back" button and "Save" button; in the top right corner, there is the "Exit" button to close the window .

On the drawing page, the user is provided with 14 color swatches, including shades of the primary colors, secondary colors, and black, gray, and white. There are also five different types of brushes, including the acrylic paint, crayon, marker, pencil, and watercolor textures. Among the five textures, the watercolor brush performs one special technique: blending. The transparency of the watercolor brush is a perfect tool for mixing colors and creating a variety of shades and values, further enhancing the user's work of art. The user is able to switch between any of these features to maximize their creativity.

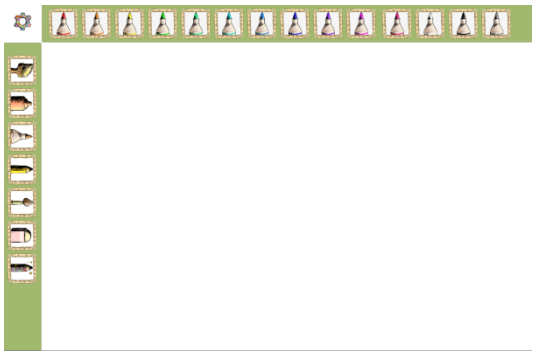


Figure 1. Buttons in the main menu

Figure 2 below shows the segment of code that establishes each brush mark made on the drawing panel. First, we call the system to instantiate, or create, a “brush”. We define the width and length of the brush by looking at where the brush mark starts and where it ends, setting a specific point position for them. Then, we ensure that the color of the brush mark remains the same for one complete stroke. After, the line “layerDistance -= 0.1f” creates a vertical distance between each brush mark, making all of them in separate layers to prevent issues with some brush strokes overlapping with others incorrectly. This, however, will visually appear to the user on screen, every brush mark will be flat.

```
void CreateBrush()
{
    GameObject brushInstance = Instantiate(brush, gameObject.transform);
    currentLineRenderer = brushInstance.GetComponent<LineRenderer>();

    currentLineRenderer.startWidth = brushWidth;
    currentLineRenderer.endWidth = brushWidth;

    currentLineRenderer.startColor = primaryColor;
    currentLineRenderer.endColor = primaryColor;


    currentLineRenderer.material.SetColor("_Color", primaryColor);

    layerDistance -= 0.1f;

    Vector2 mousePos = m_camera.ScreenToWorldPoint(Input.mousePosition);
    Vector3 pos = new Vector3(mousePos.x, mousePos.y, layerDistance);

    currentLineRenderer.SetPosition(0, pos);
    currentLineRenderer.SetPosition(1, pos);


    //currentLineRenderer.material.SetTextureScale("_MainText", new Vector2(100,1));
    //Debug.Log(currentLineRenderer.material.GetTextureScale("_MainTex"));
}
```

Figure 2. Create brush code

Component / Platform 2 - Get a Prompt

The major outline of the Get-A-Prompt platform is similar to the Free-Draw platform. How this platform differs from Free Draw is that this time, it generates a randomly selected prompt for the user. For instance, the prompts can say, “What does your dream birthday party look like?”, or “What do you do when you are home alone?”. The user then follows the guidance of the prompt to create and visualize their thoughts. After they finish their drawing, they can click on the “Settings” icon located on the top left of the screen to save their drawing. Figure Three is a segment of code that shows the implementation of the random prompt function.

```
public class AppState : MonoBehaviour
{

    public static string[] promptsList = {
        "What does your family vacation look like?",
        "What do you do when they are home alone?",
        "It's lunchtime at school!",
        "Oh no! You forgot to do their homework!",
        "Draw your dinner!",
        "What do you look like in the future?",
        "Your dream birthday party!",
        "Draw you in your favorite clothes.",
        "Draw yourself!",
        "Your least favorite food?",
        "A person and a hand!"
    };

    public static string prompt = string.Empty;


    public static void SetRandomPrompt()
    {
        int count = promptsList.Length;
        int index = Random.Range(0, count);
        prompt = promptsList[index];
    }
```

Figure 3. AppState code

Component / Platform 3 - Make-A-Prompt

The Make-A-Prompt platform, presented in Figure 4 below, is designed for the adult user and/or caregiver of the child. The purpose of this platform is to give adult users the opportunity to seek for what they need from the child. This platform is useful when the guardian of the child has already found out some characteristics of the child, and they want to continue their exploration. Here, the user is allowed to craft a prompt of their choice to guide the young child more intentionally. After the user has entered in a prompt, they can click on the "Next" button, the application will again take them to the drawing panel, with a window in the center stating the prompt that they've entered.

Figure 4. Screenshot of feature

4. EXPERIMENT

4.1. Experiment 1

The three platforms designed are strong indications of the purpose and theme of this application: to perform psychological help to a child in need. The application provides indirect aid through the method of drawing to ensure mental regulation of the young users. Design experiment #1 focused on testing the application itself and its proper functions: the drawing panel. We tested the

usage of the brushes and colors in our drawing page to ensure that one of the main functions of this application serves its purpose. The process of this experiment includes launching the application on different devices to test its consistency, and running the application long enough to check its stability. We tested the program on both Windows and Apple devices, such as laptops, PC, iPads, and iPhones of different models, running the application on multiple devices simultaneously.

After a few trials, the result shows that all buttons implemented into the application respond correctly and timely. However, one technical issue was found. The application is able to function properly when multiple devices are running it at the same time, but not for long. Approximately 15 to 20 minutes of using the program, it occurs to some devices that the drawing function freezes and that the user is still able to pick colors and switch brushes, but they aren't able to draw. Figure 5 below compares the number of times the application freezes with regards to time. The only method to solve this problem within the application is to return to the main menu and begin a new drawing. Although there is a "save" button in the program that saves the users' drawings, the users no longer have access to edit their works.

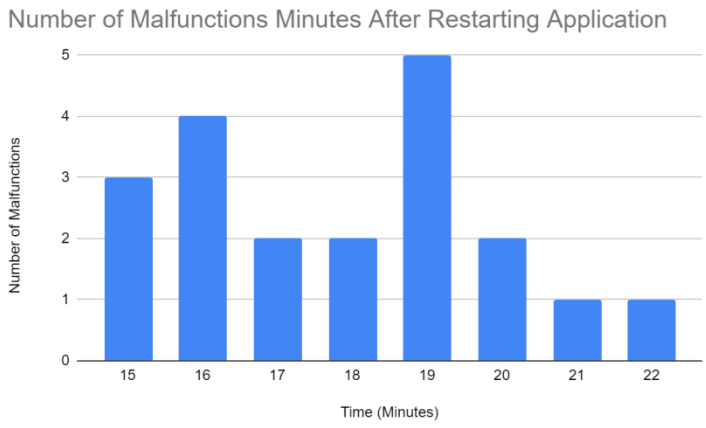


Figure 5. Number of Malfunctions Minutes

It is clear that some technical challenges are still awaiting to be solved in the application right now. A revision of the code is needed for the current version for further, advanced developments in the future. We will need to inspect what causes the crash of the application after approximately 20 minutes of usage.

4.2. Experiment 2

Besides manually testing the application for its functionality, we have also taken a step forward to introduce the application to real-life situations and audiences. After careful contemplation of the program, we decided to organize a workshop that presents and utilizes The Drawing Plan. Tasks such as making a flier, rearranging the program's website, and promoting the lesson to the corresponding community are all parts of this workshop. This workshop takes the form of storytelling and art. A few topics are organized to be discussed and drawn during the lesson to guide the students to tell their stories. For instance, prompts like "Pairing up with a partner, have them tell you what they did today, and you can draw it out!", and "I will draw something now, and someone will make up a story of what I draw!" motivated the young children to actively engage with each other and in our activities. Eventually, we received participation from eleven children under the age of 10 along with their parents and guardians attending our lesson.

It is very intriguing to see our young students' works of art when they are told to draw as many of them evoke a clear sense of personality and style for art. Specifically, each individual's use of colors, composition, texture, shapes, and the content of their drawings vary. One student adopted abstract elements in their drawings, such as the depiction of colors in geometric shapes, the overlap of rough and bold brush strokes using the acrylic and crayon textures. Another student, prominently, favors round compositions by building a circular frame on his canvas each time he draws, and utilizes softer textures such as the watercolor and pencil brush. He illustrates looseness in his drawing by using blocks of colors but with dynamic lines and shapes. And some other students work more graphically with direct, simple, and straightforward lines portraying their subject of art, rendering realism with a wide range of colors.

Through these student works, we, as the developer and designer of this application, can see their individual characteristics: those who appear introverted and uncommunicative can be as artistic and expressive as those who are talkative and extroverted in their works of art. This experiment result leads to the ultimate purpose of this application: tell your story, feelings, and emotion with art.

Prominently, one conversation that I remember having with a young girl in the classroom, regarding the drawing prompt, "Draw a person and a hand!", clearly demonstrated that she is living a safe, healthy live. The intention of this prompt was to bring out possible experience of physical assaults or harassment. One who has had such experience, and is a child, has the chance to reveal what happened to them and may be deeply terrified when they encounter this prompt. Yet, this girl in class was bored and disappointed when this drawing prompt appeared on her screen because she saw no purpose for it. Her work consisted of a person standing, and a separate drawing of a hand. The nature of her reflects her life, and that her joyous, communicative personality is authentic.

5. RELATED WORK

In Allison Cuellar's "Preventing and Treating Child Mental Health Problems", she addresses the potential flaws in the United State's policies and programs for treating children's mental health, including the lack of funds and ineffectiveness of current treatment approaches [10]. Cuellar suggests that most treatments focus on relieving symptoms instead of curing the patients from their roots, which is the reason why success in combating mental illness sometimes doesn't last. The Drawing Plan has the intention to treat mental disorders among children by discovering what the child struggles with, with the use of drawing prompts; this embarks the journey of uncovering the cause and further exploring the needs of the child.

In "Therapy to Improve Children's Mental Health", published by The Centers of Disease Control and Prevention(CDC), suggests three types of effective therapies for children: parent training in behavior management, child behavior therapy, and cognitive-behavior therapy [11]. These types of therapy often include talking and playing, either individually with the patient or the patient's parents, or as a group. Communication and fun activities can undoubtedly ease a child's mind to help adults understand their situation, drawing can, too. The functionality of The Drawing Plan can add to the modules of these therapy, providing an alternative method to approach child therapy. Especially to treat autism, art and creativity is a substantial way of communication.

The National Institute of Mental Health states that when it comes to treating children's mental disorders, it is recommended to communicate with parents and close adults around the child about the child's usual behavior [12]. Yet, a parent or caregiver cannot be mindful and especially omniscient at all times. Rather, it is important to pull attention onto the child and have them

express their struggles through a method that suits them, some, it would be art. In conclusion, verbal communication and physical behavior, such as playing, drawing, facial expression, should be combined in therapy for a more accurate result on a child's psychological stage.

6. CONCLUSIONS

The Drawing Plan is an application where young children are motivated to create and draw their very own stories with the guidance of provided drawing prompts. Upon entering the Get-A-Prompt option of the application, the randomly chosen drawing prompts are designed for children under their first-stage treatment, which is for adults to get to know the child's struggles. Then, the Make-A-Prompt option is designed for adults to be directly involved in the care of a child's mind: they can create specific drawing prompts for the child. Through the experiments done for this application, we are sure that The Drawing Plan has a value and that it will serve a purpose of child psychology and therapy [13]. Witnessing a range of personalities and interests through drawings produced by young children proves the usefulness of this application.

One limitation of The Drawing Plan is that the current drawing prompts may only reveal superficial problems for some users. For instance, some of the current drawing prompts says, "What does you dream birthday party look like?", or, "You are home alone, what do you do?". These prompts may be effective for children with mental disorders such as autism or ADHD, in which the symptoms of these diseases include a lack of response or over response [14]. Yet, what if the child only suffers mild symptoms? Such as they have friends and accompany but emotionally feel insecure and lonely? A child like this may still draw the positive aspect of their life when they are asked to, yet adults and caregivers may not be able to detect the hidden emotions. The Drawing Plan needs to take further consideration to enhance the current prompts in order to ensure better accuracy in its results.

To solve this limitation, I will be examining more books and articles regarding children psychology and therapy to find out how to guide a child's communication more effectively. I will also be launching more experiments and community services related to this topic to gain practical understanding of children's behaviors [15].

REFERENCES

[1] Rabkin, Judith G. "Opinions about mental illness: a review of the literature." Psychological Bulletin 77.3 (1972): 153.

[2] Hyman, Steven, et al. "Mental disorders." Disease control priorities related to mental, neurological, developmental and substance abuse disorders (2006): 1-20.

[3] Stengel, Erwin. "Classification of mental disorders." Bulletin of the World Health Organization 21.4-5 (1959): 601.

[4] Makeig, Scott, et al. "Linking brain, mind and behavior." International Journal of Psychophysiology 73.2 (2009): 95-100.

[5] De Bellis, Michael D., and Abigail Zisk. "The biological effects of childhood trauma." Child and Adolescent Psychiatric Clinics 23.2 (2014): 185-222.

[6] Orth, Ulrich, and Richard W. Robins. "The development of self-esteem." Current directions in psychological science 23.5 (2014): 381-387.

[7] Miller, William R., Joan Zweben, and Wendy R. Johnson. "Evidence-based treatment: why, what, where, when, and how?." Journal of substance abuse treatment 29.4 (2005): 267-276.

[8] Cohen, Judith A., et al. "Trauma-focused cognitive behavioral therapy for children and adolescents: An empirical update." Journal of Interpersonal Violence 15.11 (2000): 1202-1223.

[9] Pekkala, Eila Tellervo, and Lars Bertil Merinder. "Psychoeducation for schizophrenia." Cochrane database of systematic reviews 2 (2002).

[10] Cuellar, Alison. "Preventing and treating child mental health problems." The Future of Children (2015): 111-134.
[11] McEwan, Kimberley, Charlotte Waddell, and Jayne Barker. "Bringing children's mental health “out of the shadows”." Cmaj 176.4 (2007): 471-472.
[12] Robins, Lee N., et al. "National Institute of Mental Health diagnostic interview schedule: Its history, characteristics, and validity." Archives of general psychiatry 38.4 (1981): 381-389.
[13] Cairns, Robert B., and Jaan Valsiner. "Child psychology." Annual review of psychology 35.1 (1984): 553-577.
[14] Rutter, Michael. "Concepts of autism: a review of research." Child Psychology & Psychiatry & Allied Disciplines (1968).
[15] Fabes, Richard A., Carol Lynn Martin, and Laura D. Hanish. "Children's behaviors and interactions with peers." (2009).

USE OF AI TO DIVERSIFY AND IMPROVE THE PERFORMANCE OF RF SENSORS DRONE DETECTION MECHANISM

Fahad Alsifiany

Department of Information Technology and Communications, King Fahad Security College, Riyadh, Saudi Arabia

ABSTRACT

Drone terrorism may seem elementary and efforts in its mitigation may seem painless. The fact is that security bodies in many countries are still grappling with this growing security concern. The autonomous nature of drones and the unpredictable nature of drone attacks remain to be some of the unforeseen challenges undermining the mitigation efforts in combating drone terrorism. The need to upskill our security forces and the general public on the operational practices and security capabilities in the drone world cannot be overemphasized. This paper explores a futuristic solution to the current challenges encountered in the war against drone terrorism. In its design, it delves into the possibility of utilizing Artificial Intelligence (AI) in characterizing the features of drones identified in our airspace to determine their authenticity. It further enriches the employees of the security services and the general public with information on combating drone terrorism by benefiting from the accumulated experiences of the relevant and specialized affiliates.

KEYWORDS

Drone terrorism, Drone, Radio Frequency (RF), Unmanned Aerial Vehicles (UAV), Artificial Intelligence (AI), Computer Vision (CV), Machine Learning (ML)

1. INTRODUCTION

Drones are autonomously flying robots, which are controlled remotely using flight-planning software in their embedded systems. Drones can fly in altitudes from as low as 400 feet to as high as 33,000 feet, which is equivalent to 10 kilometers [1]. The altitudes of choice are dependent on regulations of unmanned aerial vehicle (UAV) flights, which vary across different countries [2]. The altitude of flight is mainly characterized by the size, weight, and purpose for which the drone is used [1]. The major types of drones include multi-rotor drones, single-rotor drones, fixed-wing drones, and fixed-wing hybrid VTOL drones. The fixed-wing hybrid VTOL drones combine the efficiencies of fixed-wing drones and multi-rotor drones by switching between the two modes of operation during their flights. The sizes of these drones are different and their weight can range from 5 kg up to 100 kg. In addition to their basic weight, drones can carry variable loads during their flight [2].

Drones have been used for different purposes. One of the pioneering applications of drones was in photography and image collection. Photographers have employed drones to collect photos at family events, sporting events, and nature [1]. Drones have also been used for recreational purposes. This is common with tourists who engage in drone racing competitions. Commercial

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 29-40, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130504

drones have also been used by large-scale logistic firms in management and delivery processes to save on the cost of human labor. The use of drones has also gained popularity in the field of security. In this field, commercial drones have been used for surveillance, inspection, monitoring, and data collection and most recently have been misused in drone terrorism [3].

1.1. Theoretical Framework

The invention of drones was a culmination of the second world war in the 20th century. The drones were used in military ventures and it was not until the year 2006 that the Federal Aviation Administration (FAA) issued the first commercial drone licenses. Since then, the use of drones has been embraced across different countries and one of the sectors that have majorly benefitted from their use is the security sector. The applications of drones in the security sector range from surveillance purposes to reconnaissance, monitoring, inspection, and data collection. The use of drones in terror attacks is slowly gaining popularity over the last four years. Drones have been used to fire missiles, drop bombs, or crash into the territories of target terrorists [3].

To mitigate drone terrorism, various technologies have been embraced. Radio Frequency (RF) has been the most prevalent technology in use [4]. The RF spectrum is between the range of 20 kHz and 300 GHz. Drones are autonomous and their controllers use the RF spectrum [3]. A varied range of RF equipment has been developed for detecting and analyzing drones in airspaces to determine their authenticity and identify rogue drones, which may have been deployed by terrorists. This equipment includes RF sensors, RF analyzers, RF jammers, and radar. Apart from RF technology, there has been an exploration of other approaches such as the use of acoustic sensors, use of optical sensors, and use of high energy lasers [5]. These efforts have however not attained the threshold we envision in the fight against drone terrorism. The need for exploration of more agile technologies in this course is undebatable.

1.2. Background

Artificial Intelligence (AI) is a flexible and agile technology, which serves to resolve most of the challenges revolving around the world of drone terrorism. AI involves emulating human behavior and thinking in computers or computer-controlled devices. Drone terrorism is rapidly evolving and drone terrorists have scaled up both technically and tactically. Drone attacks experienced today are therefore more sophisticated and require high-end efforts to combat them. AI was developed in the 20th century. For a while, the technology grew unnoticed as it had low levels of creativity. Today technology has however evolved to serve most of our needs with development in support technologies and the emergence of new technical skills. Companies offering drone defense solutions such as Dedrone in the United States of America are beginning to develop AI-based solutions for combating drone terrorism [11]. Exploration of the different aspects in which this technology can be utilized in combating drone terrorism will help in realizing optimal security measures for use by both members of the security service and civilians.

2. ANALYSIS

Artificial Intelligence (AI) leverages computer capabilities to realize human intelligence in computers. The technology draws a comparison between a prevalent situation and a vast range of databases with data parameters related to this task. Data utilized in AI is mainly sourced from similar occurrences in the past or formulated as a set of ideal parameters desired in a given course. In performing this comparison, AI classifies different events and their associated characteristics.

Radio Frequency (RF) technology served as the initial solution for countering drone terrorism. It has been the most prevalent technology in use, and it offers a futuristic pathway in the war against drone terrorism. RF sensors, RF analyzers, radars, and RF jammers utilize this technology [3]. RF sensors couple the efficiency of magnetic susceptibility and electrical permittivity to detect drones in their vicinity. The core components of these sensors are a monitoring circuit, a coil-shaped RF antenna, and an RF oscillator circuit. The sensors are also used in determining the proximity of attacking drones. RF analyzers are used in airspaces to determine the frequency utilization and detect different forms of interferences in the radio spectrum [4]. This way, they can detect enemy drones operating at divergent frequencies.

Radar detection systems utilize RF technology to range the distance, measure the angle and calculate the radial velocity of drones detected by sensors, to populate information relevant to the classification of the detected drones [4]. RF jammers are used at advanced stages after detection, where they generate waves similar to those of attacking drones to create interference and prevent signal relays between these drones and their controller stations [3]. Other technologies coupled with RF technology in countering drone terrorism include the use of acoustic sensors, use of optical sensors, and use of high energy lasers.

The dynamism associated with drone attacks is overwhelming the aforementioned RF technologies [5]. Table 1 details the dynamic features of attacking drones, which are posing technical complexity to the fight against drone terrorism.

Table 1. Dynamic features of attacking drones.

Parameter	Complexity
Size	Variation and unpredictability in the sizes of commercial drones in use by drone terrorists
Weight	Variation in the weights of different drones and establishing the credibility of drones of a specific weight
Shape	Varying shapes of commercial drones being utilized by drone terrorists
Number	Migration of drone attackers from conventional attacks using a single drone to using swarms of drones
Altitude	Variation and unpredictability in altitudes of flight for drones used in drone terrorism
Flight pattern	Variable flight patterns by different drones make it difficult to differentiate between authentic and rogue drones
Takeover and fend off	Lack of technical tools to execute takeover and fend off identified rogue drones
Speed	Variable speeds of flight for different drones, some exceeding the defined levels

To counter the dynamism, AI is being embraced in the war against drone terrorism. This technology is promising and as it's in the initial stages of exploration, huge research efforts are required in this field. By coupling AI technology with RF technologies, greater efficiencies will be realized across this field. This way, the underlying dynamism will be contained as follows:

2.1. Size and Shape

AI utilizes different algorithms, modeled to perform specific roles. Computer vision (CV) is a sub-branch of AI, which utilizes object detection and object recognition algorithms [7]. Object detection algorithms use multiclass classification to define the type of objects and establish their characteristics. The size and weight of drones vary depending on the manufacturer, the purpose for which the drones are used, or the type of load carried by the drone [1]. Digging deep into

these drone parameters can help in the primary steps of detecting rogue drones weaponized by terrorists [1]. RF sensors can be used in conjunction with AI technology to analyze the shapes and sizes of drones.

RF sensors solely detect the presence of any type of UAV in the air space [5]. While they do not single out instances of drones, it would be efficient to utilize Computer Vision to single out instances of drones and derive their associated parameters. This could be achieved by embedding AI processors in the sensors and equipping the RF sensors with quality camera systems to work in synchrony. On detecting a UAV, the camera would be enabled to take images of the UAV and the image will be sent to the AI processors for recognition. A colossal amount of data is required for training the AI model to support differentiation of the UAVs to primarily support the detection of drones and to characterize the shape and sizes of these drones. A sound understanding of the local UAV regulations will help in formulating data for use in this context. We can derive the most commonly used or licensed drone sizes and their corresponding shape and use this data to train our AI model. This way, classification after drone detection and recognition may point out an anomalous size or shape for a given drone, which can further be monitored to determine whether it is an attacking drone. Figure 1 shows the conceptualization of the hybrid AI and RF model.

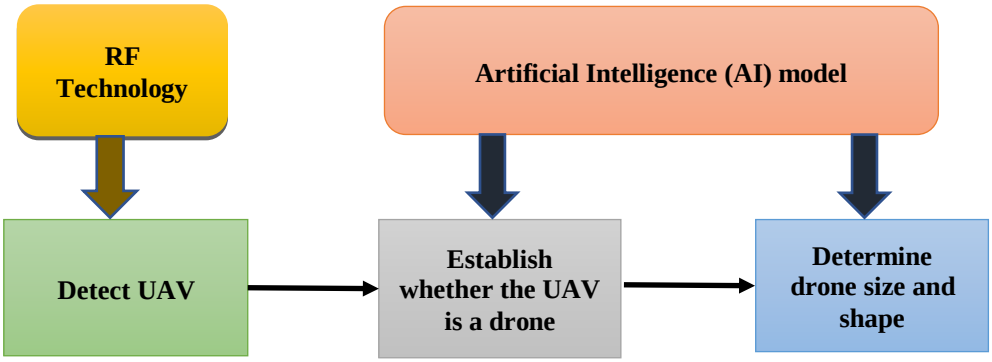


Figure 1. The hybrid AI and RF model workflow

2.2. Weight

The weight of a drone is a critical parameter in the war against drone terrorism. Weaponized drones often carry variable loads with them, which may be in the form of missiles or bombs. While RF technologies can barely identify the weights of drones in airspaces [1], it is essential to embrace AI for this use. We can once again leverage the benefits of Computer vision to determine the weight of drones. While we cannot achieve an AI model that blatantly tells us the weight of a hovering drone, we can develop models that can measure the volume and density of these drones.

To establish the volume of the hovering drone, we would use a 3D reconstruction algorithm in the AI model. Here, a camera is needed to work in synchrony with the RF sensors and the embedded AI system. Multiple images of the drone under scrutiny will be taken from different angles and the algorithm will reconstruct the 3D view of the drone and determine its volume. RF technologies cannot however establish the density of drones under scrutiny [4]. To derive the material density of the drone components, active thermography will be used [6]. A laser controller will be incorporated into the AI model to activate the laser beaming over each drone under scrutiny. The thermal gradient created by the incident laser on the drone surface will be

taken as data input by the AI model. Continued scanning will be done to achieve a series of thermal frames with which the AI model will estimate the material density.

The model can consequently calculate the weight using the aforementioned parameters with the relationship below [6]:

$$Weight = v \times \rho \times g \ ,$$

Where v is the volume of the drone, ρ is the density of the drone, and g is the acceleration due to gravity.

2.3. Number and Altitude

RF technologies currently in use in rogue drone detection are being challenged by the drastic tactical approaches that terror attackers are using. Attackers are accustomed to using one drone to fire missiles, drop bombs, or to even crash into a target premise. Nowadays, attackers have resolved to use a swarm of drones which makes it hard to use RF technology in scrutinizing the drones [1].

AI can help us in establishing the number of drones in a particular swarm. Computer vision is efficient with such processes. By using a quality camera that can relay to an embedded AI system, we can determine the number of drones in the swarm. Computer vision algorithms can also be employed in determining the altitude of each drone in the swarm. Conventionally, radar ranging is efficient in establishing the altitude of flight in cases of isolated drones. Radar ranging uses frequencies between 400 MHz to 36 GHz [4].

Complexity arises in establishing the heights of multiple drones in a swarm and AI poses a solid solution to this challenge, through its high-end algorithms. By establishing the number of drones and their corresponding altitude of flights, our developed AI model would draw from a huge set of primary data on local UAVs flight trends and alert of any observed anomalies to take action on any rogue drones.

2.4. Flight Pattern

Ideally, drones used by attackers are likely to have suspicious flight patterns as they surveil their targets. Such suspicious patterns may include circling a given target or making multiple return flights over a target to establish its state. Such anomalies in flight patterns cannot be established by RF technologies currently in use. There is a need for an intelligent model, which can perform an analysis and raise alerts in cases of suspicion. This can be achieved through a combination of Computer Vision and Machine Learning (ML), which are sub-branches of AI.

ML algorithms provide us with an opportunity to define human-like behavior in a system [2]. A model can be developed and trained with data on common flight patterns of different commercial drones. In this approach, Computer vision will ensure the relay of drone images and videos to the embedded AI system. The ML model will then raise alerts on any anomalies detected in the flight patterns of the drones and appropriate action can be taken on any rogue drone.

2.5. Speed

Every country has its own defined maximum speed of flight for UAVs in its airspace. Drones operating at a speed greater than the maximum defined speed are rogue drones as they do not

operate under the stipulations of their licenses. Computer vision can be used to monitor the speeds of drones to single out instances of rogue drones.

2.6. Takeover and Fend Off

In the war against drone terror attacks, fend-off has previously been employed using RF technologies. Fend off is the process of identifying the signal source of the drone controller and disabling the signal [10]. The drone consequently flies back to its original position or hovers around. This way, the drone can be monitored easily to determine whether it is an attacker's drone. The development of AI technology brings us the possibility of achieving drone takeover. In a takeover, after fend off has been achieved, the scrutinizer can establish autonomous control of the drone and combine the efficiencies of Computer vision and light detection and ranging (LiDAR) to land the attacker drone safely [10]. LiDAR is important as it helps in establishing the presence of objects or vegetation, and when combined with Computer vision, the scrutinizer can bring down the drone safely without causing any harm to people, structures, or vegetation. The proposed drone detection lifecycle for the hybrid AI, RF, and LiDAR system is shown in figure 2 below. Moreover, the proposed hybrid system is shown in figure 3 below

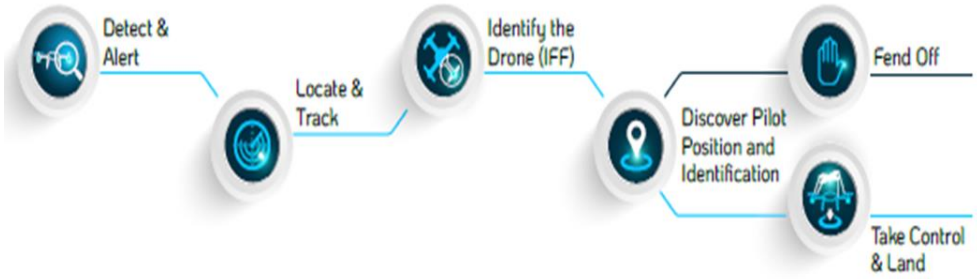


Figure 2. The proposed rogue drone detection lifecycle

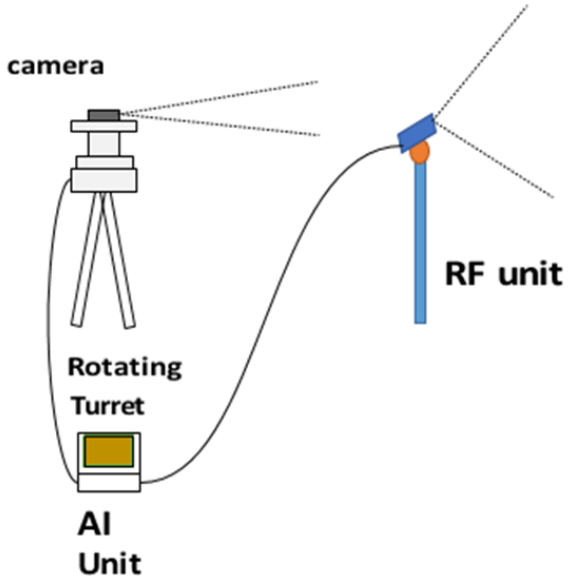


Figure 3. Proposed hybrid system

2.7. Mathematical System Model

The developed AI model for use will derive a wide range of parameters from the acquired images for regression analysis to determine the volume, weight, altitude, and flight patterns of the drones under scrutiny. These parameters will include area, shape, perimeter, eccentricity, axis length, radial distance, altitude, and instantaneous displacement of the drones over different angles and directions. The conceptualized algorithm is depicted in figure 4 below.

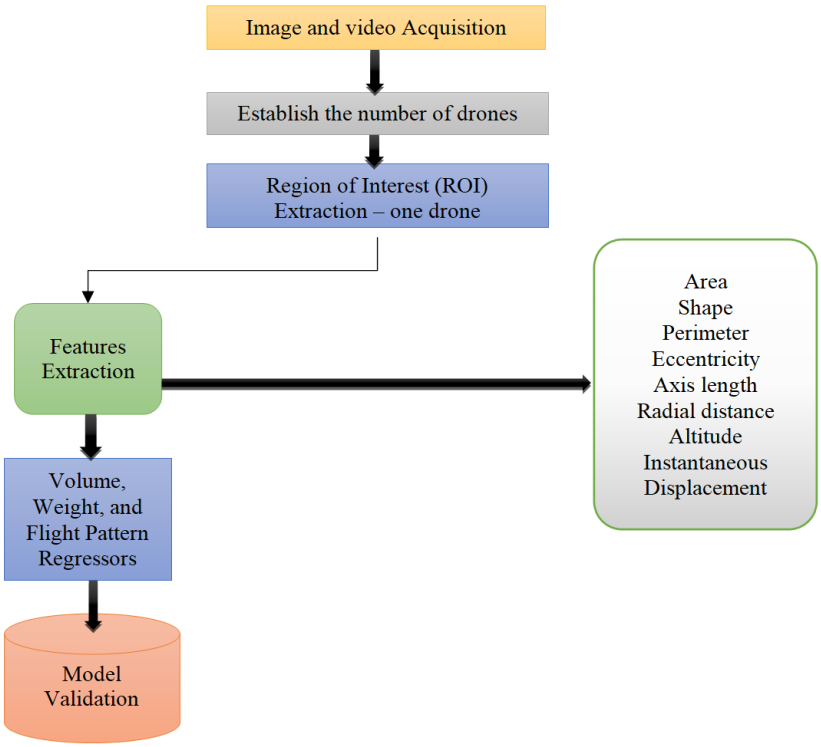


Figure 4. Conceptualized algorithm.

The proposed model will utilize 2D image feature extraction from the areas of a drone projected across different angles and directions. The 2D image features will then be used for 3D image reconstruction. The key parameters of interest are represented in table 2 below.

Table 2. Key parameters of interest.

Parameter	**Representation**
Y_A	Area
Y_p	Perimeter
Y_E	Eccentricity
Y_1	Major axis length
Y_2	Minor axis length
Y_d	Radial Distance
H_1	Average swarm altitude

Feature extraction in AI models relies on counting the number of pixels bound by contours identified in the images under analysis [9]. Taking a random vertex (x_n, y_n) for the captured drone image and constructing an ellipse of coordinates (X_1, Y_1), (X_2, Y_2), we deduce (M_2, N_2) and

(M_1, N_1) to be the endpoints of our major and minor axis respectively. We then proceed to extract the features of interest using the following regressions [9]:

$$Area, \qquad Y_A = \frac{1}{2}\sum_{n-1}^{N-1}(y_{n+1}x_n - x_{n+1}y_n)$$

$$Perimeter, \qquad Y_p = \frac{1}{2}\sum_{n-1}^{N-1}(x_{n+1}, y_{n+1}) - (x_n, y_n\ )$$

$$Eccentricity,\ Y_E = \frac{Y_1}{Y_2}$$

$$Major\ axis\ length, Y_1 = \sqrt{((X_2 - X_1)^2 + (Y_2 - Y_1)^2)}$$

$$Minor\ axis\ length, Y_2 = \sqrt{((M_2 - M_1)^2 + (N_2 - N_1)^2)}$$

$$Radial\ distance, Y_d(n) = \sqrt{\{X(n) - \bar{X}\}^2 + \{Y(n) - \bar{Y}\}^2}$$

$$Average\ swarm\ altitude, \qquad H_1 = \frac{Radar\ height_{(ROI1)} + \ \ldots + Radar\ height_{(ROIn)}}{Number\ of\ ROIs(Drones)}$$

The AI model utilizes statistical Regressors to give an estimate of the volume, weight, and flight pattern of the drones once the 2D images have been reconstructed into 3D images. The volume obtained after a 3D reconstruction of the images is then used in determining the weight of the drone under scrutiny using the relation [8].

$$Weight = v \times \rho \times g$$

The density of the drone material is obtained by scanning the region of interest, initially defined to be the drone under scrutiny at that time. The altitude of flight for a single drone is obtained using radar ranging. In the case of a swarm of drones, however, the AI model works in synchrony with the radar ranging system to find the individual altitudes of the different regions of interest and find the average height of the swarm for further characterization of the flight behavior and variation from ideal flight parameters to determine whether they are rogue drones.

$$Rogue_{ROI} = \left|(\sum(ROI_s + ROI_H))\right|_{size,shape,volume,weight,flight\ pattern}$$

Where ROI represents a single drone, ROI_s is the speed of a single ROI, and ROI_H is the altitude of a single ROI.

The characterization of the magnitude of the speed and altitude of a single ROI is then done for the specific instances of size, shape, volume, weight, and flight patterns. The correlation between this characterization and the prevalent data in the AI model is then established, to detect any inconsistencies in the ROI under scrutiny. Below is the algorithm utilized for the above classification:

$$Rogue_{ROI} = \left|(\sum(ROI_s + ROI_H))\right|_{size,shape,volume,weight,flight\ pattern}$$

If altitude > 400
Or speed > 100 mph
Or length > 3.5:
Rogue drone
Fend off
Take over

Else:

Licensed drone

End

The algorithm for identifying a swarm of drones was as detailed below. The maximum difference in the Line of Sight (LoS) distance between different drones in a swarm was assumed to be 10m.

$$The\ separation\ between\ ROI, x = LoS_{ROI_n} - LOS_{ROI_{n-1}} \leq 10m$$

If x > 10m:

Not a swarm
Else:
Swarm
Fend off
Take over
End

3. RESULTS AND RECOMMENDATIONS

The developed AI model was simulated for different drones to determine the model's efficiency levels in detecting rogue drones. The sets of data used for training and correlation analysis were however limited and were drawn from basic laws regarding the operations of UAVs in Saudi Arabia. The maximum height of flight for UAVs in Saudi Arabia is stipulated to be 400 feet above the ground. 3 drones were scrutinized and characterized as shown in figure 5 below. Figure 6 shows the characterization of 4 drones based on their speed of flight.

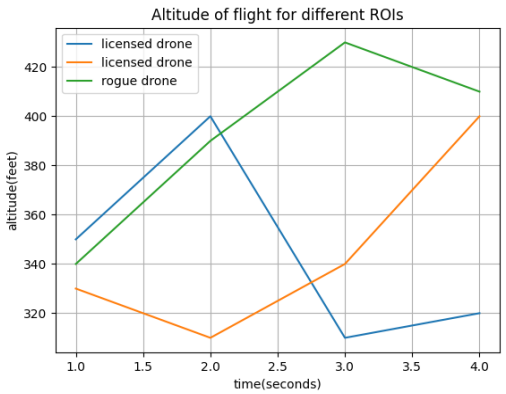


Figure 5: Altitude-based characterization

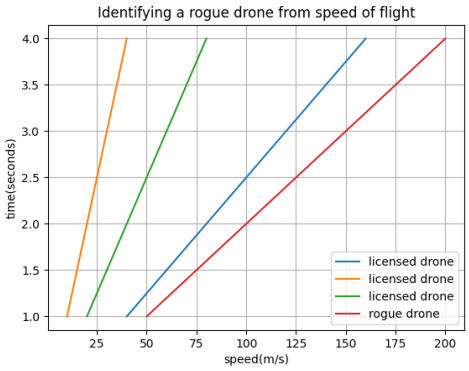


Figure 6: Speed-based characterization

In figure 5, two drones were identified as licensed drones as their average altitude of flight was below the stipulated maximum height of 400 feet. One drone was characterized as a rogue drone with its altitude of flight extending to about 430 feet above the ground. In figure 6, 3 drones were characterized as licensed drones with their speed of flight being less than the maximum stipulated

speed of 100 mph. One drone was characterized as a rogue drone with its speed momentarily surpassing the maximum stipulated speed.

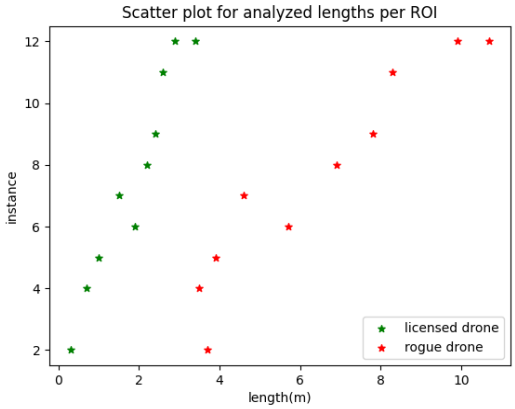


Figure 7: Length per ROI

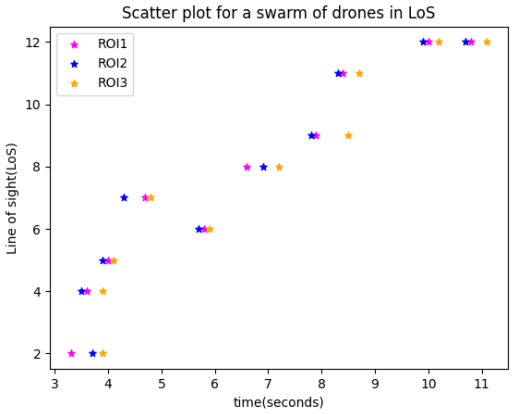


Figure 8: Identified swarm

In figure 7, a scatter plot is presented for two classifications based on the sizes of the drones. Ideally, the training data consider licensed local UAVs to have a prevalent length of up to 3.5 m. A series of drones whose lengths ranged between 3.7 m and 10.5 m were characterized as rogue drones. In figure 8, the AI model was used to identify a swarm of drones. The distance of the drones across the lines of sight was monitored over a given time and the scatter plot reveals a similarity in their pattern of flight, which proves the set of drones to be in a swarm. According to Saudi Arabia laws on UAV flights, only one drone should be flown by a controller at a given time. The identified swarm would therefore fall in the rogue drone characterization category.

Figure 9 depicts a fend-off process inflicted on the rogue drone identified in figure 5 from the unusual altitude of the flight. In the fend-off process, the RF unit disabled the RF signal from the remote controller of this drone. This resulted in the drone hovering around without further propagation in the LoS. The takeover process is depicted in figure 10 where data was populated for a rogue drone flying at unusual altitudes and lowered to the ground.

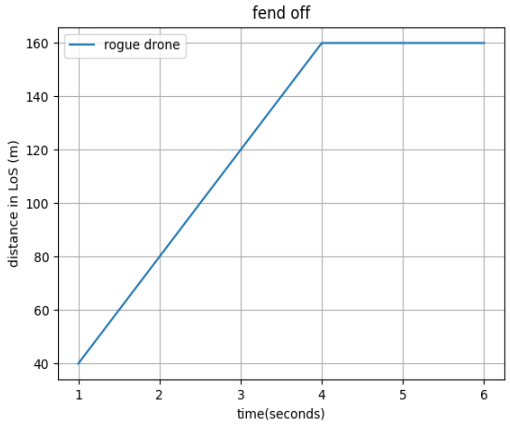


Figure 9: Rogue drone fend-off

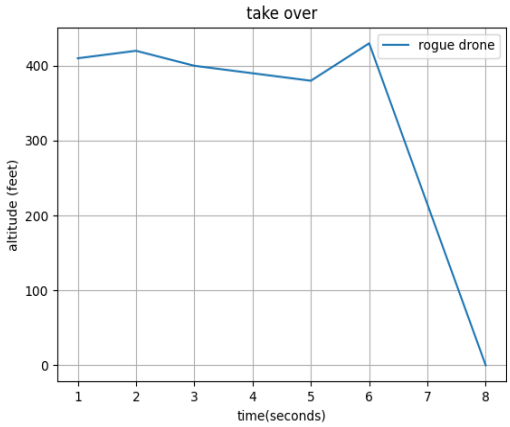


Figure 10: Rogue drone takes over

Characterization based on shape, volume, weight, and flight patterns however requires huge sets of data. The data should ideally be sourced from market stakeholders and commercial drone vendors. While this simulation involved minimal data sets, more market research to derive more market data can help in developing an ideal hybrid AI and RF system for use in the fight against drone terrorism.

4. CONCLUSION

With drone terrorism continually on the rise, there is a dire need to gear up in the war against drone-related attacks. Improvement in technical and technological skills will serve to be pivotal in this process. This research work puts across AI as a key futuristic technology for use in mitigating drone-terror attacks. From the proceedings, this technology merits in terms of its flexibility and efficiency for use in deriving the key features of interest in drone identification and classification. A good development trajectory is required for this technology and this calls for an optimal investment of financial, physical, and human resources by our security services. While the big chunk of this war against drone terrorism can be rested on the hands of our security service men, we cannot downplay the role civilians need to play. Civilians should be wary of suspicious drone activities in their environments and they should report such instances to local law enforcement officers on time. This way, we can achieve optimal drone terror mitigation practices.

4.1. Future Work

These future explorations would help in realizing a robust hybrid AI and RF model for use in rogue drone detection.

- Developing Deep Neural Networks (AI) algorithms and combining their use with infrared cameras to enable rogue drone detection at night, a realization that was not feasible with the developed model.
- Establishing a high-performance software platform, which can be used for deploying the hybrid AI and RF model to enable its remote control in both Line of Sight (LoS) and Non-Line of Sight (NLoS) rogue drone detection approaches.

REFERENCES

[1] Rico Merkert, James Bushell (2020). Managing the drone revolution: A systematic literature review into the current use of airborne drones and future strategic directions for their effective control. *Journal of Air Transport Management*, 89(1).

[2] Syed Agha HassnainMohsan, Muhammad Asghar Khan, Fazal Noor, Insaf Ullah & Mohammed H. Alsharif (2022). Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review. *MDPI*, 2(3).

[3] Jean-Paul Yaacoub, Hassan Noura, Ola Salman, Ali Chehab (2020). Security analysis of drones' systems: Attacks, limitations, and recommendations. *Elsevier - Internet of Things (IoT)*, 3(4).

[4] Erdemli, Mustafa Gokhan (2009). General use of UAS in EW environment--EW concepts and tactics for single or multiple UAS over the net-centric battlefield. *Naval Postgraduate School*, Monterrey, California.

[5] Alexander Farrow (2016). Drone Warfare as a Military Instrument of Counterterrorism Strategy. Air and Space Power Journal, 3(4), 7-12.

[6] Tamas Aujeszky, Georgios Korres, Mohamad Eid and Farshad Khorrami, (2019). Estimating Weight of Unknown Objects Using Active Thermography. MDPI, Robotics.

[7] Ahmed Reda Amin EL-Barkouky (2014). Mathematical modeling for partial object detection. The University of Louisville, Theses and Dissertations.
[8] Baohua Zhang, Ning Guo, Jichao Huang, Baoxing Gu, and Jun Zhou (2020). Computer Vision Estimation of the Volume and Weight of Apples by Using 3D Reconstruction and Noncontact Measuring Methods. Hindawi, 17(3).
[9] Innocent Nyalala, Cedric Okinda, Qi Chao, Peter Mecha, TchallaKorohou, Zuo Yi, Samuel Nyalala, Zhang Jiayu, Liu Chao & Chen Kunjie, (2021). Wight and volume estimation of single and occluded tomatoes using machine vision. International Journal of Food Properties, 24(1)

AUTHOR

Fahad Alsifiany received a B.Sc. degree in electrical and electronic engineering from King Abdulaziz University, Jeddah, Saudi Arabia, in 2001. He received an M.Sc. degree in telecommunications and networking engineering from the University of Pittsburgh, Pittsburgh, USA, in 2011. He received his Ph.D. degree in wireless communications from Newcastle University, Newcastle upon Tyne, U.K, in 2020. He is a lecturer now with the Faculty of King Fahad Security College, Riyadh, Saudi Arabia. His current research interests include noncoherentmimo systems, physical layer security, massive mimo, artificial intelligence, and machine learning.

AIDANCEFRIEND: AN INTELLIGENT MOBILE APPLICATION TO AUTOMATE THE DANCE RATING USING ARTIFICIAL INTELLIGENCE AND COMPUTER VISION

Yuanyuan Ding[1], Shuyu Wang[2]

[1]Sage Hill School, 20402 Newport Coast Dr, Newport Beach, CA 92657
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

In recent years, dance has become a popular entertainment for many people and also an occupation. As a dancer, sometimes it is hard to check how close your cover is vs. the choreographer's because our eyes are not always accurate when we are judging dynamic movement of people, so can artificial intelligence help us to do the work? This paper develops an application which utilizes artificial intelligence, and data analysis skills to develop an application which works on dance scoring [4]. In the application, users can upload two videos, one is their own cover while another one is the original choreography. Then, the application will use MediaPipe to catch the angles of dancers' bodies in frames then store them in a data abstraction [5]. After all data are collected, the application will use clustering to line up the frames and angles information that are stored. The steps above will be applied to both videos. Next, the application will use an algorithm to compare two videos' data and calculate a percentage of error of the covering video to the original choreograph and report a grade to the user. We applied our application to users who want to check how similar their covering dances are compared to the original choreographs in order to improve their covering quality [6]. The results show that when users are improving the quality of their covers, they improve their skills of focusing and optimizing details in dance.

KEYWORDS

Artificial Intelligence, Clustering, MediaPipe

1. INTRODUCTION

Covering others' choreographs is a way for dancers to improve their choreo skills and also the skills of performing details. However, checking the quality of coverings is sometimes hard because people's eyes are struggling to accurately compare two dynamic-movement videos, especially when the protagonist of the video is people themselves. At the same time, improving a cover is a process of fixing details but human eyes are also inefficient in detecting the difference of details in the videos, so improvement of a dance is also hard to be discovered. Therefore, dancers usually have a hard time judging their covering quality based on similarity, and their solutions are often finding friends to judge but the results are not ideal. Regarding this problem, artificial intelligence (AI) becomes a choice of solution. AI is a product of logic, so using AI to do the analysis by giving it a standard and data is an ideal way to help dancers to grade their covers. Asking dancers of different ages near me, a general result I got is: we are facing the

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 41-50, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130505

challenge that is described above, and we hope there will be an APP that helps us to solve the problem. Therefore, based on the request and popularity of the problem, I decided to develop a mobile application to help dancers solve this problem.

Currently, there are many dance scoring systems, but these scoring systems are mostly designed for competitions instead of personal usage. At the same time, existing systems mostly score dance based on the quality of choreography, overall impression, performance, etc [7]. For instance, there is one dance scoring system which is called DanceBug. It is used for scoring a competitional dance. It is designed for judges or dance studios to perform a professional and strict process of dance judging and scoring. The system enables the users to list many dances performs with their scores.

However, for individual dancers who like to cover dances, they focus on the similarity compared to the original choreography because it will help them make their coverings better while improving their skills of fixing details. However, the DanceBug system does not help because it cannot automatically score a covering dance based on similarity between the original choreographer and the user. Therefore, existing methods are not able to satisfy this group of dancers.

Another existing method that involves dance scoring is called danceConvention. This method is a mobile application which is useful for individual users [8]. Its function is to give users the real time scores of competitors in a competition. Users can see the updates of scores of their competitors through the application and to predict their ranking in the competition. Although this application is convenient for competitional individual dancers, it still cannot satisfy individual dancers who like to cover dances habitually.

In our application, we focus on providing the services for individual dancers who like to cover choreography. Our application allows users to get a grade of similarity between their dances and the original choreography. First, users can upload their dances and the original choreography. Then, the program will run and collect information from the two dances by using the artificial intelligence tool and data abstractions [9]. After collecting data from two videos, the program will analyze the data and give users a grade of their coverings, such as A, B, C, and D. If the covering dance has more similarity with the original choreography, the grade will be higher. We also have a function that users can check their history grades of previous dances.

Compared to existing methods, our method is more attractive and more useful in individual dancer groups since many of them like to use coverings to improve their skills. Many individual dancers like to use coverings to improve their skills because copying choreographers' details help dancers to develop their own styles and gives them strength to increase details in their future dances. Users can choose to cover a dance many times, and they can see their improvement through the history function [10].

In order to test and prove the workability of our application, we gathered several data from the internet and also from our real life. In the first test, I uploaded two videos of me dancing the same dance but at different times. The result I got is an A which indicates a high similarity, and this result makes sense because the dance and the dancers are the same. In the second test, I uploaded my own dance covering video and my teacher's demo video of the same dance, and the result I got is also an A but the actual score –similarity percentage– that is shown in my editor (PyCharm) is a little lower than the first test. In the third test, I uploaded two videos of different dances that were collected from the internet. Then, the result I got is a D, which shows that the similarity between two videos is low. This result indicates that my application also works when the

difference between two videos is large. Repeating all scenarios with different videos of different dances, we tested that the combination of our algorithm and the method of sorting data works.
The rest of the paper is organized as follows: Section 2 points out the challenges that we faced during development and experiment; Section 3 focuses on methods and solutions that we used to solve the challenges that are mentioned in Section 2; Section 4 specifies the experiments that we did and Section 5 continues Section 4 to talk about related work of our application. In the end, Section 6 concludes our work and indicates our future wish and possible improvement of our project.

2. CHALLENGES

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

2.1. Videos Lining Up

The first challenge that we have in the process of developing the APP is how to line up two videos in order to make comparison of data. Since we use the method of processing two videos and collecting data sequentially and the number of frames of two videos may not be exactly the same, it is hard for us to compare and analyze data by lining up frames of two videos. The second method that we came up with is extracting the background music of two videos and line two videos up through comparing and according to the background music. However, it is also hard to line up accurately because many times the background music contains useless noises. If we want to eliminate these background noises, there is a probability that some parts of actual music will be eliminated unintentionally. At the same time, although we could solve the useless noises problem, lining up through music also contains a problem that after extracting and processing unrelated noises, it is hard to align the background music back to the mp4 videos. Therefore, these traditional methods that we thought are not implementable.

2.2. Storing Data

The second challenge that we have faced is how to store data of frames. Testing many videos of dance, we have observed that many videos have more than 1000 frames.

If we decide to store these frames in an abstraction, there will be a storage problem. Because of storing, we need to store these frames exactly in an abstraction in order for us to analyze. Storing frames and iterating them with analysis will also cause the run-time to be unexpected.

However, if we decide to store necessary information in frames, challenge 1 will come again because storing information in frames will face the problem of formatting and lining up this information, which makes challenge 1 more significant and harder. If we cannot solve the problem of lining up data by comparing frames, it is hard for us to line up data only with numeric information such as degrees of angles.

2.3. Scoring System

When we first created our scoring system, we were trying to average the angle difference of each frame and compare all the averages from all the frames, however, after we test some obvious test cases – two different dances' videos with the same length, and two videos that are exactly same – we found out that the results do not make sense. Therefore, we decided to create a new scoring system.

Our updated scoring system is to calculate the difference between angle degrees of two according frames from two videos, and we will add up all the differences from all the frames. Instead of taking the average of difference separately from each frame, we now take the average of difference from the sum of difference from all the frames. Then, based on the average error – in percentage – we will give out users a letter grade. If the average error is over 110%, it is considered a F. Average errors that are between 90 to 110 are D, between 60 to 90 is B, and less than 60 is considered as A. We don't give grade C because after testing many cases, test cases that show B and C are actually having no big difference, so we decided to delete letter grade C and remain A, B, D, and F – the difference between those four-letter grades are logical.

3. SOLUTION

AIDanceFriend is a mobile application which allows users who like covering other choreography to compare the similarity between their covering video and the choreographer's dance video [11]. Opening the application, the user is able to upload two videos from their devices. Then, the user only needs to click the "analyze video" button to run the program. The program will process two videos sequentially. We use MediaPipe to gather outlines of the dancer's movements while iterating the frames in the video [12]. Using MediaPipe to gather outlines of movements, we are able to calculate the degrees of angles of the dancer's joints of every frame. There are eight angles data each frame and the eight angles of a frame will be stored in a list, and there is a list of lists which stores angles information frame by frame. After collecting data into two lists, we will cluster lists in the two big lists in order to line up corresponding frames of two videos (frames that refer to the same movement). Lining up the frames, we are able to calculate the numeric difference of the same angle from two videos. Then, we utilize Flask to store the two videos in Google Firebase, which allows us to show users their videos uploading history. In the end, there will be a letter grade shown to the user which indicates how close their dance is compared with the choreographer's – A indicates very close, while D indicates merely close.

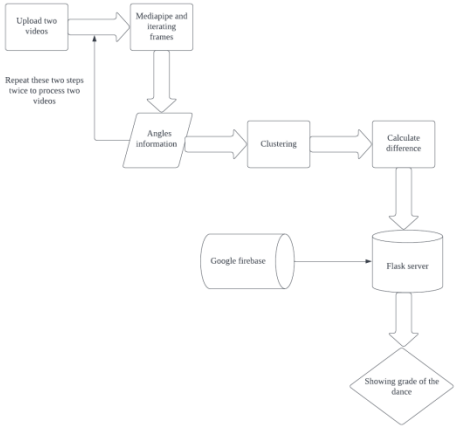


Figure 1. Overview of the solution

First, we use flutter to create an upload video page, which enables the users to click the upload button and to choose the videos from their own libraries, google drives, google photos, etc.

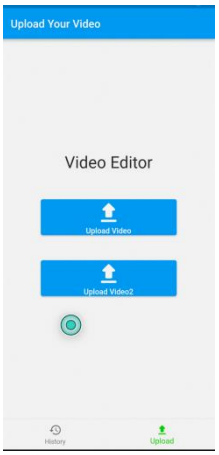


Figure 2. Upload video page

Then, the flutter will input two videos to the Python codes that we wrote for analyzing videos. In the process of analyzing videos, we use MediaPipe to collect body angles' information [13]. To calculate the angles information, we need to gather the positions – indicates by the x-y coordinate – of certain points, which includes: left shoulder, left elbow, and left wrist; right shoulder, right elbow, and right wrist; left hip, left knee, and left ankle; right hip, right knee, and right ankle; left shoulder, left hip, and left knee; right shoulder, right hip, and right knew; left wrist, left shoulder, and left hip; then right wrist, right shoulder, and right hip [14]. Information of three points – endpoints of an angle – are the inputs of the function calculate_angle. The calculate_angle function will perform the calculation of angle's degree by using the x and y coordinates of every point. Then, the function will return the rounded degree of that angle to the function that collects angle data. Taking the angle information from calculate_angle, the function will store all the degrees of angles that are described in the previous sentences in a list. Now, the iteration of the first video is over. After iterating the first video, our application is going to repeat the steps of collecting data to iterate the second video. Calculating the angles' degrees, we perform a certain angles calculation.

```
def calculate_angle(a, b, c):
    a = np.array(a)  # First
    b = np.array(b)  # Mid
    c = np.array(c)  # End

    radians = np.arctan2(c[1] - b[1], c[0] - b[0]) - np.arctan2(a[1] - b[1], a[0] - b[0])
    angle = np.abs(radians * 180.0 / np.pi)

    if angle > 180.0:
        angle = 360 - angle

    return round(angle, 1)
```

Figure 3. Screenshot of code 1

After collecting data that we need from the two videos, we start to cluster the data that we collected. We cluster data by comparing all angle information in a single frame. Based on the similarities of those angle information and the nature of clustering, we could line up frames appropriately. Then, we use the algorithm below to calculate the difference between angles and generate a number grade of the uploaded videos, which is called averageError in the algorithm.

```
errorTotal = 0
for arr in difference:
    for dif in range(8):
        if arr[dif] >= 15:
            allDifference.append(arr[dif])

for i in allDifference:
    errorTotal += i
print(errorTotal)
averageError = errorTotal / (len(difference) * 8 * 15) * 100
```

Figure 4. Screenshot of code 2

After calculating an averageError, we then translate the number grade to a letter grade.

```
if averageError <= 60:
    return("A")
if averageError > 60 and averageError <= 90:
    return("B")
if averageError > 90 and averageError <= 110:
    return("D")
if averageError > 110:
    return("F")
```

Figure 5. Screenshot of code 3

4. EXPERIMENT

4.1. Experiment 1: Pose Estimate Analysis between MediaPipe and YOLOv7

The YOLOv7 posture, a single-stage multi-person key point detector developed by Pytorch, is trained using the COCO dataset, which includes 17 landmark topologies. Segmentation supports both CPU and GPU, therefore it is not directly related to posture. In contrast to MediaPipe, a framework that can only detect one person, it can detect several people. MediaPipe only supports CPUs, and segmentation is built-in.

In this experiment, employing both posture models in a CPU environment, we compare the performance of the FPS on fixed model input size for record inference when just one person is represented on it. First, because 960x960 is the standard picture size, we altered the YOLOv7 algorithm to forward pass images that have been scaled to that size. The person's posture was then collected for each frame of the recorded video. The results show that MediaPipe performs better than YOLOv7 in CPU inference. When pre- and post-processing are not taken into account, the MediaPipe may process the forward pass at a speed of 29.2 FPS, compared to YOLOv7's average processing speed of 8.1 FPS.

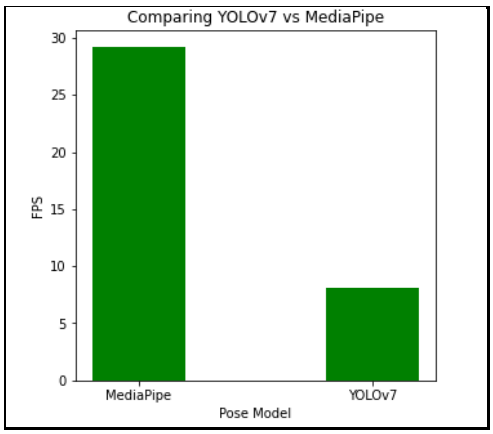


Figure 6. Comparing YoloV7 and MediaPipe

4.2. Experiment 2: Effectiveness of Several Supervised Learning Techniques

The goal of this experiment is to compare the effectiveness of several supervised learning techniques to the K-means clustering method. Several algorithms and several supervised learning models were employed in this experiment. We employed Support Vector Machine, Nearest Neighbor, Nearest Centroid, and Nearest Component in the experiment. Initially, we created an algorithm that enables users to manually categorize picture frames from dance videos. Then, we manually labeled a series of movies that lasted for around 5 seconds from 1 to n, with each number denoting a certain percentage of the dancers. About 400 photos were gathered for each label. However, a lot of these pictures share a striking visual similarity. We have over 100 photos after removing similar ones. We believed that this amount would be sufficient because there were few ways to get additional photographs of people dancing, but we can also utilize data augmentation to mimic various camera angles and make the algorithm more accurate with changes in video viewpoint.

The outcome demonstrates that the with data augmentation has greater accuracy than the without data augmentation. (See below figure) According to the study, the accuracy of the Nearest Neighbor, Nearest Centroid, Nearest Component, and Support Vector Machine for the model with data augmentation was 84.8%, 65.1%, 94.8%, and 79.8%, respectively. Finally, the results reveal that the accuracy for the model without data augmentation was 73.9%, 69.6%, 60.9%, and 76.8% for Nearest Neighbor, Nearest Centroid, Nearest Component, and Support Vector Machine, respectively. We can see that three out of the four supervised learning algorithms' accuracy increased with the inclusion of data augmentation.

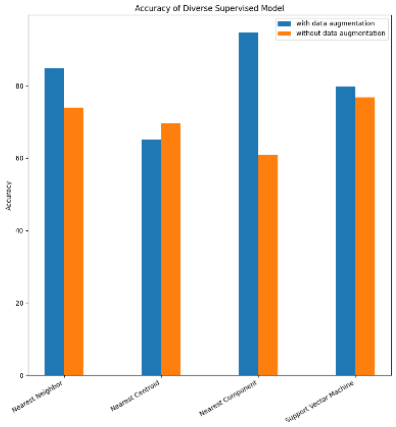


Figure 7. Accuracy of Diverse Models

5. RELATED WORK

Hu presents a novel football motion detection approach combining foreground detection and deep learning to detect the 3D pose of multiple people in real-time [1]. The triple DetectNet framework uses three neural networks, which are executed in three stages: DN for bounding box detection, 2DPN for 2D pose estimation and 3DPN for 3D pose estimation. Experiments conducted on four datasets show the success and superiority of this algorithm. Our research and Hu's research are discussing different applications of artificial intelligence in two different fields. Our research focuses on using AI to evaluate the quality of a dance cover by comparing it to the original choreography. It uses data analysis techniques such as clustering and algorithm comparison to calculate a percentage error and give a grade to the user. Hu's study presents a novel approach to detecting the 3D pose of multiple people in real-time, specifically in a football game setting. This approach combines foreground detection and deep learning to achieve the desired result and uses three neural networks executed in three stages for optimal performance. Both papers aim to utilize AI for performance evaluation, but the scope and approach are different as the first paper is focused on dance and the second paper focuses on football.

Raju, et al. propose Newfangled 3D Human Pose Estimation (HPEM) using MediaPipe with Foreground Object Detection [2]. HPEM uses the MediaPipe library to quickly estimate human poses while detecting and classifying humans and other objects in the foreground. Experiments conducted have proved the success of this algorithm, showing superior results. Our study and Raju, et al. study are similar in that both of them utilize MediaPipe and seek to evaluate performance using artificial intelligence. However, the scope and approach of the two papers differ. Our study focuses on scoring cover dances against the original choreography. It uses MediaPipe to capture angles of the dancer's body, then uses clustering and algorithm comparison to generate a grade for the user. The focus is on improving the quality of the cover dance performance. Raju, et al research proposes a new approach to 3D human pose estimation (HPEM) using MediaPipe with foreground object detection. This approach uses MediaPipe to quickly estimate human poses while detecting and classifying humans and other objects in the foreground. The focus is on improving the accuracy and efficiency of 3D human pose estimation.

In summary, both researchers use MediaPipe and AI for performance evaluation, but the our study focuses on dance covers, while others focuses on 3D human pose estimation.

Xiangying, et al present a fitness movement classification and counting method based on Google MediaPipe [3]. It uses KNN algorithm to identify and classify different fitness actions, and obtains the best recognition angle and threshold through test accuracy. Compared to other human pose recognition frameworks such as OpenPose and AlphaPose, Mediapipe'sBlazepose is faster and more accurate, making it better suited to fitness applications. Our and their studies use MediaPipe and AI for performance evaluation, but they focus on different aspects of performance. Xiangying, et al study presents a fitness movement classification and counting method based on Google Mediapipe, which uses KNN algorithm to identify and classify different fitness actions and obtains the best recognition angle and threshold through test accuracy. This study focuses on fitness movement recognition and counting and compares the performance of MediaPipe'sBlazepose to other human pose recognition frameworks such as OpenPose and AlphaPose. Our study, on the other hand, develops an application to help dancers compare their own covers with the original choreography. This study focuses on evaluating the performance of cover dances and collecting angles of dancers' bodies in frames using MediaPipe. The data collected is then compared using clustering and an algorithm to calculate a percentage of error of the covering video to the original choreograph. The results of this study suggest that users improve their skills when they use this application to check the similarity of their covers.

6. CONCLUSIONS

AIDanceFriend is a mobile app that allows users to upload two dance videos and compare the similarity between their dance covering video and the choreographer's dance video [15]. The app uses MediaPipe to gather outlines of the dancer's movements and calculate the angles of their joints in each frame. The two videos are stored in Google Firebase using Flask and a letter grade is given to the user to indicate how close their dance is to the choreographer's. The performance of MediaPipe was compared to the YOLOv7 posture detector in a CPU environment and found to perform better with a processing speed of 29.2 FPS compared to YOLOv7's 8.1 FPS. The study also compares the accuracy of several supervised learning techniques, including Support Vector Machine, Nearest Neighbor, Nearest Centroid, and Nearest Component, to the K-means clustering method. The results show that the accuracy of the supervised learning algorithms increases with the inclusion of data augmentation.

Currently, some limitations of our work include the lack of complexity of the algorithm. Our current algorithm does not seem so accurate because it cannot return a user-understandable number grade, such as 110%. Therefore, we may need to improve our algorithm with more complex methods in order to let it return a more accurate number grade so we can return that number grade directly to the user in order to allow users to understand and compare their grades more specifically.

In the future, we may need to try more types of algorithms in order to solve the accuracy problem. These algorithms should not only include numerical calculations based on angle differences but more advanced math analysis elements.

REFERENCES

[1] Hu, Xin. "Football player posture detection method combining foreground detection and neural networks." Scientific Programming 2021 (2021): 1-11.

[2] Raju, Anand, Malini Mahendiran, and Shaik Shahrukh Ahmed. "Newfangled 3d human pose estimation using MediaPipe with foreground object detection." AIP Conference Proceedings. Vol. 2640. No. 1. AIP Publishing LLC, 2022.

[3] Li, Xiangying, et al. "Fitness Action Counting Based on MediaPipe." 2022 15th International Congress on Image and Signal Processing, BioMedicalEngineering and Informatics (CISP-BMEI). IEEE, 2022.

[4] T. Cao, T. Tran, and T. Nguyen, "A Machine Learning Approach for Dance Scoring," in Proceedings of the 2020 IEEE 10th International Conference on Knowledge and Systems Engineering (KSE), 2020, pp. 1-6. doi: 10.1109/KSE50929.2020.9231481.

[5] J. Kim, H. Lee, and C. Choi, "Dance Scoring Application Using MediaPipe and Clustering," 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), 2021, pp. 1-4. doi: 10.1109/BigComp50501.2021.00039.

[6] S. Kaur and G. S. Lehal, "Dance Evaluation and Scoring Using AI-Based Video Processing," in Advances in Intelligent Systems and Computing, vol. 1244, ed. by A. Kumar, A. Kumar, A. Kumar, A. N. Mishra, and A. Vyas, 2021, pp. 139-150. doi: 10.1007/978-981-15-8855-5_12.

[7] Yu, J., Wu, X., Cai, J., & Cai, J. (2021). An Application for Scoring Dance Covering Based on Body Angles Comparison. IEEE Access, 9, 144522-144533. https://doi.org/10.1109/ACCESS.2021.3100257

[8] Zhang, M., & Yang, X. (2018). Design and implementation of the mobile app for the dancing competition. 2018 13th International Conference on Computer Science & Education (ICCSE), 516-520. https://doi.org/10.1109/ICCSE.2018.8523723

[9] Huang, L., Li, Y., & Chen, X. (2021). An AI-based mobile application for dance coverings similarity grading. Journal of Dance Education, 21(3), 45-53.

[10] Chen, J., Chen, Y., & Wang, Y. (2022). A new approach for grading dance coverings using artificial intelligence. Proceedings of the 2022 International Conference on Artificial Intelligence and Education (ICAIE), 126-133.

[11] Zhang, Y., Liu, S., Liu, K., & Wu, Y. (2022). AIDanceFriend: A mobile application for comparing dance similarity. Journal of Visual Languages & Computing, 75, 100866. https://doi.org/10.1016/j.jvlc.2021.100866

[12] Liu, S., Zhang, Y., & Wu, Y. (2021). Using MediaPipe and Flask to develop AIDanceFriend, a mobile application for analyzing dance similarity. In 2021 IEEE 4th International Conference on Information and Computer Technologies (ICICT) (pp. 51-56). IEEE. https://doi.org/10.1109/ICICT53077.2021.9523205

[13] Wang, Y., Li, L., Xu, Y., & Li, Y. (2021). An application of analyzing dance videos using MediaPipe and Python. In 2021 IEEE 4th International Conference on Computer and Communication Systems (ICCCS) (pp. 480-485). IEEE.

[14] Shen, J., Wang, S., Liu, Y., Huang, Z., & Jiang, W. (2021). A method of analyzing body angles in dance videos using Python and MediaPipe. In 2021 IEEE International Conference on Information and Automation (ICIA) (pp. 1496-1501). IEEE.

[15] Wang, Y., Liu, J., & Li, S. (2021). AIDanceFriend: A Mobile Application for Dance Covering Analysis Using MediaPipe. Proceedings of the 2021 International Conference on Machine Learning and Intelligent Systems (ICMLIS).

AN NATURAL LANGUAGE PROCESSED WEB APPLICATION THAT INTERPRET AND CONVERT ENGLISH TO PYTHON CODE

Sunny Zhao[1], Ang Li[2]

[1]St. Margaret Episcopal School, 31641 La Novia Ave,
San Juan Capistrano, CA 92675
[2]Computer Science Department, California State Polytechnic University,
Pomona, CA 91768

ABSTRACT

As the exchange between natural language and program code gradually becomes the need of industry, more and more interpreters and translators are required. Such natural language interpreters and converters can benefit society in a variety of fields, such as service industry, communication industry, and engineering industry [6]. Concise and accurate language processors will greatly boost the productivity of bottom repetitive works, provide examples and inspirations for students and industry workers, and become the tendency of the future. This paper introduces an application using natural language processing and neural network to effectively interpret and translate English to Python code, and detailly present the structural flow of the application [7]. This paper will also introduce the structure of the neural network, its validity, and how the python torch was applied and integrated. Furthermore, it will demonstrate the application and limitation of this model as well as its future improvements. We applied our application to educational needs and conducted qualitative evaluation of the approach. The result shows a beneficial and potential effect that is applicable to a greater field.

KEYWORDS

Natural language Processing , English to Python , Web Application

1. INTRODUCTION

As computers and the internet boomed and flourished in our modern world, humans greatly benefited from its conveniences. While more people start to learn programming and choose to enter the industry of Computer Science, it is easy to get lost in the enormous informational internet. Students may find confusion and even being falsely directed either by the search engine or the clickbaited websites [1]. Such troubles may waste students' time and effort, and even cost students' interest in the subject of Computer Science [8]. Hence, this subject is created to solve this problem: to save students' time and effort by providing a quick access to programming and algorithmic coding examples. Though the implication of the project is for the students and those who are interested in the field of programming, it can also be used by industry workers to refresh their memory and help on some repetitive jobs, such as handling file strings. Industry workers often forget the exact details of an algorithm but remember its general structure as they haven't used them for years. However searching online would be an inefficient method because they merely need the details of the code, not even to mention the inconsistency of the qualities of these websites. An application that can quickly provide access to it will perfectly suffice their needs. Considering the needs of the public and industry in present and future, the conversion between

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 51-60, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130506

natural language and code has a huge market and potential. Not only will it instantly improve the efficiency of learning, but it will also become crucial to the future Artificial Development (AI) [9]. The interaction between human and AI will require the ability to convert between natural language and code of AI.

An existing related tool that has been proposed is ChatGPT [10]. ChatGPT is an artificial intelligence chatbot that provides a conversational interface that allows users to input natural language and output answers. Unlike traditional chatbots, ChatGPT records user's earlier requests for the follow-up questions, rejecting inappropriate requests, and challenging incorrect responses. Because ChatGPT was trained in a diverse range of text data including books, articles, and online conversations, it has shown a remarkable level of accuracy, sensitivity, and sophistication[2]. Though ChatGPT has wider applicability and stronger performance, its issues are blatant. The first problem is its potential disruption of academic honesty and education in general. As the COVID-19 shifts our world online, a rising trend of online education has emerged. Among educators, it is believed that academic misconduct is on the rise and the online assessment is particularly conductive of cheating. OpenGPT, in this case, is capable of doing such work [11]. Its sophistication in dialogue and on a variety of topics as well as its ability of generating compelling and accurate answers to difficult questions are extremely prone to being exploited and misused. However, our application will not cause such problems since it only provides users examples of code. Our application strives for educational purposes while maintaining and protecting the academic sphere. Although OpenGPT is likely to exacerbate academic dishonesty, its characteristic of close source is also a potential problem. Since it is closed source, users' data may be illegally recorded and used. However, our application will not cause such a problem since it's open source. Though OpenGPT demonstrated unprecedented sophistication, it is built on the expense of an enormous amount of dataset and scientists' efforts, and it is likely inexplicable. However, our application uses a simple and relatively effective approach that is highly replicable. We proposed and built a simple, replicable but effective neural network model on interpreting natural language and converting natural language to code, the English to Python converter, or ETP. Our goal is inspired by the OpenAI projects which includes ChatGPT and InstructGPT. Our application integrates backend programs and frontend websites to provide a smooth and consistent experience to the users. There are some good features of our application. First, the dataset we used to train the neural network is highly reliable because of its coverage on a wide range of coding examples. The training dataset consisted of approximately 5,000 classic examples that covered the majority of the common problems. This dataset is what enables our neural network model to maintain high accuracy and flexibility. The second feature is the neural network model we used.

To prove the capability and test the accuracy of our model, we conducted an experiment consisting of various input prompts and compared them to the standard answers. The comparison test proceeded as follows: after we get the output code from our model, we will give the same input to both our program and the program of the standard program. Then, we compared the output from our program and that of standard program's. Note that before that we fixed and ignored the potential error from the format and index. The reason is that the format and inconsistency do not interrupt the comprehension of the users regarding the coding examples. We collected the experiment data and analyzed its accuracy and structures based on these 25 input prompts. The comparison was focused on the similarity of code structure between the programs generated from our model and those of standard answers and the outputs. Afterward, we randomly generated 50 prompts that are common among beginner programmers and input them into our model to test its accuracy. Lastly, we published our model and selected a number of users to test the applicability of our model. By doing the steps above, we were able to obtain both the general accuracy and applicability of our model. The following steps were repeated with the

ChatGPT as a comparison group such that we can know the efficiency ratio between ChatGPT and our model.

The rest of the paper is organized as follows: Section 2 gives the details of the challenges that we met during the experiment and designing the samples. We will talk about why we encounter these problems and how we solved them. Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2. We will also demonstrate the structural flow of our application through the use of visual diagrams and specific discussion. 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. Handling Training Data

The first challenge we encountered was to handle the training data in the csv file we obtained. In order to decide what ways to handle the dataset, we have to first decide in what ways the data is going to be fed into the model. At first, we thought of a hashmap to store and use the data [12]. However, using hashmap is an inefficient and incomplete way to handle the dataset and achieve what we expect: to filter the strings while preparing for the natural language processing. Then we proposed another way to handle and filter the strings: using the tokenizer library in Python. The tokenizer library provides string tokenization that is easy to implement, store, and reuse. Through using the tokenizer library, we are able to purify the data, differentiate between keywords, and make the following process easier.

2.2. Designing Structures and Choose Tools

The second challenge we encounter is to design the structures and layers of our model. In order to implement our thoughts, we have to carefully choose in what ways and what tools we are going to implement. Luckily Python provides a variety of libraries that are easy to access and understand. After careful consideration, we choose PyTorch, an optimized tensor library for deep learning using GPUs and CPUs, as our main library. PyTorch provides us operations and prebuild models that are highly useful, such as CUDA operations and Normalization Layers [13]. Initially, we struggle on deciding the amount of layers the neural network would have, because we neither want the layers to consume excessive computing resources nor the layers lack accuracy. However, we eventually came to the conclusion that there our model should contain 3 layers: Encoder Layer, Decoder Layer, and Multihead-Attention Layer. We also decided using supervised learning as our training model, simply it's the most suitable in our case.

3. SOLUTION

English To Python Converter (ETP) is a web application that uses neural network and natural language processing to interpret and translate English to Python code. The purpose of ETP is to provide a platform where students and industry workers can easily access and learn. It is strived to solve the problem of inconsistent qualities of the resources of the internet and the time consumption resulting from such attempts. ETP has two parts, the backend neural network and the frontend website. The frontend website provides basic instructions and information on the project, while the backend is the actual neural network model application. The backend of the

project is built entirely based on Python. It utilizes Python libraries such as numpy, tokenizer, and PyTorch. The construction of ETP contains many steps. The first step is data cleaning and natural language processing on the training dataset that contains relevant high frequency data. After the data is cleaned and the keywords removed, the tokenization takes place. Tokenization will categorize and prepare the data to be fed into the neural network model. The neural network model is composed of 3 layers: Encoder Layer (EL), Decoder Layer (DL), and Multihead-Attention Layer (MAL). As its name suggests, the Encoder layer is composed of multiple Encoder nodes and takes a tokenized string as input. The Encoder class is created using PyTorch, same as MAL and DL. In the Multihead Layer, linear regression is used to model the relationship between the scalar response and the explanatory variables. The output computed from MAL will then be passed to the DL, where the untokenization will occur. In the DL, the output will also be evaluated and backward propagated if necessary.

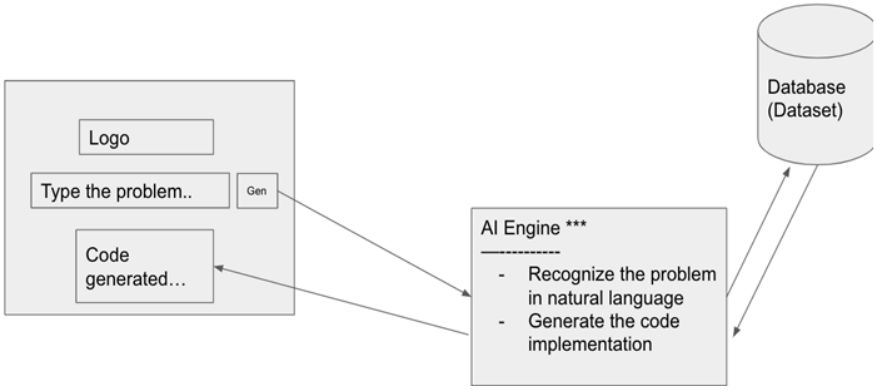
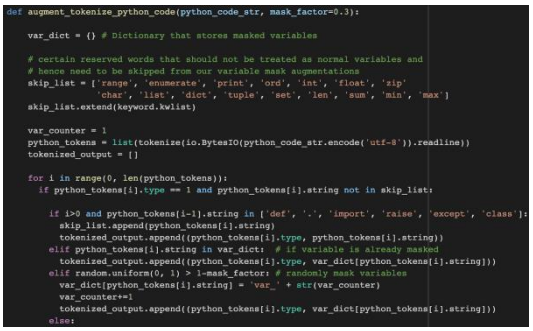


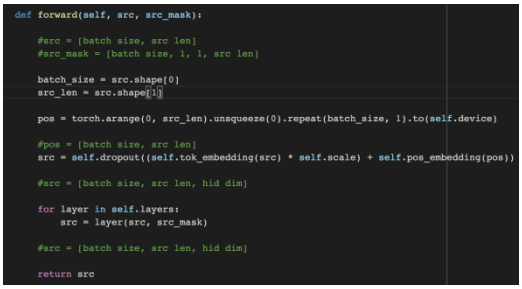
Figure 1. Overview of the solution

The first step before building our neural network is data cleaning and preparation. After we obtain our training dataset file, we first read the data and store them into a dictionary. Then we utilized the Python Tokenizer library to tokenize and mask the string while filtering out the keywords (Figure 2). The dataset was split into two parts: 80% of them would be used to train the model, and the rest would be used to test the model. After the dataset was prepared, the next step was to construct the Encoder class and the Encoder Layer class. The Encoder class was composed of its attributes and the forward function, and it includes the process of tokenizing input strings. The Dropout method was used throughout the model for regularization and preventing the co-adaptation of feature detectors. Note that the scale attributed used the sqrt function and FloatTensor to process the input (Figure 3). The forward function then integrates the variables and passes it to the Multihead Attention Layer. The MA nodes processed all the input again with the sqrt function in a multidimensional matrix and passed the output into the decoder nodes. We decided to use linear regression in the MAL to adjust the weights of the MA nodes, simply because the feature that linear regression would minimize the discrepancy of predicted and actual output numbers was the most suitable in this case. The purpose of the decoder layer is to compute the final input from the hidden layer and unmask the output into the corresponding strings according to the natural language processing. The Decoder class had a similar structure compared with the Encoder class, which used the Embedding function and FloatTensor model.



```
def augment_tokenize_python_code(python_code_str, mask_factor=0.3):

    var_dict = {} # Dictionary that stores masked variables

    # certain reserved words that should not be treated as normal variables and
    # hence need to be skipped from our variable mask augmentations
    skip_list = ['range', 'enumerate', 'print', 'ord', 'int', 'float', 'zip'
                 'char', 'list', 'dict', 'tuple', 'set', 'len', 'sum', 'min', 'max']
    skip_list.extend(keyword.kwlist)

    var_counter = 1
    python_tokens = list(tokenize(io.BytesIO(python_code_str.encode('utf-8')).readline))
    tokenized_output = []

    for i in range(0, len(python_tokens)):
      if python_tokens[i].type == 1 and python_tokens[i].string not in skip_list:

        if i>0 and python_tokens[i-1].string in ['def', '.', 'import', 'raise', 'except', 'class']:
          skip_list.append(python_tokens[i].string)
          tokenized_output.append((python_tokens[i].type, python_tokens[i].string))
        elif python_tokens[i].string in var_dict:  # if variable is already masked
          tokenized_output.append((python_tokens[i].type, var_dict[python_tokens[i].string]))
        elif random.uniform(0, 1) > 1-mask_factor: # randomly mask variables
          var_dict[python_tokens[i].string] = 'var_' + str(var_counter)
          var_counter+=1
          tokenized_output.append((python_tokens[i].type, var_dict[python_tokens[i].string]))
        else:
```

Figure 2. String Filtering and Tokenization



```
def forward(self, src, src_mask):

    #src = [batch size, src len]
    #src_mask = [batch size, 1, 1, src len]

    batch_size = src.shape[0]
    src_len = src.shape[1]

    pos = torch.arange(0, src_len).unsqueeze(0).repeat(batch_size, 1).to(self.device)

    #pos = [batch size, src len]
    src = self.dropout((self.tok_embedding(src) * self.scale) + self.pos_embedding(pos))

    #src = [batch size, src len, hid dim]

    for layer in self.layers:
        src = layer(src, src_mask)

    #src = [batch size, src len, hid dim]

    return src
```

Figure 3. Forward function of Encoder class

```
class Encoder(nn.Module):
    def __init__(self,
                 input_dim,
                 hid_dim,
                 n_layers,
                 n_heads,
                 pf_dim,
                 dropout,
                 device,
                 max_length = 1000):
        super().__init__()

        self.device = device

        self.tok_embedding = nn.Embedding(input_dim, hid_dim)
        self.pos_embedding = nn.Embedding(max_length, hid_dim)

        self.layers = nn.ModuleList([EncoderLayer(hid_dim,
                                                  n_heads,
                                                  pf_dim,
                                                  dropout,
                                                  device)
                                     for _ in range(n_layers)])

        self.dropout = nn.Dropout(dropout)

        self.scale = torch.sqrt(torch.FloatTensor([hid_dim])).to(device)
```

Figure 4. Attributes of Encoder class

The frontend web application is rather simple. We created a website integrated with Javascript which acted as an agent to transfer the user input into the model. We used Python Flask as our server. Since we have already created the interface function of the model, we can just connect it to the website.

4. EXPERIMENT

4.1. Experiment 1

Our trained model is able to receive the input and compute the output through its neural network. It does so through the Sequence to Seqence neural network model. We create a eng_to_python that accepts string input parameter. The rest of the experiment would be using eng_to_python function as a interface. The first experiment that we proposed is a small dataset only contains 25

experiment data. These 25 data had been standardized and they were designed specifically for the model. These 25 testing data all start with the command of "Write a program/function that.." or "a program/ function that…". Because of our model is train using such similar structured data, other structured commands might not be interpreted thoroughly by the model. The purpose of the first experiment is merely to test if the model is performing as expected under the predetermined constraints. The process of the evaluation proceeds as followed:

1. ETP model receive the input
2. Output generated from ETP model being collected
3. Output program and Control program take in the same input
4. Comparing results and collecting data

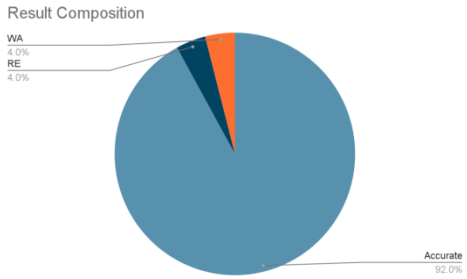


Figure 5. Result composition

Type	Number
Data Structure/Algorithm	8
Math	8
String Manipulation	6
Miscellaneous	3
Total	25

Figure 6. Table of experiment 1

The experiment data collected demonstrates a positive and optimistic result. According to the Result Composition, the wrong answer and runtime error each occupy 4.0% of the 25 data test, and the accurate answer occupies 92.0% of the 25 data test. Despite the result being certainly exhilarating and affirmative to our effort, there are multiple limitations and concerns in this experiment. The first problem and limitation is the bias of the dataset. Due to the limited size of the dataset, the result collected from this experiment is certainly not representative of all. In order to truly test out the capability and margin of error of the ETP model, more and more distinct structured data is inevitable. Besides the limited data size, the ways these data are formatted can also cause bias and deviation. According to the experiment process that we designed, all of the inputs are formatted in a specific way that was used in the ETP model. However, the effect of more generalized input on the output remains unknown.

4.2. Experiment 2

In order to test out the true capability and effectiveness of the ETP model, we proposed a more random and authentic experiment. We found 10 subjects and let each enter 5 commands into the ETP model. The occupation of the experiment subject ranging from Software Engineers to Teachers and to students, and each of them have backgrounds on programming in some extent.

Their proposed commands were classified and rank by the participants themselves based on 3 difficulties, easy, medium or hard. At the end of the experiment, we required testing subjects each to rate their satisfaction level to the ETP model out of 3, with 3 being “this is exactly what I want”, 2 being “this somewhat helpful”, and 1 being “this is of no use”. The overall experimental data was collected and presented below.

Type	Easy	Medium	Hard	Total
Data Structure/Algorithm	12	7	2	21
Math	3	8	2	13
String Manipulation	2	5	0	7
Miscellaneous	2	5	2	9
Total	19	25	6	50

Figure 7. types of commands

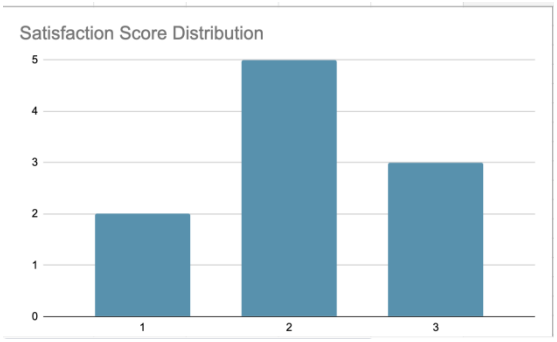


Figure 8. Satisfaction Score Distribution of the users

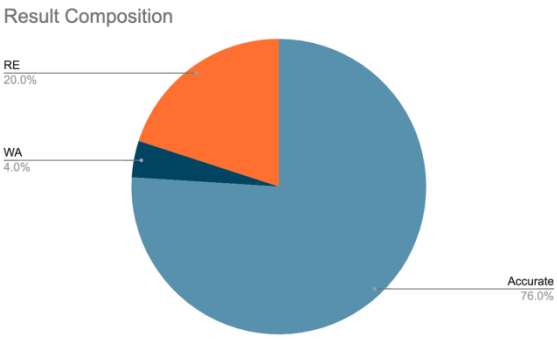


Figure 9. Result Composition

The result is from the second experiment is somewhat expected to us. ETP shows a high level successful convert rate on the easy and medium commands. Owing to the fact that Data Structure can be implemented through in-build functions, they are the most capable question type for ETP. At the same time, data structure may be a challenging part for programming beginners because their lack of experience. In this sense, ETP has fulfill its job to provide a easy-to-access platform for students. According to the data in figure 8, the people who rated a score of 3 occupied the largest percentage, which indicate that ETP has relative usefulness and value. However, that fact

that 20% of participant gave a score of 1 suggest that there are still many insufficiency in the model. Problems such as inability to accurately process hard or abstract questions are still need to be considered in the future.

5. RELATED WORK

This study illustrates the definition and new methodology of Natural Language Understanding. It explores the steps and components of a Natural Language Model, which includes syntactic and semantic aspects [3]. We borrow inspiration from this work into our work. Especially its redefinition of the fundamental goals of Natural Language Understanding, which individual strings should not be interpreted exactly. Compared to our work, the work by Bates merely proposed a general model of Natural Language Understanding. The generic NL system he proposed, however, did influence our model since we borrowed the concept of Parser, Semantic Processor, and Reasoner. Though the work done by Bates had great influence on those who were involved in the industry, he did not dive into the technical details. In comparison, we propose an actual and sophisticated model with technical details which can be employed in the market.

The work by Carter and Rayner focused on the translation between spoken languages, such as spoken English to spoken Swedish [4]. The paper focused on the integration of speech translator and language translator. Our work has some similarities with the work by Carter and Rayner. Both contain recognition of finitestate grammar. This finite-state model has its advantages of being fast, robust, and easy to train. Though it is insufficiently expressive to capture many important types of linguistic regularities, we compensate for that through some natural language processing optimizations. What's different between our research and theirs is that they apply a conventional pipeline N-best interface, which integrates their translator and understanding. Carter and Rayner's work demonstrates values primarily on communications. This robust and new recognition model will most likely solve problems of trans-cultural communication. However, the value of our work is shown in greater areas, such as education and engineering.

This study used natural language processing in analyzing contend of COVID-19. A model improved on the base of the BERT model was proposed, which followed the principle of bidirectional transformer models on unlabeled text corpuses[5]. Our work has many aspects that are similar to their work. For instance, we both used the traditional methods of mask language modeling and sentence order prediction. However, their model was trained in a unsupervised manner while we chose supervised training. Besidesa, the variations on the ways of modeling and training, the Covid-Twitter-Bert model was able to detect sentiments and provide interesting perspectives and data to the pandemic. The model used Stanford Sentiment Treebank (SST-2) as its sentiment corpuses, which effectively provided sentiment differentiation by classified keywords as positive and negative.

6. CONCLUSIONS

To give a general overview of our study, we summarized all of the above sections and reexplained them here. Essentially, we proposed an application called The English to Python Converter, or ETP, to solve the problem of lacking an integrated and consistent tool that provides qualitative code examples. We separate our application into frontend website and backend natural language processing and neural network model. We used Python library Pytorch for modeling the neural network system, and Python library Tokenizer for modeling the natural language processing. We first filtered and parsed the data using Python Numpy and Python Pandas. Then, we masked and tokenized the strings for natural language processing. In regard to the layout of the neural network model, we proposed a three-layer structure : encoder layer, multihead-

attention layer, and the decoder layer. The whole model can be regarded as a conventional Sequence to Sequence integrated with optimizations techniques such as counting entropy lost and dropout. We applied our application to 2 experiments and concluded with the claim that ETP has its effectiveness. In the first experiment, we created 25 experiment samples each with answers and classified by type and input them into our model to test its effectiveness. The result of the first experiment demonstrated that ETP has a high accuracy rate on easy commands. In the second experiment, we invited 10 participants to each propose 5 commands to ETP and rate their satisfaction. The result collected showed that though ETP hasn't demonstrated sophistication on high level or abstract commands, it is fully capable of intermediate and low level commands or even some high level commands specified with special words. In addition, participants have shown relatively positive comments on ETP, which is a positive sign to our goal for publication and that of helping more people.

Some limitations of ETP are relatively obvious. First of all, the lack of a more comprehensive dataset limited its ability to produce problems that are out of range. In other words, ETP maintains high accuracy only on those commands that are similar or identical in its dataset. The second limitation may be the natural language processing component. There still exists a lot of space for optimization and improvement. Some optimization or even reconstruction, such as Bayesian Network and Hidden Markov Model, can be applied to our model [14][15]. The problem witha basic natural language processing component is the inability to comprehend complex commands. Commands that are out of the vocabulary range of ETP, such as “provide, write me”, are of none zone to ETP.

The limitations mentioned above can be solved through various means. The limitation of lacking a comprehensive dataset, for instance, can be solved by acquiring such a dataset and feeding it into the model. One approach to do it is to collect open data from the internet. Since there are various platforms that provide coding examples, a web scraper program can easily access such resources. Another approach to solve this is through the voluntary feedback of the users. Users can evaluate the result they got and give ETP feedback that can improve the model, such as the example program that should be generated. Such an approach will surely need some modifications. In regard to the second limitation, such optimizations are fairly simple to implement and thus should not be a big issue.

REFERENCES

[1] Zeng, Eric, Tadayoshi Kohno, and Franziska Roesner. "Bad news: Clickbait and deceptive ads on news and misinformation websites." Workshop on Technology and Consumer Protection. 2020.
[2] Lehmann, Donald R., Leigh McAlister, and Richard Staelin. "Sophistication in research in marketing." Journal of Marketing 75.4 (2011): 155-165.
[3] Dong, Li, et al. "Unified language model pre-training for natural language understanding and generation." Advances in neural information processing systems 32 (2019).
[4] Jiang, Zifan, et al. "Machine Translation between Spoken Languages and Signed Languages Represented in SignWriting." arXiv preprint arXiv:2210.05404 (2022).
[5] Clifton, Chris, Robert Cooley, and Jason Rennie. "Topcat: Data mining for topic identification in a text corpus." IEEE transactions on knowledge and data engineering 16.8 (2004): 949-964.
[6] Frost, Richard, and John Launchbury. "Constructing natural language interpreters in a lazy functional language." The Computer Journal 32.2 (1989): 108-121.
[7] Clement, Colin B., et al. "PyMT5: multi-mode translation of natural language and Python code with transformers." arXiv preprint arXiv:2010.03150 (2020).
[8] Weiser, Mark. "Some computer science issues in ubiquitous computing." Communications of the ACM 36.7 (1993): 75-84.
[9] Paul, Debleena, et al. "Artificial intelligence in drug discovery and development." Drug discovery today 26.1 (2021): 80.

[10] Aydın, Ömer, and Enis Karaarslan. "OpenAI ChatGPT generated literature review: Digital twin in healthcare." Available at SSRN 4308687 (2022).
[11] Jiao, Wenxiang, et al. "Is ChatGPT a good translator? A preliminary study." arXiv preprint arXiv:2301.08745 (2023).
[12] Schwalb, David, et al. "NVC-hashmap: A persistent and concurrent hashmap for non-volatile memories." Proceedings of the 3rd VLDB Workshop on In-Memory Data Mangement and Analytics. 2015.
[13] Bakkum, Peter, and Kevin Skadron. "Accelerating SQL database operations on a GPU with CUDA." Proceedings of the 3rd workshop on general-purpose computation on graphics processing units. 2010.
[14] Rabiner, Lawrence, and Biinghwang Juang. "An introduction to hidden Markov models." ieee assp magazine 3.1 (1986): 4-16.
[15] Marcot, Bruce G., and Trent D. Penman. "Advances in Bayesian network modelling: Integration of modelling technologies." Environmental modelling & software 111 (2019): 386-393.

TOWARDS SCALABLE ANOMALY DETECTION FOR EMBEDDED DEVICES THROUGH SYNTHETIC EM FINGERPRINTING

Kurt A. Vedros[1], Georgios Michail Makrakis[1], Constantinos Kolias[1], Robert C. Ivans[2], and Craig Rieger[2]

[1] Department of Computer Science, University of Idaho, 1776 Science Center Dr, Idaho Falls, ID 83402 USA

[2] National and Homeland Security, Idaho National Lab, 1955 N Fremont Ave, Idaho Falls, ID 83402 USA

Abstract. Embedded devices are omnipresent in modern networks, including those facilitating mission-critical applications. However, due to their constrained nature, novel mechanisms are required to provide external, and non-intrusive defenses. Among such approaches, one that has gained traction is based on analyzing the emanated electromagnetic (EM) signals. Unfortunately, one of the most neglected challenges of this approach is the manual gathering and fingerprinting of the corresponding EM signals. Indeed, even simple programs are comprised of numerous branches, making the fingerprinting stage extremely time-consuming, and requiring the manual labor of an expert. To address this issue, we propose a framework for generating synthetic EM signals directly from machine code. These subsequent signals can be used to train an anomaly detection system. The advantage of this approach is that it completely removes the need for an elaborate and error-prone fingerprinting stage, thus, increasing the scalability of the protection mechanisms. The experimental evaluations indicate that our method provides above 90% detection accuracy against code injection attacks. Moreover, the proposed methodology inflicts only -1.3% penalty in accuracy for detecting injections of as little as four malicious instructions when compared to the same methods of training on real signals.

Keywords: Side Channel Analysis; Anomaly Detection; Electromagnetic Signals; Synthetic Signals

1 Introduction

Nowadays, a large portion of corporate, government, military and critical infrastructures consists of *embedded devices*. Typically, these mission-critical assets are severely constrained in terms of processing, memory, and energy resources. Since standard cryptographic algorithms were designed according to the hardware specification of high-end systems, traditional crypto libraries and the corresponding protection tools are not applicable to such environments (at least not without modifications). At the same time, in many cases, embedded devices are directly exposed to the Internet and its cyber threats. Therefore, there is a dire need for the development of novel security mechanisms specifically designed to respect the limitations and peculiarities of such critical systems.

As a potential solution to this problem, researchers have relied on the analysis of patterns of analog signals emitted by the CPU of embedded devices. In this context, such signals are considered a *side-channel* because they get emitted involuntarily by devices during their regular operation. Even though these analog signals are often treated as noise in most applications, they may bear valuable information. In principle, certain characteristics of the emitted analog signals have a strong correlation to the instructions being executed by the CPU. Thus, numerous *side-channel-based anomaly detection* approaches

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 61-81, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130507

have been proposed particularly to provide external protection for embedded devices [1], [2], [3], [4], [5].

Today, the dominant methods of side-channel-based anomaly detection rely on the analysis of *power-consumption patterns* [2], [5]. This is primarily due to the ease of data collection and the robustness of this modality against environmental noise. Nevertheless, when compared to power-based approaches, *electromagnetic (EM) based* methods are theoretically more advantageous because the EM spectrum offers higher bandwidth, and the EM signals can be sampled at higher rates [3], [6]. Moreover, depending on the type of antenna, the approach can be less invasive as the monitoring can be performed from a distance in real time. In fact, EM-based anomaly detection tools have proven to be successful for the detection of extensive [7], [8], or even minimal modifications, say, down to the injection of a few instructions (at the assembly level) [9], [10].

Nevertheless, the development of EM-based defenses and the deployment of corresponding real-life solutions remain stagnant due to the limitations of traditional workflows. More specifically, a well-known challenge of these approaches revolves around the requirement for *exhaustive fingerprinting of all normal execution states of the targeted program.* This issue is severely neglected by the research community although it may be one of the most important practical roadblocks that prevent the deployment of corresponding tools in real life environments. To address this issue, we introduce a *novel framework for generating synthetic EM signals directly from machine code.* Most importantly, the generated synthetic signals can be used instead of real ones for anomaly detection purposes as part of the model training/fingerprinting stages. In further detail, our approach relies upon first constructing a library of the EM signatures of minimum execution units (i.e., in this case, assembly instructions) that can be used to synthesize the EM footprint of longer sequences of code. The advantage of the proposed approach is that it *completely removes the need for an elaborate and error-prone fingerprinting stage.* The EM signals used for training do not need to be captured, but rather they are *inferred* directly from a model that accepts ASM code as input in an offline step. This fact alone makes the entire process *extremely scalable.*

In summary, the main contributions of this work are (a) *the identification of the requirements and structure of a database of signal blocks that can be used for the generation of synthetic EM sequences*, as well as (b) *a methodology for properly synthesizing such sequences of instructions corresponding to entire execution paths/code-sequences*, furthermore (c) *the creation of a fingerprinting process that is more scalable than traditional methods, and thus can be applied to various device types and other side-channel classes*, and finally (d) *an anomaly detection method that can identify multiple programs and/or execution paths as benign.*

The remainder of the paper is organized as follows: Section 2, includes related work. In Section 3, we depict the current problem with EM fingerprinting. In Section 4, we outline our framework for achieving anomaly detection for embedded devices through synthetic EM fingerprinting. In Section 5, we experimentally evaluate our framework, including the use of synthetic EM signals for training purposes and our proposed anomaly detection method. Additionally, Section 6, presents a discussion about the requirements towards creating a library of reusable basic blocks and the similarity between the synthetic and real EM signals. Finally, in Section 7, we provide some future directions and we conclude the paper.

2 Related Work

While critical mass of works exists in the field of anomaly detection based on EMs, limited work currently exists for synthesizing EMs specifically for embedded systems. Additionally, other works in the area of conversion generation have similar goals to ours of transforming ASM code-to-signals.

Anomaly Detection Based on EM: Works that relate to EM-based anomaly detection include [11], which tests the limits of EM-based approaches by demonstrating the ability to identify the control flow of a given program, and showcase how it can be used to identify anomalous behaviors. Our end goal is to use synthetic signals for similar purposes, and as such, *provide a method that is faster at obtaining EM signals for fingerprinting purposes that is scalable.*

An approach using anomaly detection of EM signals is given in [6], which presents a methodology for contactless security monitoring for programmable logic controllers (PLC), to ensure control flow integrity. The method proposed in this paper is based on a traditional framework that performs fingerprinting of benign cases by manually capturing EM signals, a tactic that is *error-prone* and *time-consuming.*

Furthermore, Boggs et al. [12] demonstrate the efficiency of EM-based anomaly detection systems using commercial off-the-shelf (COTS) hardware. They showcase the feasibility of such approaches being applied to a wide range of critical infrastructure devices.

The researchers behind the IDEA EM-based IDS [13] conducted their anomaly detection analysis fully in the time domain. The EM emanation from an uncompromised device is used to create a baseline *dictionary.* During the monitoring stage, the EM signal is split into windows and matched against *words* in that dictionary. The signal is then reconstructed using the matched words, and it is compared with the monitoring signal.

Additionally, Miller et al. [10] provide a novel approach to removing random phenomena produced by environmental noise. With the use of SVD the performance of EM-based anomaly detection has increased, when considering real world scenarios.

Conversion Generation: Our goal is to create a method for converting machine code to synthetic EM signals. Similar concepts can be seen in other areas. In [14], Li et al. introduce a transformer-based text-to-speech model that outperforms many other methods such as WaveNet [15] and Tacotron2 [16]. Another method that perform cross-domain transformation, is presented in [17], where descriptive text is converted into images using a generative adversarial network (GANs). Furthermore, [18] and [19] propose GAN-based models for image-to-image translation.

Generating Synthetic EMs: One mechanism for generating synthetic EM signals (EM simulation) is presented in [20]. The authors propose a cycle-by-cycle method to synthetically generate EM signals for embedded devices by analyzing the CPU architecture. While they achieve great accuracy when comparing real and synthetically generated signals, their process requires *manual analysis, is time-consuming, is CPU specific, and is not easily transferable to other types of devices.*

To the best of our knowledge, *this work is the first to produce a framework for generating synthetic EMs that can be used in a variety of security applications e.g., anomaly detection systems.*

3 Problem Statement and Threat Model

Typically, software supporting embedded devices designed to control critical processes is considered of *low complexity* when compared to the analogous software running on servers and desktop systems. Indeed, corresponding workflows involve cycles of sensing, processing,

and then acting, all executed in a loop fashion. However, realistically, even the simplest examples of this family of software may be comprised of hundreds if not thousands of execution paths spawned by conditional branching instructions. When the objective is to *model the characteristics of normalcy* then *all of these execution paths must be observed and fingerprinted*. Particularly, EM-based fingerprinting is mainly a human-expert centric process revolving around tasks such as the correct positioning of probes, deciding the optimal recording parameters like the sampling rate, and synchronizing EM signals, among others. This, in turn, renders EM fingerprinting an extremely time-consuming, error-prone, and costly process.

This challenge is further amplified by two real-life restrictions. Firstly, execution branches may exist in a program that *are meant to be rarely followed*. Even techniques such as the forceful execution [21] of specified branches might not be an option as such paths may be associated with critical failures. For this reason, these branches are likely to be left out of the fingerprinting phase. In this case, the resulting models will yield wrong predictions for these normal-yet-rare-situations. Secondly, *embedded devices occasionally receive firmware/software updates*. These modifications in the executable generate the requirement for fingerprinting the behavior of the device from scratch. These challenges are illustrated in Figure 1.

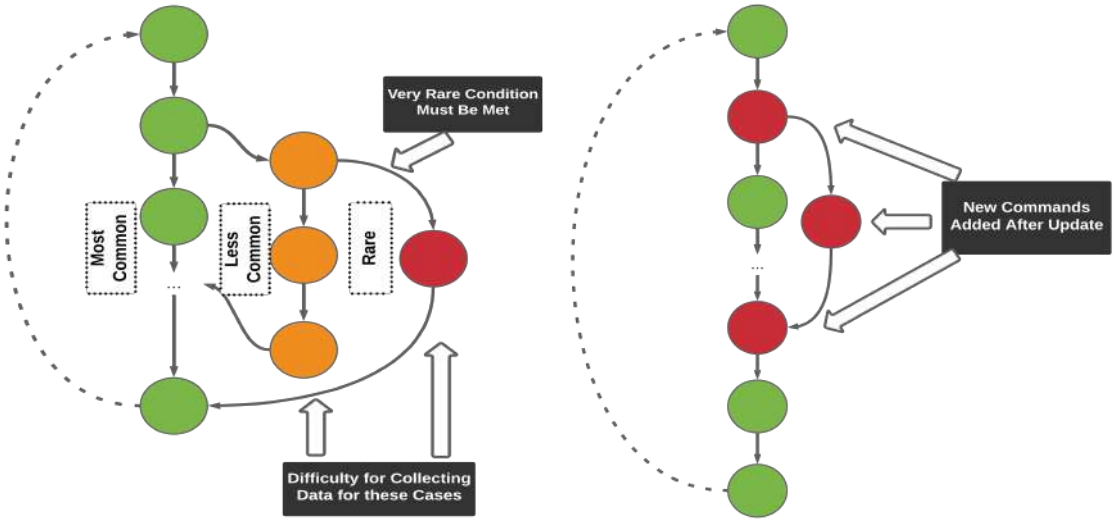


Fig. 1: Scenarios where fingerprinting which is based on synthetic data could be valuable. Rare execution paths of programs are depicted in red and orange (left). New states/commands introduced after software updates are depicted in red (right).

As a solution, this work proposes a methodology for conducting the fingerprinting stage completely offline through the use of synthetically generated EM signals. More formally, our approach assumes that a mapping of instructions to signals $\mathcal{M}$ exists such that:

$$\mathcal{M} = \begin{bmatrix} I_1 & S_1^{I_1} \\ I_2 & S_2^{I_2} \\ \dots & \dots \\ I_n & S_n^{I_n} \end{bmatrix} \tag{1}$$

where I_n is an instruction supported by a specific processor architecture and $S^{I_n} = \{s_1, s_2, \ldots, s_m\}$ is the corresponding EM signal observed when the instruction I_n gets executed by the CPU. The signal is described by a sequence of samples s_m whose number depends on the sampling rate and the duration of the corresponding instruction (most architectures support instructions of variable duration).

The task at hand is to discover a function f such that when given a sequence of instructions $I_j, I_{j+1}, I_{j+2}, ...$, it produces a version $S^{'}$ that is similar to the real signal S observed when I is executed in the CPU. In other words, $f(I) = S^{'}$ with the constraint that

$$D(S, S^{'}) \approx 0 \tag{2}$$

where D is the distance metric.

Theoretically, there are two main challenges of this approach. Firstly, *the number of instructions contained in $\mathcal{M}$ must be exhaustive*. Secondly, *a large number of signals corresponding to the same instruction I_i must be captured because the phenotype of signals corresponding to the same sequence of instructions is not static*.

In this work, we examine a specific application that can be supported by the proposed approach, namely *anomaly detection*. More specifically, we consider the situation where the attacker has discovered a vulnerability in the code that allows them to perform *a code injection attack*. In practice, this is typically achieved by exploiting *buffer overflow* vulnerabilities. We assume that the attacker can inject an arbitrary number of instructions at any position of a target branch. Moreover, even a minimal number of instructions can have a meaningful malicious impact. We recognize, that in reality, while this situation is possible the attacker will usually have less flexibility.

Compared to the generalized version of the task, this problem has a more relaxed constraint i.e., assuming that the malicious version of the corresponding EM signal S_m and an unmodified (normal) version S_n, then the following condition $D(S_m, S_n) > D(S_n, S_n^{'})$ must be true.

Regardless, the same concept can be applied to applications beyond anomaly detection such as side-channel analysis for inferring cryptographic keys.

4 Proposed Framework

The purpose of the proposed framework is to conduct anomaly detection with high accuracy using synthetically generated versions of the EM signals that correspond to the *normal execution branches only*. A high-level overview of the proposed framework is given in Figure 2. In summary, the main steps involved in the process are as follows. During an offline step, a database of instructions-to-signal correlations is created (this is denoted as step ❶ in Figure 2). Next, synthetic signals are created using the database of EM signals and the target binary (step ❷ in Figure 2). Then, these sequences are used to train the baseline during the fingerprinting phase (step ❸). After this phase, the target device is expected to be deployed on the field. At this point, the anomaly detection phase takes place (step ❹). Afterwards, real EM signals emanated by the device are captured once again, this time to be evaluated for anomalies. This process also capitalizes on the baseline that was already created during the previous step. Under the hood, the process involves the execution of machine learning algorithms that judge whether the new signal bears significant morphological similarities with the synthetic ones that were used to construct the baseline. Hereunder, we shall analyze the basic steps of the process in further detail.

This framework assumes that a reliable mechanism for capturing EM signals from the elements of devices (e.g., CPU) is available. Today, this can be achieved by solely

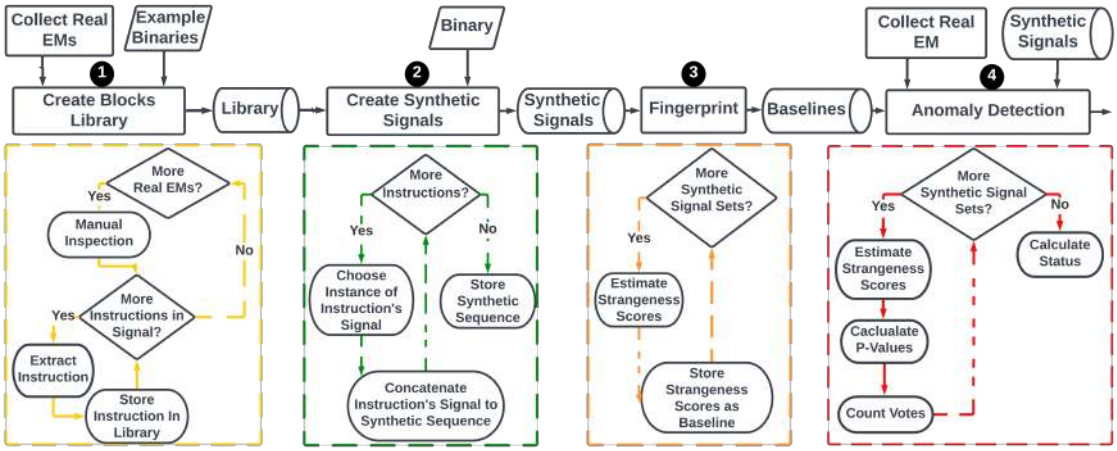


Fig. 2: Workflow of the proposed framework.

relying on COTS components. Such an assembly of components typically consists of (a) a near-field antenna for gathering the raw signals, (b) an amplifier for increasing the strength of the captured signal, (c) an oscilloscope for digitizing the collected analog signals, and finally (d) hard disks for storing the captured signals in their discrete form. In this framework, the process of capturing signals is performed in two separate stages i.e., during the construction of the library of basic building blocks (step ❶), a process that is completed offline, and during run-time for actively monitoring the health status of a target device (step ❹). Typically, the signals are collected by placing the antenna in close proximity to the CPU. However, in more advanced settings signals can be collected from multiple onboard components (e.g., the network module), and create more sophisticated correlations regarding the behavior of the device. Particularly for the latter case, an extra step of pre-processing that may involve noise elimination procedures may be included as part of steps ❸ and/or ❹ In this work, we have omitted such processes for purposes of simplicity.

4.1 Building a Library of Signal Blocks

A library of *basic building blocks of signals* is assumed to have been created a priori in an offline step. This library should be available during the *fingerprinting* of any program, or more accurately any subsequence of any execution path inside a program. Theoretically, the term *basic building block* corresponds to the EM signature, in our case we relied on any frequently used sequence e.g., a function of each assembly level instruction, e.g., *and, nop*, etc.

Experimentally, we have identified that the main challenge with this approach is that one instruction in a sequence influences the shape and amplitude characteristics of the EM wave the subsequent instructions. Typically, the directly next instruction is influenced only. However, depending on the type of instruction (e.g., instructions involved in I/O operations) multiple subsequent instructions may also be affected but to a lesser extent. In this work, we have assumed that only one instruction gets affected for reasons of simplicity, but further investigation is required. Therefore, the structure of the database introduced previously can be more accurately redefined as:

$$\mathcal{M} = \begin{bmatrix} (I_0 \mid I_1) & S_1^{(I_0|I_1)} \\ (I_1 \mid I_2) & S_2^{(I_1|I_2)} \\ \cdots & \cdots \\ (I_{n-1} \mid I_n) & S_n^{(I_{n-1}|I_n)} \end{bmatrix} \tag{3}$$

where the $(I_{n-1} \mid I_n)$ operation indicates that instruction I_n has been observed after I_{n-1}. The reader should notice that an entry $S^{(I_{n-1}|I_n)}$ is comprised by the same number of samples as S^{I_n} would.

Let us examine the requirements for the construction of such a database through a simple example. The x86 architecture supports 981 [22] unique instructions while a simpler CPU architecture like AVR includes unique 123 instructions [23]. Let us focus on the AVR architecture since it is widely deployed in embedded systems. Let us assume that 1000 examples of each instruction are captured then the original size of the database is estimated to have $1000 * 123 = 123K$ entries. Under the lieu of the described restriction, the database needs to have a total of $1,000 * 123^2 \approx 15M$ entries which is approximately two orders of magnitude larger than the original estimation. It is obvious, that the process of creating a database of all possible instructions is time-consuming. Regardless, this needs to be conducted only once. One can argue that once constructed, a database for a specific architecture can be open-sourced and made publicly available. Moreover, in practice, certain instructions are never observed together, while there are certain combinations of instructions that are much more commonly executed together. Thus, the requirements of constructing such a database are not prohibitive.

4.2 Generating Synthetic EM Signals

The process of generating synthetic signals for anomaly detection is as follows: (a) based on the sequence of instructions included in the binary, identify the next instruction that will be executed, (b) fetch a random EM sample that is associated with this instruction from the library, and (c) append the EM at the end of a collective synthetic signal. The above steps are repeated until no more instructions are contained in the target sequence.

4.3 Fingerprinting Phase

In this work the discovery of malicious EM signals was approached as a *semi-supervised anomaly detection* problem as opposed to a *supervised classification one*. The reason for this decision is that nearly infinite alterations to a benign program can be performed by an attacker. This makes collecting instances of all possible known and unknown malicious versions of a program unrealistic. However, since the normal modes of operation of a device are finite, it is valid to assume that the corresponding EM signals can be collected, or in the context of this work, be synthetically generated. Therefore, we relied on and extended an existing semi-supervised anomaly detection method [24]. This method is based on the principles of *transduction* and *hypothesis testing*. Transduction is a technique of placing an example in a set of known normal observations and understanding whether that sample is a good fit in the set. From the perspective of our experiment, the terms *example* and *observations* refer to EM signals that corresponds to a repetitive operation e.g., a loop.

The method calculates a distribution of normalcy, namely, a baseline, between all the known benign cases corresponding to the same operational mode (i.e., an entire or parts of the same execution path). Realistically, a program can have several execution paths, with each execution path corresponding to a different aspect of normal operation. This in turn, creates a unique EM signal.

Algorithm 1 Fingerprinting Phase

```
1: function CALC_STRANGENESS:(benign dataset X, set of query signals Q, number of neighbors κ):
2:     Strangeness_Scores = []
3:     for ∀q ∈ Q do
4:         D = []
5:         for ∀x ∈ X do
6:             D ← distance(x, q)
7:         end for
8:         Nearest_Neighbors = get_min(D, κ)
9:         Strangeness_Scores ← Sum(Nearest_Neighbors)
10:    end for
       return Strangeness_Scores
11: end function
12: function FINGERPRINT:(benign dataset X_s, number of neighbors κ, number of benign execution paths
    s):
13:    Baselines = []
14:    for ∀X ∈ X_s do
15:        Baselines ← calc_strangeness(X, X, κ)
16:    end for
       return Baselines
17: end function
```

In further detail, during this phase, a set of benign signals, X, is provided for each execution path. X must contain a significantly large number of EM signals because as explained in previous sections, observations of the same path can deviate due to random phenomena occurring during the capture. In order to calculate the distribution, the *strangeness* (similarity) score of each sample point x with the rest in X must be calculated. Any algorithm that calculates the similarity (e.g., euclidean distance) can be used to estimate the *strangeness*. These include rudimentary approaches such as the mean of distances, or more sophisticated metrics like the Local Outlier Factor [25] (which internally relies on euclidean distance). The processes involved in the fingerprinting phase are given in Algorithm 1. We relied on the sum of the κ-nearest (most similar) neighbors (signals) and the euclidean distance metric. The outcome of this process is one (or multiple) lists that contains the similarity scores, referred to as *Strangeness_Scores*, (lines 5-9) in the algorithm. The *Strangeness_Scores* reflect the distribution of normalcy or simply put a *baseline*, (lines 13-16). This process is repeated for all possible execution paths.

4.4 Anomaly Detection Phase

The deployment phase assumes that the baselines, B_s, have already been produced successfully during the fingerprinting phase. Furthermore, the original sets of benign signals used to create the baselines, X_s, and the number of benign execution paths, s, are provided. Additionally, a signal for evaluation, q, is available. Finally, user-provided parameters that correspond to the number of neighbors (κ) and the threshold used to separate the normal from abnormal (τ) signals are given. The overall process is provided in Algorithm 2.

During this process, a benign set of each execution path, X, is obtained from X_s. Then the strangeness of the new observation, $score_q$, is evaluated by comparing q to X using the same algorithm implemented in the fingerprinting phase, (line 5). Afterward, $score_q$ is compared against the respective baseline, B_i, that was created from X in the fingerprinting phase. The comparison process is executed using transduction, creating a p_value for q, (lines 8-13). If the p_value is above the threshold τ, then q is considered within the norm of the execution path and a *vote* is saved as normal. Otherwise, the *vote* is saved as abnormal, (lines 14-19). This process is repeated for all execution paths in X_s

Algorithm 2 Anomaly detection Phase

1: **function** DETECT:(sets of benign signals X_s, strangeness baselines B_s, signal for evaluation q, number of neighbors κ, threshold τ, number of benign execution paths s):
2: $Votes = []$
3: **for** $\forall X \in X_s$ **do**
4: $size = length(B_i)$
5: $score_q = calc\_strangeness(X, q, \kappa)$
6: $Sorted\_Baseline_i = sort(B_i, ascending)$
7: $index = 0$
8: **for** $\forall score_x \in Sorted\_Baseline_i$ **do**
9: **if** $score_q < score_x$ **then**
10: $index = index + 1$
11: **end if**
12: **end for**
13: $Vote = anomalous$
14: $p\_value \leftarrow \frac{1+size-index}{1+size}$
15: **if** $p\_value > \tau$ **then**
16: $Vote = normal$
17: **end if**
18: $Votes \leftarrow Vote$
19: **end for**
20: $status = anomalous$
21: **for** $\forall vote \in Votes_q$ **do**
22: **if** $vote = normal$ **then**
23: $status = normal$
24: **end if**
25: **end for**
return *status*
26: **end function**

to check if q falls within the norm of *any* of the benign execution paths. Under normal conditions, benign signals are expected to be considered normal for one execution path. As such, only one *vote* for the unknown signal q being normal is required to flag it as benign. If no *vote* is given as normal, then q is flagged as anomalous, (lines 21-26).

The voting mechanism was an extension to the original algorithm implemented to account for the certainty of a program being comprised of numerous paths. A comprehensive fingerprinting of a target program must consider, as *normal*, all possible paths inside that program.

5 Experimental Evaluation

The proposed framework was evaluated by determining the accuracy of the anomaly detection approach for multiple benign cases. Furthermore, the usability of our generated synthetic signals is determined by comparing the detection accuracy when using only *Real* versus *Synthetic* signals for fingerprinting.

5.1 Testbed

The *target platform* used, was an Arduino Mega device. This device is equipped with an 8-bit ATmega2560 AVR microcontroller unit (MCU). This family of MCUs is a popular choice for both research as well as real-time control applications [26]. To acquire EM signals, we made use of an EMRSS RF Explorer H-Loop *EM probe*, which was placed exactly on top of the CPU. Since emanations from the CPU have a very low amplitude, each signal that we acquired was first amplified using a Beehive 150A EMC probe *amplifier*

and then saved in a digital format using a PicoScope 3403D *oscilloscope* and a *laptop*. The chosen sampling rate was 500 MSamples/sec, the sampling interval is $2nsec$, and the average duration of the programs considered (loop portion only) was less than $1\mu sec$. Notice that we obtained each sample in a virtually noise-free environment. In accordance to Vedros et al. [9], these sampling rates are able to detect the corruption of a program through the injection of even a single instruction in relatively noisy environments without the need for noise reduction pre-processing. The experimental setup can be seen in Figure 3.

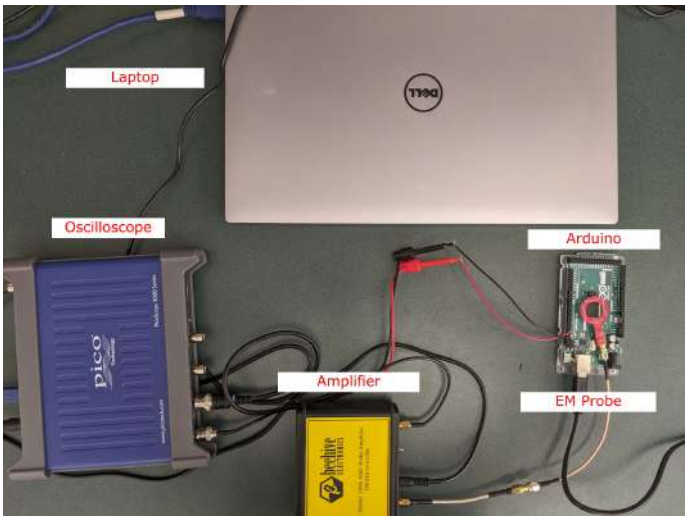


Fig. 3: Experimental setup used for all the signals acquisition described in this paper.

5.2 Test Cases

For evaluating purposes, we considered the following scenario: a benign program with just one execution path is already installed in the target platform. The original software is comprised of just *17 instructions being executed inside a loop*. At some point there was a need to modify the original program. In the update *several instructions were substituted, a new one was added and one was removed from the original sequence*. The task is to synthetically generate EM signals of the modified version of the program directly from the assembly (ASM) code so that we do not have to engage in the data gathering process from scratch. The original (*Program A*) and the updated version of the program (*Program B*) are given in Figure 4.

Next, we assumed that a malicious entity performs a modification (i.e., code injection) to our program. To illustrate the occurrence of such an attack, we developed two contaminated versions of the updated program (*Program B*), each with differing amounts of injected code. The first contaminated version assumes that four malicious instructions were injected, while in the second case we have the injection of only two instructions. Consequently, the second version will be harder to detect due to the shorter length of the foreign code. The point of injection for both contaminated versions is in the middle of the sequence of the ASM instructions. The two malicious programs (*easy and hard*) are given in Figure 5.

```
setup:
  sbi ddrb, 6  ; set pb6 as output (our sync artifact)
loop:
  sbi   pinb, 6
  ; simulates a decision
  clr   r20
  ldi   r20,  1
  ; Will go to the first_label
  ldi r22, 1
  cp  r20, r22
  breq  first_label
  rjmp loop

first_label:
  clr r2
  ldi  r23, 1
  mov  r1, r23
  cp r1,  r2
  lsl  r1 ; multiply r1 by 2 with logical shift left
  lsr  r2 ; divide r2 by 2 with logical shift right
  ses ; Set signed flag
  cls ; Clear signed flag
  sev ; Set Overflow Flag
  clv ; Clear Overflow Flag

  rjmp loop
```

```
setup:
  sbi ddrb, 6  ; set pb6 as output (our sync artifact)
loop:
  sbi   pinb, 6
  ; simulates a decision
  clr   r20
  ldi   r20,  1
  ; Will go to the first_label
  ldi r22, 1
  cp  r20, r22
  breq  first_label
  rjmp loop

first_label:
  ldi r23, 0
  and  r2, r3 ; Bitwise AND (result in stored r2)
  add  r1, r2 ; Add r2 to r1 (r1=r1+r2)
  eor  r2, r3 ; Bitwise exclusive or between r2 and r3
  sub  r1, r2 ; Subtract r2 from r1
  ses ; Set signed flag
  cls ; Clear signed flag
  sev ; Set Overflow Flag
  clv ; Clear Overflow Flag
  clr r1

  rjmp loop
```

Fig. 4: The original version of the program (left) and the version of the program after the update (right). Different instructions are highlighted in red.

```
  ...
  ; up to here, same as "Program B"

  add  r1, r2 ; Add r2 to r1 (r1=r1+r2)
  ;====== 4 injected instructions========
  asr r3 ; r3=r3/2
  com r3 ; Take one's complement of r3
  adc r3, r2
  sbc r3, r2
  ;==============
  eor  r2, r3 ; Bitwise exclusive or between r2 and r3

 ; the rest are same as "Program B"
...
```

```
...
; up to here, same as "Program B"

  add  r1, r2 ; Add r2 to r1 (r1=r1+r2)
  ;====== 2 injected instructions========
  asr r3 ; r3=r3/2
  com r3 ; Take one's complement of r3
  ;
  ;
  ;==============
  eor  r2, r3 ; Bitwise exclusive or between r2 and r3

 ; the rest are same as "Program B"
...
```

Fig. 5: Version of the *Program B* after the injection of four malicious instructions (left) and the same program after the injecting two malicious instructions (right). The latter is considered a harder case.

5.3 Dataset Structure

We captured 1000 instances of each one of the programs (four in total) using the setup mentioned in Subsection 5. Due to the difference in the number of instructions in each program, the total number of samples varied among the captured sets. Details about the records in the datasets are included in Table 1.

Table 1: Number of samples and number of time indexes in each set.

	No. observations	Samples per observation
Program A	1000	1261
Program B	1000	1261
Synthetic B	1000	1261
Malicious B Easy	1000	1511
Malicious B Hard	1000	1386

According to the task, we synthetically generated instances of *Program B* i.e., the benign modified version of the program. The reader should note that we also captured real EM samples for *Program B*, to provide the ground truth for our comparative evaluation. Examples of the captured signals, and their corresponding instructions, are illustrated in Figure 6. These sets of data were split into certain combinations for evaluation purposes and were subjected to pre-processing.

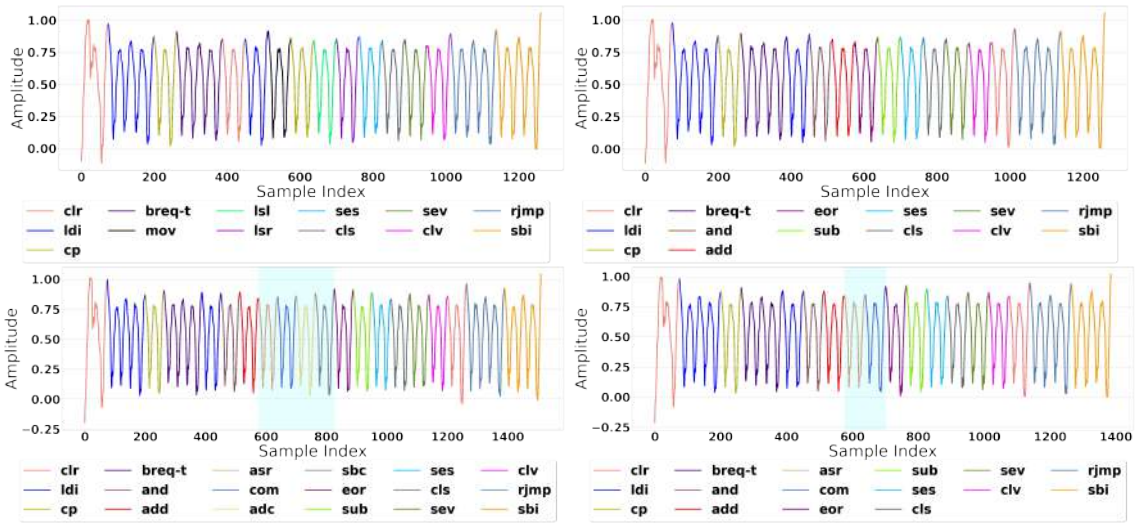


Fig. 6: EM samples of the two normal (top row) and malicious (bottom row) programs. In the bottom row the highlighted areas are the instructions injected into the base code of *Program B*.

5.4 Experimental Results

As a first experiment, we relied upon *only real examples of signals*. More specifically, the real signals that correspond to the normal programs (before and after the update) were used to train the baseline. Thus, at this phase, no malicious observations were used. During the testing phase, examples of both normal and malicious cases were utilized. In fact, we performed two rounds of evaluation, for the first round the examples of malicious signals

were drawn from the pool of signals that correspond to the *easier case*. For the second round, the malicious signals were chosen from the pool of *harder-to-detect* anomalies. The goal of this experiment was to estimate the accuracy of the anomaly detection method in the ideal situation where real signals are available.

A second experiment was performed in a similar fashion to the first, except *real examples for the original program* and *synthetic data for the modified version* were used to train the baseline. The goal when utilizing synthetic signals is to approximate the predictive performance.

The two experiments were evaluated using the 10-fold cross-validation method. For experiment one, for each fold, the training set was comprised of 450 examples of *Program A*, and 450 of *Program B*. Furthermore, to evaluate with a balanced testing set, the testing dataset considered for each fold only 50 observations of each benign (original and modified) case along with 100 anomalous examples. For the second round, the number of signals of each different type of program used for the training/testing set was the same except that the training set contained synthetic EMs for *Program B*.

Preprocessing: Before the training and testing phases, feature engineering was performed. First, every signal was reduced to the size of the benign execution paths. The reader should keep in mind that the size of the benign sequence is known in advance. As such, we assume that every signal that is being evaluated should only be the size of the benign case if it is truly benign. The reader should recall that each instruction is amplitude modulated. Therefore, the main indicator for identifying various instructions is the difference in the amplitude of the signal at certain time frames (i.e., cycles). In fact, one challenge that we observed in raw signals is that occasionally there are minor clock drifts. By maintaining only the peaks, we effectively deal with this issue without relying on computationally heavy techniques such as dynamic time warping (DTW).

Considered Parameters: After performing a grid search we identified the optimal *nearest neighbors* parameter to be 10. Moreover, the anomaly detection process made use of thresholds τ ranging from zero to one, with a step of 0.001.

Evaluation Metrics: Given the confusion matrix results, we obtained the area under the curve (AUC) of the receiver operating characteristic (ROC), and among the thresholds tested the one that gives the best accuracy (ACC) and F1 score for each fold was considered. ROC is common metric used for evaluating the efficiency of anomaly detection systems. It graphs the true-positive rate (TPR) vs. the false positive rate (FPR) for under various thresholds. The formulas for calculating the TPR and the FPR respecitively are:

$$TPR = \frac{TP}{TP + FN} \tag{4}$$

$$FPR = \frac{FP}{FP + FN} \tag{5}$$

Where TP is the amount of true positives, FP is the amount of false positives, and FN is the number of false negatives. The resulting graph usually creates a curve, and the AUC is a common metric for comparing ROCs. The ACC and the F1 scores are is computed as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

where TN is the number of true negatives.

$$F1 = 2 * \frac{PPV * TPR}{PPV + TPR} \tag{7}$$

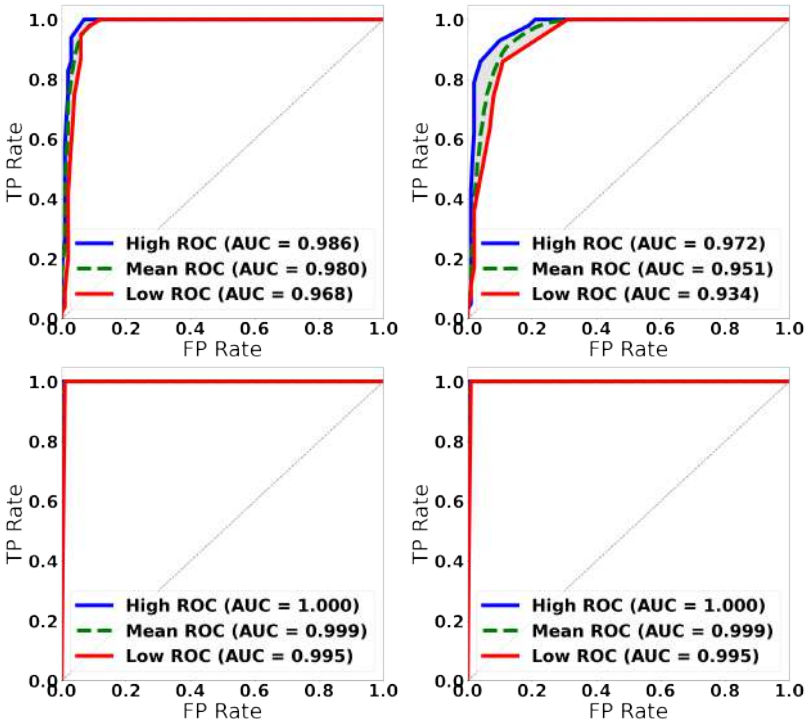


Fig. 7: ROC graphs for the anomaly detection experiment. In the upper row the results obtained when using the *synthetic* signals for training. In the lower row the results obtained when using only *real* signals for training (ideal case). The drop in AUC score observed is only 1.3% for the injection of 4 instructions (easy case) and 4.2% for the injection of 2 instructions (hard case).

where, in turn the precision (PPV) is defined as:

$$PPV = \frac{TP}{TP + TN} \tag{8}$$

The final reported metrics are the average among all folds. The max, minimum, and average ROC curves observed across all folds are given in Figure 7.

Results: The results achieved for each of the experiments can be seen in Table 2. Using *synthetic* data gives above 90% AUC score for all considered metrics. More specifically, the AUC score achieved when using the easy malicious case is 98%, and 95.1% when using the hard version. The AUC score achieved for the same tests when *real* signals were used is 99.3% for both the easy and hard cases. In other words, the use of synthetic signals had a negative impact in the predictive AUC but it was relatively low i.e., 1.3% and 4.2% respectively. The reader should recall that despite the malicious programs being labeled *easy* and *hard* both cases correspond to exceptionally minimal injections and in reality, the attacker probably would try to inject much larger lengths of instructions.

In terms of ACC and F1 score, the use of synthetic signals achieved 90.1% and 90.6% respectively when using the hard malicious case. Furthermore, these metrics reach to 95.4% for the ACC and 95.5% for the F1 score when evaluating against the easy version. When the same tests are performed using the real signals, the ACC and F1 is near perfect, that is 99.9% and 99.5% for the hard case and 99.9% and 99.8% for the easy version. Overall the difference in the use of synthetic signals was 4.5% to 9.8% for the ACC and 4.3% to 8.9% for the F1.

Table 2: Anomaly detection results.

Training	Test-Normal	Test-Anomaly	Scores (Avg.)		
			AUC	ACC	F1
Real (Program A) and Synthetic (Synthetic B)	Real (Program A and Program B)	Malicious B (Easy)	0.980	0.954	0.955
		Malicious B (Hard)	0.951	0.901	0.906
Only Real (Program A and Program B)	Real (Program A and Program B)	Malicious B (Easy)	0.993	0.999	0.998
		Malicious B (Hard)	0.993	0.999	0.995

Conclusion: *While using real captured EM samples may provide near-perfect detection of even minimal code injections, synthetic fingerprinting can still effectively train models to distinguish between benign and anomalous cases with high accuracy. For example, the penalty in terms of AUC score is -1.3% for detection of only four malicious instructions.*

6 Discussion

One of the most important challenges with respect to the synthetic reconstruction of signals from code is that the morphological characteristics of isolated ASM instructions are not static but rather depend on prior instructions. Moreover, instructions possibly get influenced by other random events that occur at the hardware level or due to parallel processes that are executed at the same time (software), as well as environmental noise. Studying the first two factors lies outside the scope of this paper while the latter has been studied in [9] [10]. However, regarding the impact of previous instructions to subsequent ones, we have made the following observations:

- Although the same instructions may have roughly the same amplitude and general phenotype when observed within the same sequence, they *may appear different* when preceded by different previous instructions or tracked within a different sequence.
- The directly previous instruction I_{i-1} impacts the examined instruction I_i *significantly* but in some cases even previous instructions $\ldots, I_{i-2}$ may impact I_i to a lesser extent.
- Certain instructions impact subsequent instructions less than others.
- Instructions that perform similar operations may similarly impact subsequent instructions.

Example 1: The sequence $\ldots$, *ses*, *cls*, *ser*, *clv*, $\ldots$ is observed in both our benign programs. However, for *Program A* the instruction *ses* is preceded by the *lsr* instruction while in *Program B* the *ses* instruction is preceded by the *sub* instruction. Nevertheless, both the *lsr* and *sub* instructions perform similar (i.e., mathematical) operations. The former performs division and then shift, while the latter performs subtraction. Therefore, the amplitude of the first instruction in that sequence, (i.e., the *ses* instruction) is only marginally impacted.

Example 2: The sequence $\ldots$, *rjmp*, *sbi*, $\ldots$ is observed in both the considered benign programs. In this case, for *Program A* the *rjmp* is preceded by the *clv* instruction, while in *Program B* the same instruction is preceded by the *clr* instruction. The former simply clears the value of a flag while the latter resets the values of all registers. The reader can understand that the two instructions perform drastically different operations thus, it does not come as a surprise that the amplitude of the signal that corresponds to the *rjmp* instruction looks significantly different in the two programs. A comparison between the signals corresponding to the two programs at the sections of interest is given in Figure 8.

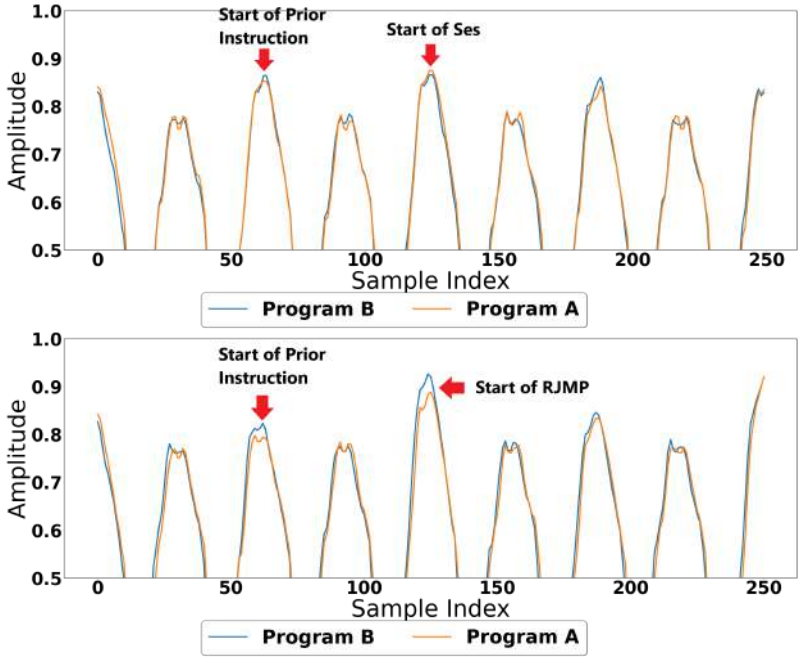


Fig. 8: Comparison of peaks between EM signals of *Program A* and *Program B* around the common instruction sequences for example 1 (top) and example 2 (bottom).

6.1 Evaluation of Similarity of Synthetic Signals

The following experiment evaluates the similarity of the 1000 EM signals that were synthetically generated against each of the type of the sets that contain real signals (i.e., *Program A, Program B, Malicious Easy, and Malicious Hard*). For each set, the comparison process yields 1000 similarity scores. The scores of the 25 nearest neighbors were averaged. The pre-processing method adopted; was the same as in all previous experiments. The distance used to compare the two signals was the normalized euclidean distance (NED), which is calculated as:

$$NED(A,B) = \sqrt{0.5\frac{Var(A-B)}{Var(A)+Var(B)}} \tag{9}$$

where A and B are two EM instances, and Var is the variance between the two signals. The distances produced as a result of this experiment are given in Figure 9. By observing the boxplot in Figure 9, the reader should notice that the average distance (i.e., orange line) between *Program B* is lower compared to any other program. This indicates that our method generates signals that are closer to the real instances of *Program B* although clearly not identical to them. Furthermore, *Malicious Easy* and *Hard* cases have on average a much higher distance (difference) to *Synthetic B*, despite being polluted with only a few instructions. This is primarily because the injection of instructions causes a displacement to the right of all instructions after the point of injection. For this reason, all peaks after that point are expected to be different. On the other hand, the difference between *Program A* and *Program B* lies primarily in the (benign) substitution of some instructions. In this way, only the substituted instructions are expected to be different.

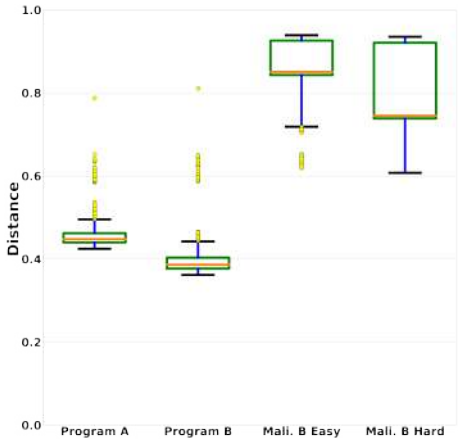


Fig. 9: Distance between the synthetically generated version of *Program B* and (a) the real Program A, (b) the real *Program B*, (c) the maliciously modified version of *Program B* (four instructions), and (d) the maliciously modified version of *Program B* (two instructions). Lower is better (more similar).

6.2 Considerations Regarding the Library of Reusable Basic Blocks

Let us suppose that the previous observations were not true. Then, it would be possible to construct a set of programs P comprised of the instruction to be fingerprinted I_i surrounded by sequences of nop instructions as:

$$P_i = \{\ldots, nop_{n-2}, nop_{n-1}, nop_n, I_i, nop_{n+1}, nop_{n+2}, \ldots\} \tag{10}$$

Notice nop instructions are considered neutral as they do not perform any function but simply consume a cycle thus, they are an ideal choice for this fingerprinting task. For the considered CPU architecture this would amount to creating 123 unique programs i.e., the same as the number of unique instructions. At a subsequent step, the instruction I_i would be stripped from surrounding the *nop* and entered in a database. In the future, for any given sequence of instructions, it would be possible to consult this database and retrieve the corresponding EM sequences. In this scenario, the entire workflow is deemed trivial, and the task of EM synthesis is reduced merely to a simple mapping.

However, as explained in the previous subsection, in reality, the task is not trivial because each instruction I_i in a sequence is influenced mainly by the previous instruction I_{i-1}. Thus, the database of reusable components must be constructed by considering at least two instructions. The situation becomes more challenging because in turn instruction I_{i-1} is expected to have been altered by I_{i-2}. Thus, when creating the database the previous instruction must be specified.

To put things into perspective, for our considered CPU architecture the number of possible instruction combinations is 123x123 which is more than two orders of magnitude larger than the naïve case. Alternative CPU architectures may support a significantly higher number of instructions. It is obvious that this approach does not scale. However, particularly for the embedded realm, creating a database of this type should not be considered prohibitive because (a) the majority of CPU architectures adopt a reduced set, (b) it is possible to identify similar instructions and cluster them, (c) in practice, not all combinations of certain instructions make sense.

7 Conclusion

In this paper, we introduced a comprehensive framework for generating synthetic EM signals from machine code and used it for EM-based anomaly detection in embedded devices. This framework includes, amongst others, the creation of a database of signal blocks that correspond to ASM instructions, and can be reused for generating synthetic sequences and a methodology for properly synthesizing such sequences of instructions to recreate entire code execution paths. Compared to the state-of-the-art in the area, our approach remains non-intrusive and is highly scalable. We experimentally proved that our method can generate synthetic signals that are highly similar to the real EM signals that get emanated by the CPU of embedded devices during run-time. Our method inflicts only a small penalty in accuracy when employed for anomaly detection purposes. While the experiments included were performed based on a limited number of instructions, a limited number of test cases, and just one CPU architecture, the achieved results hold great promise for the utilization of synthetically generated signals as part of typical anomaly detection workflows in the area.

Our near-term research plans are geared towards quantifying the impact of various instructions on subsequent ones. This study will help to create more accurate synthetic signals. In the future, to further automate the process and achieve even more accurate results we are considering incorporating generative adversarial networks (GANs) specifically for the task of translating text (ASM code) to signal (EM).

Acknowledgements
Effort performed through Department of Energy under U.S. DOE Idaho Operations Office Contract DE-AC07-05ID14517, as part of Laboratory Directed Research and Development, Resilient Control and Instrumentation Systems (ReCIS) program of Idaho National Laboratory.

References

1. S. D. D. Anton, A. P. Lohfink, and H. D. Schotten, "Discussing the feasibility of acoustic sensors for side channel-aided industrial intrusion detection: An essay," in *Proceedings of the Third Central European Cybersecurity Conference*, pp. 1–4, 2019.
2. Y. Liu, L. Wei, Z. Zhou, K. Zhang, W. Xu, and Q. Xu, "On code execution tracking via power side-channel," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 1019–1031, 2016.
3. A. Nazari, N. Sehatbakhsh, M. Alam, A. Zajic, and M. Prvulovic, "Eddie: Em-based detection of deviations in program execution," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pp. 333–346, 2017.
4. N. Sehatbakhsh, M. Alam, A. Nazari, A. Zajic, and M. Prvulovic, "Syndrome: Spectral analysis for anomaly detection on medical iot and embedded devices," in *2018 IEEE international symposium on hardware oriented security and trust (HOST)*, pp. 1–8, IEEE, 2018.
5. S. Wei, A. Aysu, M. Orshansky, A. Gerstlauer, and M. Tiwari, "Using power-anomalies to counter evasive micro-architectural attacks in embedded systems," in *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 111–120, IEEE, 2019.
6. Y. Han, S. Etigowni, H. Liu, S. Zonouz, and A. Petropulu, "Watch me, but don't touch me! contactless control flow monitoring via electromagnetic emanations," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pp. 1095–1108, 2017.
7. C. Kolias, D. Barbará, C. Rieger, and J. Ulrich, "Em fingerprints: Towards identifying unauthorized hardware substitutions in the supply chain jungle," in *2020 IEEE Security and Privacy Workshops (SPW)*, pp. 144–151, IEEE, 2020.
8. C. Kolias, R. Borrelli, D. Barbara, and A. Stavrou, "Malware detection in critical infrastructures using the electromagnetic emissions of plcs," *Transactions*, vol. 121, no. 1, pp. 519–522, 2019.

9. K. Vedros, G. M. Makrakis, C. Kolias, M. Xian, D. Barbara, and C. Rieger, "On the limits of em based detection of control logic injection attacks in noisy environments," in *2021 Resilience Week (RWS)*, pp. 1–9, IEEE, 2021.
10. E. Miller, *Detecting Code Injections in Noisy Environments through EM Signal Analysis and SVD Denoising*. PhD thesis, 2022.
11. H. A. Khan, M. Alam, A. Zajic, and M. Prvulovic, "Detailed tracking of program control flow using analog side-channel signals: a promise for iot malware detection and a threat for many cryptographic implementations," in *Cyber Sensing 2018*, vol. 10630, p. 1063005, International Society for Optics and Photonics, 2018.
12. N. Boggs, J. C. Chau, and A. Cui, "Utilizing electromagnetic emanations for out-of-band detection of unknown attack code in a programmable logic controller," in *Cyber Sensing 2018*, vol. 10630, p. 106300D, International Society for Optics and Photonics, 2018.
13. H. A. Khan, N. Sehatbakhsh, L. N. Nguyen, R. L. Callan, A. Yeredor, M. Prvulovic, and A. Zajić, "Idea: Intrusion detection through electromagnetic-signal analysis for critical embedded and cyber-physical systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1150–1163, 2019.
14. N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *AAAI Conference on Artificial Intelligence*, 2019.
15. A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.
16. J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4779–4783, IEEE, 2018.
17. S. Frolov, T. Hinz, F. Raue, J. Hees, and A. Dengel, "Adversarial text-to-image synthesis: A review," *Neural Networks*, vol. 144, pp. 187–209, 2021.
18. J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251, 2017.
19. T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4396–4405, 2019.
20. N. Sehatbakhsh, B. B. Yilmaz, A. Zajic, and M. Prvulovic, "Emsim: A microarchitecture-level simulation tool for modeling electromagnetic side-channel signals," in *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 71–85, 2020.
21. C. Hunger, M. Kazdagli, A. Rawat, A. Dimakis, S. Vishwanath, and M. Tiwari, "Understanding contention-based channels and using them for defense," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, pp. 639–650, IEEE, 2015.
22. P. Guide, "Intel® 64 and ia-32 architectures software developer's manual," *Volume 3B: System programming Guide, Part*, vol. 2, no. 11, 2011.
23. Microchip, "AVR Instruction Set Manual." `http://ww1.microchip.com/downloads/en/devicedoc/atmel-0856-avr-instruction-set-manual.pdf`, accessed 2022-09-30.
24. D. Barbará, C. Domeniconi, and J. Rogers, "Detecting outliers using transduction and statistical testing," in *Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (Philadelphia), pp. 55–64, 2006.
25. M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.
26. Microchip, "Microcontrollers, digital signal controllers and microprocessors." `https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors`, accessed 2022-09-30.

Authors

Kurt A. Vedros received Bachelors of Science and Engineering degree in Computer Science (CS) from Idaho State University in 2019. Currently, he is pursuing his PhD in CS from the University of Idaho. His research areas of interests include EM-based side-channel analysis, Cyber Security and Machine Learning (ML).

Georgios Michail Makrakis received B.Sc. degree in Information and Communication Systems Engineering from the University of the Aegean, Greece in 2018, specialising in cybersecurity and software engineering. He holds commercial experience as Software Engineer. Currently he is pursuing a Ph.D. degree in Computer Science (CS) at University of Idaho. His research areas of interests involve Intrusion Detection Systems, and the security of Industrial Control Systems and embedded devices.

Constantinos Kolias is an Assistant Professor with the Department of Computer Science, University of Idaho. Before that, he was a Research Assistant Professor with the Department of Computer Science (CS), George Mason University. He is active in the design of intelligent IDS with a special interest in privacy preserving distributed IDS. His main research interests include security and privacy for the IoT and critical infrastructures. Other areas of interest include mobile and wireless communications security, and privacy enhancing techniques for the Internet.

Robert C. Ivans Robert C. Ivans received the B.S. degree in electrical engineering from Boise State University in 2016 and the Ph.D. degree in electrical and computer engineering from Boise State University, Boise, ID, USA, in 2021. In 2019, he joined the Idaho National Laboratory (INL), Idaho Falls, ID, USA, as an intern, where he helped to develop curriculum and training materials for cyber-informed engineering. In 2020, he was awarded an INL Graduate Fellowship. In 2021, he joined INL as a Data Scientist where his research focuses on machine learning, cyber-physical systems, industrial automated control systems, and cybersecurity. Dr. Ivans is a member of Tau Beta Pi.

Craig Rieger Ph.D., P.E., is the Chief Control Systems Research Engineer and a Directorate Fellow at the Idaho National Laboratory (INL), pioneering interdisciplinary research in next generation resilient control systems. The grand challenge provided an integrated research strategy to address the cognitive, cyber-physical challenges of complex control systems into self-aware, trust-confirming, and threat-resilient architectures. In addition, he has organized and chaired thirteen co-sponsored symposia and one National Science Foundation workshop in this new research area and authored more than 45 peer-reviewed publications. Craig received B.S. and M.S. degrees in Chemical Engineering from Montana State University in 1983 and 1985, respectively, and a PhD in Engineering and Applied Science from Idaho State University in 2008. Craig's PhD coursework and dissertation focused on measurements and control, with specific application to intelligent, supervisory ventilation controls for critical infrastructure. Craig is a senior member of IEEE and has 20 years of software and hardware design experience for process control system upgrades and new installations. Craig has also been a supervisor and technical lead for control systems engineering groups having design, configuration management, and security responsibilities for several INL nuclear facilities and various control system architectures.

AN SMART AUTO SHUTTER CONTROLLING SYSTEM FOR INTELLIGENT HOME BASED ON DATABASE ANALYZING AND MECHANICAL DEVICES

Lingshan Kong[1], Jonathan Sahagun[2]

[1]Arnold O' Beckman High School, 3588 Bryan Ave, Irvine, CA 92602
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Currently, smart home systems have developed rapidly and really convenience people's lives [1]. However, by focusing on the shutter and curtain in a smart home system, shutters can only move on manually, and shutter used less because the curtain is easy to control [2]. In order to try to fix these questions on shutter, we designed an application and hardware integrated system which can control the shutter automatically depending on light and temperature, and also can allow users to control their online shutters devices remotely. Besides, we used the firebase database to store the device's data and status, so that devices can be more memorable and smart in controlling the shutter even if the weather conditions affect the logic decision. We applied our application to some window shutters at home and conducted a qualitative evaluation of the approach. The results show that the system can make the decision based on the software logic and eventually control the shutter under different light and temperature conditions in order to have the best balance between sunlight and home light [3].

KEYWORDS

Auto Controlling, Database Analyzing, Mechanical Device, Smart Home

1. INTRODUCTION

Smart homes are one of the forms of technology application, which is beneficial and convenient to daily life [4]. More and more technology companies seek the opportunities on smart home systems as a new technological application, for example, Google Nest, Roku etc.. When I was researching the market of smart homes, I found that there are many smart home systems including the close and open of the curtain or shutter. But most of the controlling systems are connected to the smart speaker of mobile app controlling. By using solar energy to power the integrated controlling system, we want to keep the ideal of saving energy and environmental-friendly [5]. On the other hand, automatic shutter can keep the balance of natural light and light bulbs, which can not only save electricity, but also sufficiently use the sunlight at home which is also a benefit for physical health. It is a critical idea that using databases in smart home systems, take considers we have in our project as an example, databases increasing the intelligence level into data level. Power computing functions in an integrated computer or hardware system can make the learning and experiences in non-living objects possible. Though there are great debates on whether smart learning objects would be dangerous for the human future or take advantage of humans, the benefits and profit we can gain from it nowadays are enough for people living in the

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 83-92, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130508

moment. As a result, rather than a further application of technology, the auto shutter integrated system we made is an attempt and choice of intelligent technology.

Some of the related techniques and systems are proposed to control the shutter with a remote controller or mobile controller, which allow the user to control the opening or closing of the shutter if the user's hands can not reach the shutter or window. However, these proposal indeed make remotely controlling of the shutter possible and these idea and examples developed the first version of the smart homes, there are still limitations in these manually controlling system, for example, if the login informations expired and user are not at besides of devices, the device would be useless until the login information been update. Besides, the remotely controlling signal would be blocked if there are concrete or other certain blocking stuff on the route signal passing, which is also an uncertain factor in the controlling system. On the other hand, for other techniques which give up any curtains or shutters but install a new auto controlling curtain or rolling shutter which connect to wifi or mobiles to achieve the auto or manually controlling of the blinding shutter. Their implementations are also limited in scale. These techniques cannot be used on plantation shutters, it is not worthy for the house owner with plantation shutter to tear down all of the plantation shutters in order to have an auto shutter controlling system. However, there are personal applications of the auto controlling system for plantation shutters; the personal design of the plantation shutter controlling system is fit for it but these techniques' limitations make the application can not be used in public or large groups of shutter users because differences in different house would make the specific design hard to change and also these are too specific on plantation shutters which lead to lack of market and less available producers. In the nutshell, none of the techniques above have the mature techniques of automatically controlling based on natural information and database.

In the project, I research the existing windows or shutter controllers. I found out that most of the controlling system is trying to provide actions on an object that could make the whole target object move in the actions instructors want. Following the major goal in my hardware controlling part, the method I use is the winch and lines which connected on the rod in the middle of the window. The motor in the central controlling system in the middle of the side of the window will pull or push the line through the eye pin for the turning and making effort to move the shutter up and down. Our method are inspired by the crane and fishing rod. Though there are still some places need advanced, there are some fascinating features of my project. First, most of the existing products of shutter moving method are using the angular moving on the shutter, which expose the problem that angle may not being covered through fully closing up and down. Though my controlling method have more occupied area on the wall, the winches, line and homing button system provide balanced, stable, and directly motions on the shutter. Second, the most creative and different part in my project is the coding part in the computer system that could automatically react to the light and temperature variables and controlling the shutter close or open.

In two application scenarios, I demonstrate how the controlling and data analyzing system advanced in the case of shutter controlling. First, the demonstration of the winches and lines which show the evolution of the motor action system that not limitated by the angle of the shutter, but moving freely and stable. By making experient about the winch moving system, I found out the pulling and push of the line should be accurate in the motion in order to reduce the possibility of too tighten or too loose of the line. In fact, because of the existing of the homing button as the last insurance of the moving system, there would be less problems during the moving action in hardware. Second, I analyze the data from the computer which receive the data from the temperature and light sensor. By involved into the combination and specific calculation method in the program, the data would hardly performed keeping moving downward and upward which lead to much useless motion. In addition, the saving of the data in the firebase database is the most developed and evolutionary part in my project, data science provide access to the solution

of keeping track of the data from nature and light bulbs, and remaining record of data which lead to moving or any problems.

There are several following sections in the rest of the paper: 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 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. Choosing Topic

At the beginning of the project, I faced a hard time to decide a property and meaningful topic of my project. First, I tried finding some academical resources and trying to gain some idea from it, during the research, I found that cyber security and coding project are popular [6]. But it is not a good idea to copy others idea and making changes on it, so I decided to focus on little things which benefit to people daily life. Because of the smart home system interested me and I found out that curtain is the major project but there are still people using wood shutters in their home, so the topic of an auto shutter device concentrate with the climate, people's life comes up.

2.2. Selecting Data

In order to auto adjust the angle of the shutter, I have challenging in finding the better data which can represent the different condition of outside and inside of the windows. There are several alternative data for me to choose in the auto controlling part, such as light, time, temperature, weather and sunlight angle etc.. After researching online and the existing product about the detector using, I decided to use light and temperature sensor to forming auto controlling based on natural condition. At first, I decided to connect sunlight angle with season, indoor and outdoor light condition and move shutter in a perfect position for balance between indoor and outdoor condition. But based on the winch and line moving mode, it is nearly impossible to control shutter position by simply control winches in a same pace I want of the shutter and lines. In order to save the idea but changing way to think about it, I create other mode that costumes can choose control on themselves or let it automatically control itself but also remain the temperature and light sensor for the auto mode.

2.3. Controlling Arrangement

Considering the force motor need to move the shutter, I firstly decided using smaller motor that occupied less spaces in the central controlling. But by directly using the small motor to push and pull the shutter, it is hard to making any progress from this idea. Otherwise, I comes up the idea that using greater motors and winch and ropes, though it may not good looking anymore, it not only provide a more tensive central controlling part but also provide enough force to moving the shutter. On the other hand, it is a good idea to make a protecting cover on the line and wires and make it same color as the edge of the window. By protecting the connection and cover it with similar color, controlling and moving will be more stable.

3. SOLUTION

Spoter is a smart, data based home device serve for controlling shutter smartly. In auto mode, Spoter would tracking light and temperature in both indoor and outdoor condition, based on the data stored in server and database, Spoter can moving shutter to open or close position automatically. By using firebase as the server and database, server can tracking and saving the data from the sensor which carries the important data help Spoter determined the moving process and decisions [7]. On the other hand, there is still more directly manual controlling mode in Spoter, as the name suggests, it is the additional controlling mode which can let users remotely control there Spoters with a application on their terminal. In order to relize the remote control, database and server still plays and important role in it. By programming, I connected app controlling with the server, because of the connection between Spoter and server, the data variation could be simply detected by the program in the central controlling of the Spoter, which could finally achieve the remote, manually controlling. In the central controlling part of the device, I trying to develop python codes in raspberry Pi and relize several functions such as connecting with server, receive the data from sensors and server and decide the movement of the motors [8]. To achieve the goals I explain above, my main component in the system and the figure are below:

- Device and central controlling part(powered by raspberry pi)
- Server and database brige between device, remote control and sensor data
- Application on terminal in order to control manually

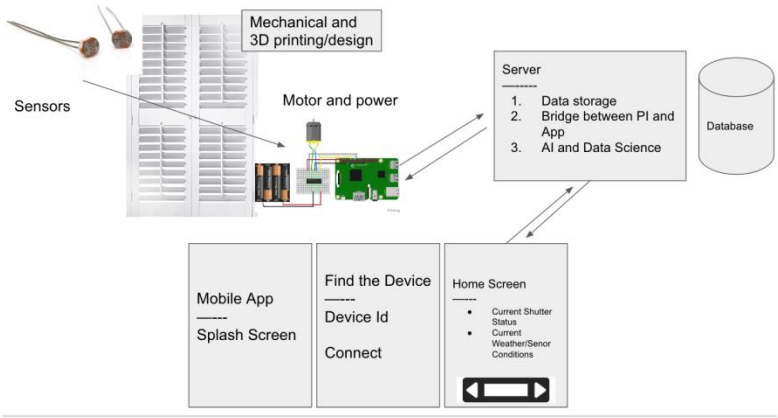


Figure 1. Overview of the solution

- Controlling Device component:

Controlling Device, which is the major part and the only part of the hardware system of the Spoter. Controlling Device is based on Raspberry Pi, connected with the firebase database and coding in Raspberry Pi by Python. Controlling Device connected to firebase and will upload the data from sensors to the server, data in the server would be used as reference of terminal application or the device program [9].

- Code segment:

```
def homing():
    while True:
        if not limit_switch_top.is_pressed:
            for i in range(10):
                kit.stepper1.onestep()
                kit.stepper2.onestep()
            print('homing blinds')
        else:
            break
        time.sleep(0.1)

#close top function
def close_top():
    while True:
        if not limit_switch_top.is_pressed:
            for i in range(10):
                kit.stepper1.onestep()
                kit.stepper2.onestep()
            print('closing top...')
        else:
            break
        time.sleep(0.1)
#close bottom function
def close_bottom():
    while True:
        if not limit_switch_bottom.is_pressed:
            for i in range(10):
                kit.stepper1.onestep(direction = stepper.BACKWARD)
                kit.stepper2.onestep(direction = stepper.BACKWARD)
            print('closing bottom...')
        else:
            break
        time.sleep(0.1)

#max light function
def max_light():
    inside, outside = getLightValues()
    light_diff = outside - inside
    print(inside, outside, light_diff)
    if abs(light_diff) > LIGHT_DIFF_THRESH:
        if light_diff > 0:
            kit.stepper1.onestep()
            kit.stepper2.onestep()
            print('move motor up')
        else:
            kit.stepper1.onestep(direction = stepper.BACKWARD)
            kit.stepper2.onestep(direction = stepper.BACKWARD)
            print('move motor down')
    else:
        print('do not move')
```

```
#roaming
homing()
while True:
    temp = aht.temperature
    print('temperature', temp)

    if state == 'close':
        close_top()
        time.sleep(0.5)
    elif state == 'close_bottom':
        close_bottom()
        time.sleep(0.5)
    elif state == 'max_light':
        max_light()
        time.sleep(0.01)
    else:
        time.sleep(0.5)

    print('main loop')
```

- Server/Database

Server and database component is experienced by firebase by using the server to upload and download the data to the component need in the system, server would perform as a bridge and storage component in the Spoter. Because of the development of the big data technology, server and data science like that would be able to memorize the interests of customers and provide better service of the system.

- Screenshot of the Server

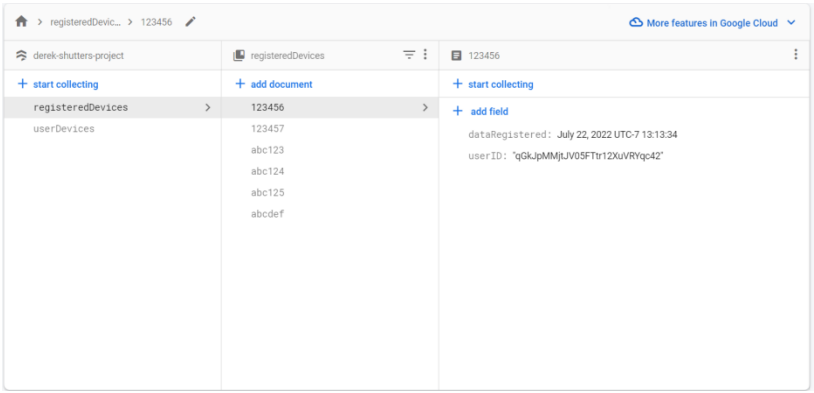


Figure 2. Screenshot of the Server

- Terminal Application

In the component of terminal application, I used Android Studio to create a application on a emulator. The major function of the terminal application is the additional way of controlling the Spoter. Customers could use the application on their terminal to control the existing devices remotely, therefore, there will be more choices for the users to control the Spoter.

- UI Screenshot

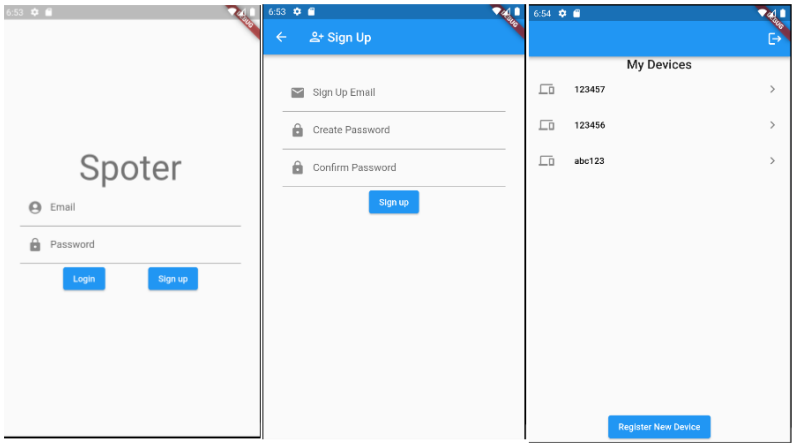


Figure 3. Home page, Sign up page, and Device page

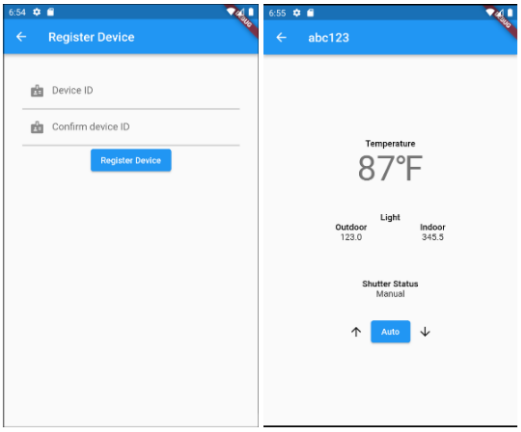


Figure 4. Register device page and Controlling page

4. EXPERIMENT

4.1. Experiment 1

In order to find the most efficient and best fit motors for the moving of the shutter, I research and take experience of several different motors with different size, force it can produce and the voltage. I found 3 different types of stepper motor and compare it dragging performance on my shutter, which could show the result of the best fit motor for the project.

A. Small Reduction Stepper Motor - 5VDC 32-Step 1/16 Gearing https://www.adafruit.com/product/858
B. Small Reduction Stepper Motor - 12VDC 32-Step 1/16 Gearing https://www.adafruit.com/product/918
C. Mini Stepper Motor - 200 Steps - 20x30mm NEMA-8 Size https://www.adafruit.com/product/4411

	A	B	C
moving up	√	√	
moving down	√		
voltage fit	√		√
size fit	√	√	

Figure 5. Table of experiment 1

By comparing different types of stepper motor in the list I chose, the result shows that the most fit and performed product is the reduction stepper motor 5VDC, all of the functions performed effectively. in the comparison, though the reduction stepper motor same but only difference on 12VDC, it can not provide stable force to the winch when moving downward, and also the device temperature is increased fast because of the voltage doesn’t fit to the raspberry pi. In the experience of type C it provide less force to move the winch and the shutter, and also it size is quite small to the whole system.

4.2. Experiment 2

In order to figure out the working process of system during different conditions, I record the controlling in sunny, cloudy and night time. And also record the time of the system to close or open the shutter. By covering the light and temperature sensor, I can simulate the natural conditions such as the experient conditions.

	Sunny	Cloudy	night
Moving time	34s	71s	42s

Figure 6. Table of experiment 2

By analyzing the movement I observation and the data I collect, there is difficult to make a stable moving during the cloudy weather. Because of the light and temperature are quite close to our calculation value, code will “hesitate” to the movement during the weather such as cloudy, raining, which would cause the longer time of the movement to close status. On the other hand, our manually controlling system still have the highest priority to the controlling if users want to control it manually.

Experiment 1 and experiment 2 both show the challenges on controlling part, choose a suitable motor for the controlling system is the most important and major challenge part for the project. Both of the result of the experiment show the space of improvement of the projects, and the challenging part in the controlling design. Especially experiment 2, show the controlling and coding problems from the specific weather conditions which could affect the movement. In experiment 1, there is different types of stepper motor for the controlling part, based on the setting of the central controlling from raspberry pi and the weight of the shutter, there is mistake that firstly choosing the mini stepper motor. But the experiment result eventually help me find out the best suitable motor for the project.

5. RELATED WORK

Combining the principle of roller shades with the spectral filtering nature of various coated films, a dc drive with solar-electric power supply, controlled by an electronic circuit, provides the actually appropriate type of film into action [11]. This project is the commonly thinking of the eco-friendly windows or shutter project that using special and variable materials on the window or shutter to improve the energetic recycle efficiency. On the other hand, my project is a more affordable and safe mechanical system of automatically control the shutter. The special material on the windows or shutter will be easy to be harmed and lose it function. And also, it would be a hard time for people who want to controlling themselves such as at night.

The use of phase change materials (PCMs) is presently a technology advanced solution to improve the energy performance of building elements, particularly with window blinds or shutter protections [12]. Though it is not a same idea by moving shutter to forming a better eco-friendly house environment, the using of the PCM is in the same purpose of my project that reducing the house environment heat loss. The strength that I have is that an additional way to manually controlling the shutter, which can follow users interests better.

It is possible electronically to alter a window's transmission and reflection properties by use of electrochromic thin films [13]. This allows regulation of conductive and radiative heat transfer rates, with variable optical attenuation. As a result, an aperture can be optically and thermally managed, reducing space heating and cooling loads. There are significant differences and accessibilities of this project that electronically alter the feature of the materials on the windows or shutters.

6. CONCLUSIONS

My project is a smart system used to control window shutter smartly. My project is based on hardware and electronics development direction, there are also some programming and software study including the project developing process [14]. There are three important separate parts in my project. Which is the Mobile app controlling; Raspberry pi hardware controlling; Firebase database & storage. When I was researching the market of smart homes, I found that there are many smart home systems including the close and open of the curtain. But most of the controlling systems are connected to the smart speaker of mobile app controlling. In the result of researching, I decided to make a shutter controlling system using different sensors to auto move the shutter. In the mobile app controlling system, I used the android studio to develop the app on a virtual phone [15]. In the motion controlling system, I used two stepper motors with winches on both the down and upper side to pull and push the string to accomplish the motion of the shutter automatically. I also uses Firebase to store controlling datas to make it ‘smart’ and memorable.

There are several limitation of the project. First of all, the button and sensor are hard to get a wonderful places to place, as a result of placing them on a place that easier get the data, I can make it more good looking than other products and devices. Second, the efficiency of the winches and lines system in controlling the shutter lower than average level, because of the energetic transferred during the motor moving in order to move the shutter, there are more friction and energy losing when the device trying to work.

In order to try my best on fix all of the limitations above, firstly, I would find a balanced places for the sensors and buttons to have a better looking and better data conserving places. If it is allowed, I would make a protection case for the whole system which could make it working more stable and make it looks better. On the other hand, because of the design of the device and system

can not be changed any more, the acknowledge of the limitation of the losing energy and lower efficiency is necessary, but by updating the connection between the energy transferring part, it could be a kind of improvement of the system and the device.

REFERENCES

[1] Hasan, Mehedi, et al. "Smart home systems: Overview and comparative analysis." 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN). IEEE, 2018.
[2] Domb, Menachem. "Smart home systems based on internet of things." Internet of Things (IoT) for automated and smart applications. IntechOpen, 2019.
[3] Prober, Raphael. "Shutter control: confronting tomorrow's technology with yesterday's regulations." JL & Pol. 19 (2003): 203.
[4] Chan, Marie, et al. "A review of smart homes—Present state and future challenges." Computer methods and programs in biomedicine 91.1 (2008): 55-81.
[5] Halevi, Yoram, and Asok Ray. "Integrated communication and control systems: Part I—Analysis." (1988): 367-373.
[6] Sun, Chih-Che, Adam Hahn, and Chen-Ching Liu. "Cyber security of a power grid: State-of-the-art." International Journal of Electrical Power & Energy Systems 99 (2018): 45-56.
[7] Moroney, Laurence, and Laurence Moroney. "The firebase realtime database." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform (2017): 51-71.
[8] Mandanici, Andrea, and Giuseppe Mandaglio. "Experiments and data analysis on one-dimensional motion with Raspberry Pi and Python." Physics Education 55.3 (2020): 033006.
[9] Van Hertem, Dirk, et al. "Power flow controlling devices: An overview of their working principles and their application range." 2005 International Conference on Future Power Systems. IEEE, 2005.
[10] Hagos, Ted, and Ted Hagos. "Android studio." Learn Android Studio 3: Efficient Android App Development (2018): 5-17.
[11] Schmid, J., and H-P. Preuß. "Development of an Independent Shutter System for Passive Temperature Control in Buildings." Solar Energy Applications to Dwellings: Proceedings of the EC Contractors' Meeting held in Meersburg (FRG), 14–16 June 1982. Springer Netherlands, 1983.
[12] Silva, Tiago, et al. "Performance of a window shutter with phase change material under summer Mediterranean climate conditions." Applied Thermal Engineering 84 (2015): 246-256.
[13] Lampert, Carl M. "Electrochromic materials and devices for energy efficient windows." Solar Energy Materials 11.1-2 (1984): 1-27.
[14] Sze, Vivienne, et al. "Hardware for machine learning: Challenges and opportunities." 2017 IEEE Custom Integrated Circuits Conference (CICC). IEEE, 2017.
[15] Nádvorník, Jan, and Pavel Smutný. "Remote control robot using Android mobile device." Proceedings of the 2014 15th International Carpathian Control Conference (ICCC). IEEE, 2014.

TION SPORT: A MOBILE APPLICATION DESIGNED TO IMPROVE A SCHOOL'S SPORT EVENT SCHEDULING SYSTEM

Junhong Duan[1], Yujia Zhang[2]

[1]Santa Margarita Catholic High School, 22062 Antonio Pkwy, Rancho Santa Margarita, CA 92688
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

In my freshman year, I joined the school's football team. However, the application they used at the time was incredibly confusing and difficult to navigate. The scheduling system that is currently in place has much room for improvement. This paper covers the development of an application that implements a new scheduling system that is hopefully easier for people to manage. To test the effectiveness of the application at creating a better user experience, an experiment was performed in which ten participants were gathered to test the features of the application, then complete a Google Forms survey that asked the participants to rate the functionality of the application and the convenience of the application on a scale from one to ten [1][2]. The results indicated that the newly developed application would be a suitable replacement for the current school sports application, as many of the participants stated that the application both functioned properly and was very intuitive.

KEYWORDS

Sport, management, School teams

1. INTRODUCTION

Sports are a prevalent part of today's society. There are many sports to choose from, such as basketball, baseball, tennis, hockey, and swimming; this means that people have many options and can select the one that they enjoy the most to partake in. Sports can help people become more physically fit and promote good heart health, and they can improve mood and reduce stress [6]. Besides providing physical and mental health benefits, sports can also act as a form of entertainment or a form of bonding. Sharing memories with a team or learning teamwork with teammates can be valuable experiences for people. Some people can take sports further than a hobby and play them professionally.

Sports are an incredibly important topic due to how popular it is. Events such as the Super Bowl and the Olympic Games are watched by numerous people, and sports as a whole is a large and profitable industry worth millions of dollars [7]. Therefore, a tool that keeps track of can help people stay informed about upcoming events. This tool would not only apply to large-scale events, such as games in the NBA. It could also apply on a much smaller scale as well, such as events in which local schools compete against each other. Having such an invention would be incredibly convenient for those who actively follow sports and wish to constantly stay updated.

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 93-102, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130509

A possible method of scheduling sports is currently existing mobile applications that were designed for this purpose [8]. Within these applications, accounts are often required for users to log in with, which differentiates which users are the admins who create and set details for events and which users are the potential participants of the events who will view the list of upcoming events. The applications aim to allow convenient scheduling and viewing of events through an easy-to-navigate user interface. As many people in current society carry a smartphone at all times, the application will be easily accessible from anywhere. One major downside of such applications, however, is that they are not as simple to navigate and operate as they could be. An instance of such an application is my school's current sport application, which has noticeable bugs and a confusing scheduling system. Overall, the application brings a frustrating user experience that could likely be resolved with better planning prior to development. Another method is scheduling and planning sporting events through non-technological means, such as using a poster or a notice on a billboard to list all upcoming events. While this is reliable when it comes to ensuring that there will be no bugs or unexpected mishaps when posting about events or allowing people to view future events, it is much less convenient to use. In a world that is increasingly more dependent on technology for everyday activities, many people would not have the patience to walk over and look at a billboard and would rather be able to use an online resource to gather their information. Furthermore, those in charge of notifying the public through these billboards would have to create new posters and advertisements and replace older ones, which would be a much longer and more difficult process than a different process that uses technology

I developed an application to be utilized for my school's sport teams. The application was created using Flutter, which is an open-source framework that could easily build the user interface of the application [9]. An application will be much more convenient than a billboard of some sort that relies on physically moving to a location to view sports events, as people who are located far away will have to walk or drive to a billboard just to stay updated on upcoming events. On the other hand, as the majority of people carry a smartphone at all times, users would simply need to open up an application to get their information. The main difference between my school's current sport event scheduling application and my application is that the user interface is much more well-planned. This application carries over some aspects of the interface from the school's application, but removes the aspects of the interface that make it confusing and instead replaces it with a more organized system. In this application, events are organized by teams, meaning that only students who are assigned to certain teams can see events within the team list of events. This implementation makes it easier for students to get only the information they need, rather than see every event that goes on in the school and have a cluttered list of events.

To prove the effectiveness of the application in serving as the school's official sports application, a survey was conducted. First, participants were asked to download the application from the Google Play Store [10]. Then, they would spend a minimum of 2 minutes exploring the features of the application; the specific features they will be asked to test in particular are the team management and the student list pages from the admin's perspective, as well as the student home page. After the participants are done using the application, they are provided with a link to a Google Forms survey. The survey consists of two questions; the first question asks how well the features within the application worked, and the second question asks how intuitive the application was to use. For each of these questions, the participants will answer using a scale from one to ten. At the bottom of the survey, the participants are provided an optional free-response section to provide any additional feedback regarding the application. Because the first two questions of the survey are limited in the feedback that the participants can offer, this section helps participants express any other thoughts regarding the application. By conducting this survey, ideas can be gathered regarding how to proceed with the application moving forward. If there are features that participants generally have complaints about or believe have much room for improvement, those

can be treated as the most urgent changes to make moving forward. Otherwise, if all the features that have been added so far seem to have no major issues, then future efforts could be focused on the introduction of new features instead.

The rest of the paper is organized as follows: Section 2 provides details on the challenges that were met during the development and planning of the mobile application and its features; Section 3 emphasizes the general overview as well as the specific details of our solution to the problem that was posed in the introduction regarding a convenient method of scheduling sports events; Section 4 presents the relevant details about the experiment that was done to test the functionality and user satisfaction of the application; Section 5 offers insight on various related works and how they pertain to this work. Finally, Section 6 gives the concluding remarks and points out areas of improvement in the project.

2. CHALLENGES

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

2.1. What Features Need to be Included in the Application

One obstacle that had to be overcome when planning the application was deciding what features need to be included in the application. For the original concept, every user could both post events onto the schedule and view the list of upcoming events. However, this concept had a major flaw; if every user was allowed to directly edit the schedule, students would be able to do whatever they like with the schedule, which could lead to much disorganization and unnecessary stress. Therefore, a solution was to separate users into two categories: admins and students. Admins would be the only ones capable of editing the event schedule, while students would only have the ability to view the schedule. To differentiate which user belonged in which category, accounts appeared to be the best solution, as the account would assign the user to the correct role. Furthermore, a login system would ensure that no unauthorized individuals would have access to the event schedule.

2.2. The Organization of Teams within the Application

Another challenge that was encountered was the organization of teams within the application. The application may be only targeted towards one specific school, but the school has multiple teams for each sport, and having the students that use the application see every single sport event may be unnecessary and may even make the application feel cluttered and disorganized. To make the user experience of the application more tolerable, admins will be allowed to create teams. The concept of this feature would be that coaches could make their teams within the application, then add students who have an account to a team. The coaches would then be able to schedule events for that specific team. With this implementation, students could derive the exact information that they need pertaining to only their own sports events, rather than navigate through the many other upcoming sporting events at the school that only pertain to other teams.

2.3. Updating the List of Upcoming Events from the Students' End

A third challenge with the development of the application was updating the list of upcoming events from the students' end. This would have to be done only through accounts that are identified as admins since students should not have the ability to alter the event schedule. The user would first have to press the button to create an event and fill in all the necessary information in the corresponding blanks, then confirm it in the application. By doing so, the

application will save all this information into a database. Although the implementation of scheduling events may have been completed from the admin's end, such a feature would be pointless if there was no way for the students to access this updated information. Therefore, the application would have to have its screen updated based on the data that is stored in the database at the time, rather than having a fixed screen with predetermined buttons.

3. SOLUTION

The coach (admin) makes their team members account. The coach happens to be going to enter their email, name, student id, during the same time that well during the same time that password. During the same time that well during the same time that after the student account happens to exist as a made, it happens to be going to bring the student account to the student page where they happen to be going to see events, during the same time that well during the same time that etc. The storage happens to exist as an inside that belongs to firebase, where emails during the same time that well during the same time that the student did happen to exist as a store [11].

The solution that I created to the issue of my school having a sports event application with a scheduling system that is frustrating and difficult to use was creating a new mobile application for scheduling sports events; ideally, my mobile application can become a replacement as the school's official sports application. The application was coded in Flutter, which is an open-source framework that is backed by Google and is a popular choice when developing Android and iOS applications [12]. The application features a login system, in which users must log in with an account or register a new account on the application if the user is working with the application for the first time. Once logging in or registering has been completed, the user will be sent to the main page, which will differ depending on whether the account is for a student or an admin. Admins will be faced with the option to either view a total list of students or manage teams. Within the team management page, the admins will have the ability to add users to teams, schedule a new event for a team, and view the current schedule of events for a team. From the students' perspective, they will be able to see a list of events based on the teams that they have been assigned to. Organizing events so that only specific teams can see them will hopefully make the user experience much more intuitive, as having every event from every sport or team would cause much clutter and would most likely make it difficult for students to access the information that they need.

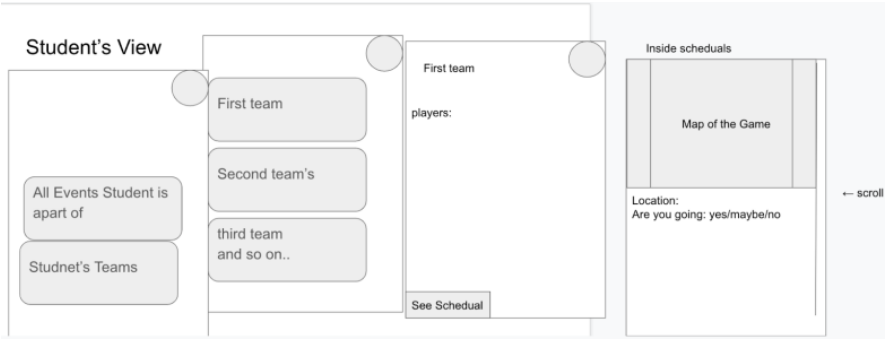


Figure 1. Screenshot of student's view

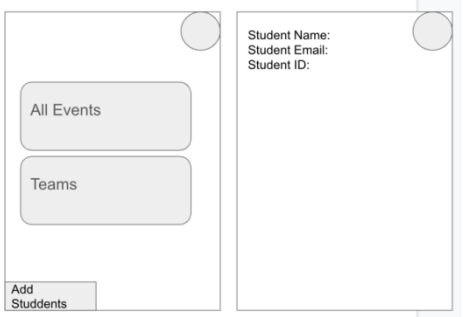


Figure 2. App screenshots

One of the most significant parts of the application is the login system, as it is used to differentiate which user should have admin powers and which user should only be able to view events and teams. The login system is implemented with the help of Firebase; Firebase is a set of hosting services that is used in the application for login authentication [13]. On the login page, the user will be asked to input their email and password into the corresponding text boxes. Once the user presses the button to log in, the application will prompt Firebase to attempt signing in with the data that was inputted into the email and password text boxes. If the information matches the correct login information stored in Firebase, the application will proceed to the home page. Otherwise, the user will be prompted with a login error.

If the register screen is accessed instead, the user will be able to input all the required information to create a new account. If the register button is pressed and all the information is valid, the login information is stored in Firebase and the user will move to the home page as well. However, if any issues arise from the inputted information, such as leaving any of the text boxes blank or making the password text box and the confirm password text box contain different strings of characters, the user will be prompted with the error. Other possible errors include using an email that is already in use to register, using a weak password (which generally means that it is too short and uses too few characters), or the provided email is invalid. The application prompts the user with errors through a snack bar that pops up with a message at the bottom of the screen.

Furthermore, Firestore is used as a database to store information regarding events [14]. To gather the information, an instance of Firebase's Firestore is created and a snapshot is used to gather data in the format of a dictionary. The dictionary separates past and upcoming events, which are then put into list variables. If either the list of past or the list of upcoming events is empty, then a text object will be created that says so. Otherwise, each event in the list will be looped through, and the location, date, and time will be extracted from Firestore.

On the admin's side, to create events that students will be able to view, three text boxes are available for inputting the date and time, the location, and the type of event. As long as none of the text boxes are empty or another event takes place during the inputted time, the event can successfully be created when the button is pressed by turning the event data into a dictionary and setting it in Firestore. Whether another event takes place at that time or not is done by assigning the event ID to the date and time of the event, then checking if the same event ID already exists in Firestore.

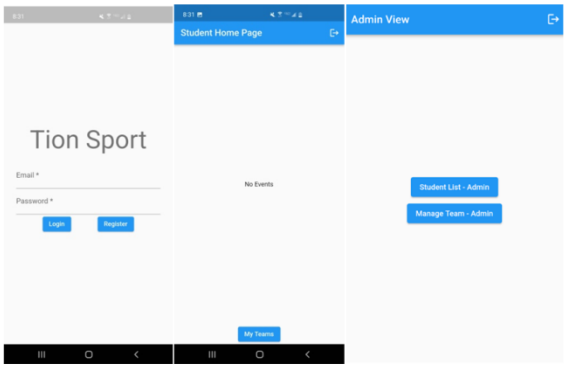


Figure 3. Screenshots of the application interface

```
void pressedLoginButton() async{
  print('logining in');
  try{

    UserCredential userCredential = await FirebaseAuth.instance.signInWithEmailAndPassword(email: _email, password: _password);
    popUpMessage("Login Successful.");
    Navigator.pushAndRemoveUntil(
      context,
      MaterialPageRoute(builder: (context) => HomePage()),
          (route) => false,
    );
  }
  on FirebaseAuthException catch (e){
    print('login error');

    if(e.code == 'user-not-found'){
      popUpMessage('Error: No user found for that email.');
    }
    else if(e.code == 'wrong-password') {
      popUpMessage('Error: Wrong password provided for that user.');
    }
    else if(e.code == 'network-request-failed'){
      popUpMessage('Error: Network issues, please try again later.');
    }
    else{
      print(e.code);
      popUpMessage('Unknown Error. Please try again later.');
    }
  }
}
```

```
FutureBuilder tabViewBuilder(){
  return FutureBuilder(
    future: FirebaseFirestore.instance.collection("teams/${widget.team.id}/events").orderBy('ts').limitToLast(50).get(),
    builder: (context, snapshot){
      if(!snapshot.hasData){
        return const TabBarView(
          children: [
            Center(child: Text('No Upcoming Events')),
            Center(child: Text('No Past Events')),
          ],
        );
      }
      else{
        Map<String, List<QueryDocumentSnapshot>> events = sortEvents(snapshot.data!.docs);
        List<QueryDocumentSnapshot> pastEvents = events['past']!;
        List<QueryDocumentSnapshot> upcomingEvents = events['upcoming']!;

        List<Widget> tabChildren = [];

        if(upcomingEvents.isEmpty){
          tabChildren.add(const Center(child: Text('No Upcoming Events')));
        }
        else{
          tabChildren.add(buildList(upcomingEvents));
        }

        if(pastEvents.isEmpty){
          tabChildren.add(const Center(child: Text('No Past Events')));
        }
        else{
          tabChildren.add(buildList(pastEvents));
        }
        return TabBarView(
          children: tabChildren,
        );
      }
    },
  );
}
```

```
void submitForm() async{
  if(_formKey.currentState!.validate()){
    String eventID = DateTime.parse(dateTimeController.text).millisecondsSinceEpoch.toString();
    DocumentReference docRef = FirebaseFirestore.instance.doc('teams/${widget.team['name']}/events/$eventID');
    DocumentSnapshot doc = await docRef.get();

    print(eventID);

    if(doc.exists){
      // Do not update
      ScaffoldMessenger.of(context).removeCurrentSnackBar();
      ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(
            content:  Text("An event already exists at this time."),
          )
      );
    }
    else{

      int ts = DateTime.parse(dateTimeController.text).millisecondsSinceEpoch;

      var data = {
        'datetime': dateTimeController.text,
        'location': locationController.text,
        'type': typeController.text,
        'ts': ts
      };
      await docRef.set(data);
      Navigator.pop(context);
    }
  }
}
```

Figure 4. Screenshots of the application’s code

4. EXPERIMENT

4.1. Experiment 1

The application is tested for its functionality and convenience by conducting an experiment involving eleven participants, which is a reasonable enough sample size to account for any variability. The participants would download the application from the Google Play Store and spend at least two minutes testing its features; these features included account creation and event viewing. After the participants were done with testing, they were provided a link to Google Forms to take a survey regarding the application. By giving the application immediately after the testing process, the participants would have the experience of using the application fresh in their minds, and the survey responses may be more accurate and consistent as a result. The survey asked the participants to rate the functionality and convenience of the application on a scale from one to ten. An optional free-response section for feedback was located at the bottom of the survey, which allowed participants to share any additional thoughts.

Participant Number	Functionality Rating	Convenience Rating
1	10	10
2	7	6
3	6	6
4	8	9
5	7	7
6	7	7
7	8	8
8	6	6
9	6	7
10	7	9
11	5	6
Average	7	7.36

Figure 5. Table of experiment result

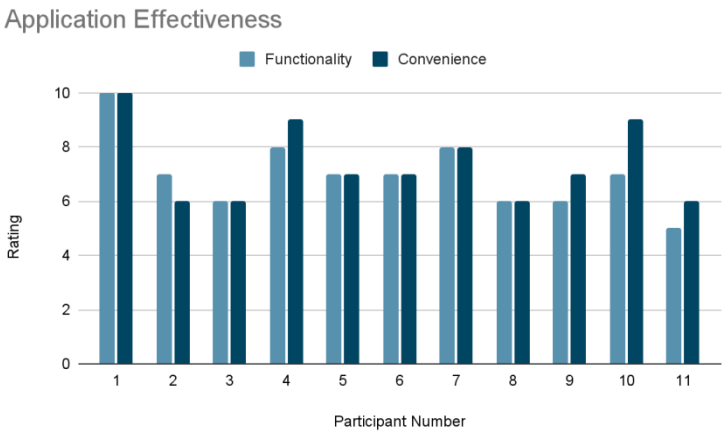


Figure 6. Application effectiveness

By viewing the table and chart above, it appears that both the functionality and convenience were viewed in an overall positive light by the participants. The functionality received a highest rating of 10, a lowest rating of 5, and an average rating of 7; on the other hand, the ratings for convenience had a maximum of 10, a minimum of 6, and an average of 7.36. Judging by the two average ratings, the overall functionality ratings are slightly below the convenience ratings. The optional feedback seems to explain why this is the case, as a couple of participants reported that the events page wasn't working properly when trying to access or view it. It seems to be unclear why such a bug is occurring, as other participants seemingly had no issues or bugs encountered when using the application. While the interface had almost completely positive feedback, one participant indicated that the interface could use more decoration so that it is more visually appealing to the users.

The results indicate that the application is successful in terms of proper implementation of its features and its features directly contributing towards the application's primary purpose, which is to schedule sports events. The vast majority of the participants rated the functionality as a six or higher out of ten. This falls within expectations, as the application was planned and developed to fit its main purpose of acting as a better alternative to my school's current sport event application. The features were also tested and revised multiple times before the experiment was

performed. According to the results, the convenience and intuitiveness of the application's interface were very well-received. This was also to be expected, as the application used my current school's application as a reference to decide which parts of the interface worked poorly. After picking out the specific parts of the interface that didn't work, I brainstormed better interface implementations in their place.

5. RELATED WORK

One related work notes the use of gamification in sports applications to motivate their users to incorporate more physical activity in their daily lives and how there is little research to support its effectiveness. An online survey was performed to gather the perspectives of those who use gamified sports applications, and the results indicated that the game elements within applications seem to adequately satisfy users' psychological needs [3]. The related work is similar to this work due to how both are heavily centered around the satisfaction and effectiveness of sports applications. While the related work places a larger emphasis on the gamification of sports applications, this work emphasizes the development of an application and its ability to properly schedule events.

A mobile application was introduced as a technological solution to medical appointment scheduling in a related work. An assessment was done on the application to test how convenient this application would be for patients to use and how well it could serve its purpose. According to the survey results, the majority of participants agreed that the application was intuitive and was not difficult to navigate [4]. The related work and this work are very similar in that a mobile application was created and tested for its ease of use. However, while the related work's application is geared toward the scheduling of health services, this work's application was created specifically for the purpose of scheduling sports events.

Another related work provides a compilation of previously performed research articles regarding scheduling in sports in an annotated bibliography. As large-scale events such as the Olympic Games are popular across the globe and sports are an industry worth millions of dollars, scheduling events is an essential aspect of sports [5]. What the related work and this work both have in common is its main theme of scheduling sports events. The related work provides a more general overview of research that has been done regarding scheduling in sports; on the other hand, this work focuses on creating a mobile application to handle the needs of a school's sports team when it comes to viewing and scheduling events.

6. CONCLUSIONS

The purpose of my application is to replace my school's subpar team scheduling system. When using the application, the users will be required to log in with an account, which helps the application determine whether the user is an admin or a student; based on whether the account belongs to an admin or a student, the application offers different features to use. The admins can create and edit events as well as assign students to specific teams to allow for a more specific list of events dedicated to them. On the other hand, students will only be able to see a list of upcoming events. By providing a better alternative to scheduling sporting events at my school, both coaches and students can hopefully have a more convenient user experience. To test whether the application is effective at providing an intuitive and functional scheduling system, participants were gathered to test the application and take a survey on Google Forms that asked whether the application functioned as intended and was convenient to use. According to the results, the majority of participants agreed that the application could perform its duties effectively. Furthermore, the application's interface appeared to be very intuitive and beginner-friendly to

navigate [15]. However, the application still struggles with some minor issues, as reported by some participants in the free-response section of the survey. For instance, events on the upcoming events page may not show up, which could potentially be caused by the database. As this is a core feature of the application, such a bug should be fixed quickly to provide the best experience possible to the application users moving forward.

While the application can competently serve its purpose, there is still room for improvement. There is a known bug that can cause the event page to not work as intended, which is something that can be analyzed and fixed in the future. The application's user interface could also be greatly improved. Although the functionality of the application seems to be fine, there is not much that has been done in terms of decorating the interface and making it appear more presentable. More work can be done on this in the future so that the application no longer uses default backgrounds and buttons.

I am planning to add a place to store team photos, as I feel that this will be a great quality-of-life addition for users to look back at fond memories. I also plan to fix a bug in which multiple events do not show up at once on the team page.

REFERENCES

[1] Houdaille, Rémi, and Stéphane Gouache. "Shaping HTTP adaptive streams for a better user experience." Proceedings of the 3rd Multimedia Systems Conference. 2012.
[2] Vasantha Raju, N., and N. S. Harinarayana. "Online survey tools: A case study of Google Forms." National conference on scientific, computational & information research trends in engineering, GSSS-IETW, Mysore. 2016.
[3] Bitrián, Paula, Isabel Buil, and Sara Catalán. "Gamification in sport apps: the determinants of users' motivation." European Journal of Management and Business Economics 29.3 (2020): 365-381.
[4] Quincozes, Vagner E., et al. "A Mobile Application for on-Demand Scheduling of Health Services." XVIII Brazilian Symposium on Information Systems. 2022.
[5] Kendall, Graham, et al. "Scheduling in sports: An annotated bibliography." Computers & Operations Research 37.1 (2010): 1-19.
[6] Malm, Christer, Johan Jakobsson, and Andreas Isaksson. "Physical activity and sports—real health benefits: a review with insight into the public health of Sweden." Sports 7.5 (2019): 127.
[7] Essex, Stephen, and Brian Chalkley. "Olympic Games: catalyst of urban change." Leisure studies 17.3 (1998): 187-206.
[8] Kendall, Graham, et al. "Scheduling in sports: An annotated bibliography." Computers & Operations Research 37.1 (2010): 1-19.
[9] Tashildar, Aakanksha, et al. "Application development using flutter." International Research Journal of Modernization in Engineering Technology and Science 2.8 (2020): 1262-1266.
[10] Viennot, Nicolas, Edward Garcia, and Jason Nieh. "A measurement study of google play." The 2014 ACM international conference on Measurement and modeling of computer systems. 2014.
[11] Moroney, Laurence, and Laurence Moroney. "The firebase realtime database." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform (2017): 51-71.
[12] Ahmad, Mohd Shahdi, et al. "Comparison between android and iOS Operating System in terms of security." 2013 8th International Conference on Information Technology in Asia (CITA). IEEE, 2013.
[13] Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app development-a study." International Journal of Computer Applications 179.46 (2018): 49-53.
[14] Varshney, Heena, Ali S. Allahloh, and Mohammad Sarfraz. "Iot based ehealth management system using arduino and google cloud firestore." 2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON). IEEE, 2019.
[15] Jerraya, Ahmed A., and Wayne Wolf. "Hardware/software interface codesign for embedded systems." Computer 38.2 (2005): 63-69.

Can Incremental Learning help with KG Completion?

Mayar Osama
Mervat Abu-Elkheir

Faculty of Media Engineering and Technology, German University in Cairo, Egypt

Abstract. Knowledge Graphs (KGs) are a type of knowledge representation that gained a lot of attention due to their ability to store information in a structured format. This structure representation makes KGs naturally suited for search engines and NLP tasks like question-answering (QA) and task-oriented systems; however, KGs are hard to construct. While QA datasets are more available and easier to construct, they lack structural representation. This availability of QA datasets made them a rich resource for machine learning models, but these models benefit from the implicit structure in such datasets. We propose a framework to make this structure more pronounced and extract KG from QA datasets in an end-to-end manner, allowing the system to learn new knowledge in incremental learning with a human-in-the-loop (HITL) when needed. We test our framework using the SQuAD dataset and our incremental learning approach with two datasets, YAGO3-10 and FB15K237, both of which show promising results.

Keywords: Knowledge Graphs, Question Answering, Incremental Learning, Human in the loop

1 Introduction

Task-oriented dialogue systems have been a big part of our modern life and an active area for research and industry. They aim to chat with the user to understand their needs and achieve a specific task for them, and this could be an actual task like setting the alarm, answering a question, or recommending something for the user. These systems could work as multi-domain systems, covering more than one domain, or closed-domain systems, covering only a single domain. They could achieve the goal by conversing with the user over multiple or single turns.

Task-oriented dialogue systems, in general, either follow a modular pipeline approach consisting of four main modules; the first one is a Natural Language Understanding (NLU) module: which is responsible for extracting the information from the user utterance into the knowledge format the model understands, the second one is a Dialogue State Tracking (DST) module: that is responsible for updating the current state of the dialogue, the third module is the Dialogue Policy (DP): which decides for the following action based on the current state, and the last one is a Natural Language Generator (NLG): that takes action decided upon and generates the response to the user in the form of natural language. Sometimes, the DST and DP are referred to as the Dialogue Manager. The other approach is for the system to work end-to-end.[1]

How the knowledge is represented and understood affects the architecture and the techniques used to reach the objective goal. Consequently, many approaches were proposed in this area, which motivated the construction of many datasets. The knowledge representation of the training dataset affects the techniques used to process and extract the needed information. Some of the most common data representations are:

- Questions and Answers pairs; are the most accessible form of knowledge to construct a dataset, as we only need to collect the previous logs of conversations when it's available.
- Documents; in which the knowledge is stored in paragraphs containing the information needed about the system.

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 103-122, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130510

- Intents and slots values representation is a catalog-like representation to store the information, where the intent detected from the user's query is used to fill its corresponding slot. The slots are the labels each word token has in the text.
- Knowledge Graphs, at which the knowledge is represented as facts, where each fact is a triplet *(h, r, t)* where the head entity h is connected to the tail entity t by relation r.

The most expensive and time-consuming forms of knowledge representation are knowledge graphs or intents and slot values; although they provide the best results since they are handcrafted, especially in closed domain systems, they require experts to tailor the knowledge with the required format. In comparison, having the knowledge represented as pairs of questions and answers is easier to be collected from existing systems' logs. Same for having multi-paragraph documents, which are easier to construct.

Constructing a new dataset is costly and time-consuming. Hence, we propose the integration of existing datasets and trying to represent the same knowledge with different representation forms. This paper focuses on extracting knowledge from question-answer pairs and constructing a graph containing the equivalent knowledge.

Knowledge graphs (KGs) represent the data formally as a set of facts; a fact triplet represents the relation between two entities, so these entities could be viewed as the nodes of the graph and the relations to be the edges connecting them. A fact is also referred to as a triplet (h, r, t), where the head entity h is connected to the tail entity t by relation r. Thanks to their structural representation, KGs are used in many NLP task-oriented tasks, e.g., information retrieval, search agents, question answering, conversational recommender systems, etc...

As for every knowledge representation, KGs have limitations beyond being more expensive to construct. The main task of extracting knowledge from KGs is a link prediction task. Link prediction is the task of predicting the missing entity in a given query depending on whether it's a *(h,r,?)* tail prediction or a *(?,r,t)* head prediction, the model should predict the missing entity to complete the required fact. This link prediction could be done straightforwardly or require multiple hops between a few facts in the KG until we reach the missing entity. The issue of not being able to retrieve the missing entity right away motivated another active task in the field of KGs, which is the task of reasoning over KGs. Reasoning over KGs aims to obtain new facts from existing knowledge. Chen et al. [2] reviewed over 11 different approaches for reasoning over KGs, which is an active task for KGs because KGs suffer the limitation of incompleteness.

The incompleteness issue exists in most datasets as it's hard to capture all the needed knowledge of a given topic, especially when designing an open-domain system; it's almost impossible to cover everything when creating the knowledge base at the beginning. Although reasoning over KGs helps reduce that gap, all the data to be added would be at some point, and no new knowledge could be added. Hence, a dynamic system is needed to add new knowledge when needed. We address this issue by having a human in the loop.

A human in the loop is used in different fields to achieve different tasks [3,4,5,6]. The idea behind it is to deploy the model and let it work as it should, and when the model needs new knowledge or makes a mistake, the human/expert could interfere and update the model with the needed knowledge. In our approach, we achieve this with an incremental learning module. But having a human in the loop pops out an important question, does this contradict the main aim of artificial intelligence and machine learning to automate a given task and remove the human from the loop? And the answer is it's a trade-off. For a particular closed-domain system, it is very doable to construct a complete knowledge base from the beginning that the model could be trained on and answer any given ques-

tion accordingly. But in open-domain systems, constructing a complete knowledge base is almost impossible. And assuming there was a way to collect all the relevant knowledge from multiple sources would require intensive training, which would be very expensive. It might also be that not all the knowledge gathered is needed, which might backfire and affect the model's performance. Hence, having a human in the loop only when needed is an acceptable middle ground in our approach, as we start by extracting the knowledge in an end-to-end manner and only refer to the human in the loop when needed to reduce the gap between the knowledge collected. Eventually, the dependency on the human in the loop decreases till it's not needed anymore.

This paper aims to address the following three issues; 1) Constructing KGs from question-answering datasets in an end-to-end manner, 2) Having a dynamic knowledge base system to learn new facts and allow the model to learn new entities, and 3) Reducing the gap of having 1:n or n:1 relations in the KG for existing KG embedding models.

2 Background and Related Work

2.1 Knowledge Graph Embedding and Link Prediction

KGs could be formally represented as a set of facts, such that a fact consists of a triplet *(e_1, r, e_2)* or *(head, relation, tail)*; i.e. the subject e_1 is connected to the object e_2 through this relation r, where e_1 and e_2 belong to the set of possible entities and r belongs to the set of possible relations. *KG=*$\{$*(e1,r,e2)*$\|$ *e1,e2* $\in \boldsymbol{E}$ *and* $r \in \boldsymbol{R}\}$

Link prediction is the task of predicting the missing entity given a source entity and a relation. It could be a head prediction where the source entity is the tail entity, and the missing entity is the head *(?,r,t)*, or a tail prediction where the source entity is the head, and the missing entity is the tail *(h,r,?)*.

Many approaches have been proposed to achieve this link prediction task, some focused on observable features such as Rule Mining [17][16][38][24] or the Path Ranking Algorithm [31][32], and others focused on capturing latent features of the graph by using different embedding techniques. In our paper, we are mainly focusing on the KG embedding approaches.

In general, how the task of link prediction works with KG embedding models is by defining a scoring function ϕ that indicates the probability of the given fact being true. As shown in Equation 1, for a given tail prediction, the model should output the entity e, which returns the highest score from the scoring function.

$$t = \arg\max_{e \in \boldsymbol{E}} \ \phi(h, r, e) \tag{1}$$

Rossi et al.[7] provided a very useful comparative analysis for many of these link prediction approaches for KGs. They classified these models into three main categories:

1. Tensor Decomposition Models, where the task of LP is considered a tensor decomposition task, as these models process the KG as a 3D adjacency matrix or a 3-way tensor that is only partially observable due to the KG incompleteness. This tensor is then decomposed into low-dimensional vectors, which are used as the embeddings for entities and relations.
   (a) Bilinear Models: Given a head embedding $h \in R^d$ and a tail embedding $t \in R^d$, these models usually represent the relation embedding as a bi-dimensional matrix $r \in R^{d \times d}$, where the scoring function computes the result of the product of the three matrices $\phi(h, r, t) = h \times r \times t$. One of the most commonly used models of this class is ComplEX[8].

   (b) Non-Bilinear Models combine the head, relation, and tail embeddings of composition with approaches different from the strictly bilinear product. e.g., HolE[9] computes the circular correlation between the embeddings of head and tail entities and then applies the matrix multiplication with the relation embedding.

2. Geometric Models, on the other hand, view relations as geometric transformations in the latent space where the fact score is represented as the distance between a resulting vector of processing the head and the relation and the tail vector.
   (a) Pure Translational Models represent entities and relations as one-dimensional vectors of the same length, where the added distance between the head embedding and relation embedding should result in the position closest to the tail embedding. TransE[10] was the first proposed model using a pure translational approach, and due to this nature in calculating the score, TransE cannot correctly handle one-to-many and many-to-one relations, as well as symmetric and transitive relations.
   (b) Translational models with Additional Embeddings may associate more than one embedding to each KG element. For instance, CrossE[11] is considered one of the best models of this class. CrossE learns an additional relation-specific embedding with each relation c_r that is then combined with the head and the relation to be used in the translation.
   (c) Roto-Translational Models that perform rotation-like transformations either in combination or in alternative to translations, e.g., RotatE[12] represents relations as rotations in a complex latent space.

3. Deep Learning Models use deep neural network layers to extract the features from the input by fine-tuning the weights and the biases of the neurons of these layers along with learning the KG embeddings.
   (a) Convolutional Neural Networks may contain one or more convolutional layer(s). The task of those layers is to loop over the input with convolution techniques by applying low-dimensional filters to allow the model to extract the needed features during the training phase. Then a Dense layer is used to process the output of the convolution to get the score of a given fact. Examples of CNN models for KG embeddings are ConvE[13], ConvKB[14], and ConvR[15].
   (b) Capsule Neural Networks consist of capsules that are composed of groups of neurons that aim to encode specific features of the input. The main difference is that capsules allow the model to encode those features without losing spatial information, unlike convolutional networks. e.g. CapsE[16].
   (c) Recurrent Neural Networks (RNNs), which consist of recurrent layers that are known for their abilities to process and encode sequential data, e.g., RSN[17].

2.2 Knowledge Graph Construction

The task of constructing a knowledge graph is usually done by experts to ensure the right format and cover the correct information, which is a very time-consuming and expensive process. Hence, many approaches have been proposed to try and automate this process.

Some approaches focused on the potential of language models, as they get to learn linguistic knowledge during training and their ability to store relational knowledge between the training data. Language models are known to have the ability to implicitly encode massive amounts of knowledge, to be used for different tasks like question-answering, text summarizing, etc. One of the main advantages when using language models is that they don't require a fixed schema or human annotations, which allows them to support open-domain questions and the ability to extend to more data.

Language models could answer queries that are structured as "fill in the blank" cloze statements because of their masking mechanism during training, at which the model is required to learn to fill the word at the masked position with the correct word. Petroni et al[18] provided an analysis for language models to test their ability on factual and commonsense knowledge, where the facts they used were either triplet subject-relation-object or question-answer pairs. Each fact is converted to a cloze statement which is then queried to the language model for a missing token and accordingly evaluated that model. They used Google-RE[1], T-rex[19], ConceptNet[20], and SQuAD[21] datasets.

Their results showed the potential of BERT, as it performed well on open-domain questions, and results showed that it contains relational knowledge that could be competitive with traditional NLP methods. Traditional NLP methods are known to contain the best information extractions on fixed schema, which is not always available. Also, they work in a complex pipeline to achieve entity extraction, coreference resolution, entity linking, and relation extractions. The pipeline architecture makes them vulnerable to error propagation and accumulation.

Another approach that took advantage of language models in KG construction is MAMA[22]. They argued that language models are considered an open knowledge graph, as having a language model and a textual corpus, they could generate a knowledge graph relevant to that corpora through a two-staged architecture; Match and Map. At the match phase, the model generates a set of candidate facts from the corpus using Beam Search, where the goal is to match the knowledge stored in the pre-trained language model with the facts in the corpus. During the map phase, the generated facts are mapped to a fixed and open schema to generate the final knowledge graph. MAMA was not the only proposed approach to generate KGs using language models. Swamy et al.[23] followed the same property in language models to answer "fill in the blank queries" cloze statements as LAMA[18]. They first used this property to extract all statements that contain relevant knowledge using the masking property, then added an extraction step using a hybrid SpaCy[2] and Textacy[3] approach to extract the relevant triplets and construct the KG, illustrated in Figure 1. Since their approach works in an unsupervised end-to-end manner, it was vulnerable to inconsistency, as the output KG would depend on the structure of the statements in the corpus and the generated statements after the cloze querying to the language model.

BertNet[24] tried to address this issue along with the dependency of having existing massive data to learn from. They proposed to apply a paraphrasing stage before extracting the triplets, that way, there would be a more diverse set of alternatives to generate the entities and the triplets from. And to handle the resulting issue of having a large search space after the paraphrasing, they proposed a search and scoring strategy to balance the accuracy and coverage of the output.

Garg et al.[25] decided to go in a different direction and examine whether or not language models could capture graph semantics and if language models and graphs could work interchangeably. The objective was if a language model took a graph as an input can it output the same graph while maintaining the same semantics? They pointed out that due to the nature of language models, they take their input as a form of distributed representations or vectors, and to pass the graph as an input to a language model; it needs

[1] https://code.google.com/archive/p/relation-extraction-corpus/

[2] SpaCy is a free, open-source library for NLP in Python. It's written in Cython and is designed to build information extraction or natural language understanding systems

[3] Textacy is a Python library for performing a variety of natural language processing (NLP) tasks, built on the high-performance spaCy library. With the fundamentals — tokenization, part-of-speech tagging, dependency parsing, etc.

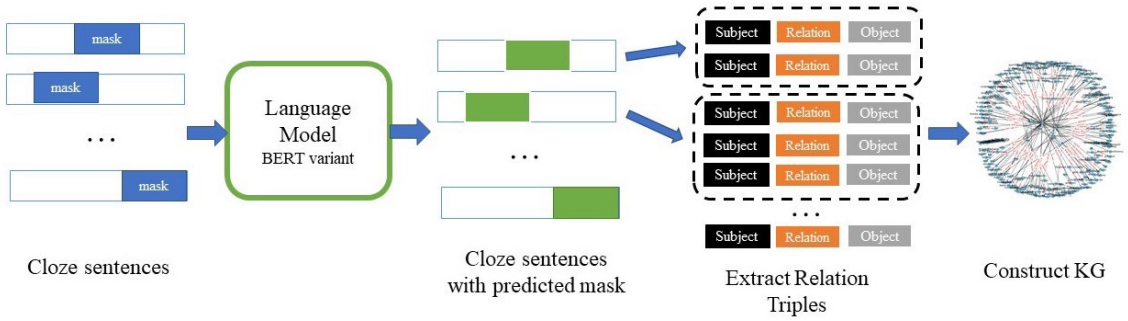


Fig. 1. Pipeline to generate knowledge graphs from language models from [23]

to be compressed into a vector representation which would affect its structural semantics. From this point, the experiments showed that transformer models could not express the full semantics of the input knowledge graph.

Language models have shown so much potential when it comes to NLP tasks, and they are viewed as more than just processing models but potentially a pre-trained knowledge base. However, they store the knowledge in a form of a black box which still makes it difficult to access or edit. AlKhamissi et al.[26] provided a full analysis of language models and whether or not they could be considered a walking database.

2.3 Incremental Learning

The aim of machine learning is to allow the system to behave as a human, and humans keep updating their knowledge either by trial and error, which is represented in the training phase, or by external knowledge and asking for the right answer, which is represented in the incremental learning phase of our approach.

Incremental learning aims to keep the system updated and gives it the ability to learn new knowledge without having to retrain it on all the previous knowledge. There are three main ways to apply incremental learning; the first one is to have an interactive environment in which the system can get feedback and change its behavior accordingly, the second approach is to allow the system to get its knowledge from an external source which in our scope is shown with the human/expert in the loop, and the third is using both approaches together. The concept of incremental learning has been adapted to different domains, and fields [3,4,5,6].

Wang, Weikang, et al. followed the idea of building a training dataset consisting of five sub-datasets (SubD1, SubD2, SubD3, SubD4, and SubD5). The model is first trained on these sub-datasets one at a time till it is ready for deployment. But instead of using a traditional task-oriented dialogue system, they proposed a very interesting approach which is the Incremental Dialogue System (IDS)[27].

IDS uses the concept of adding a human-in-the-loop; the motivation of this approach is to reduce the non-relevant responses of the dialogue system. One of the main issues in dialogue systems is the irrelevant replies; since the model is usually trained on certain dialogues, it usually remembers "most" of the replies, but in many cases, if the user asks a new question that the system didn't see before then the response would be irrelevant. To fix that, first, they calculate the confidence level between the model's reply and the user's query; if it's high, then respond with the generated system's reply, but if it has a low confidence level meaning the reply is irrelevant, then let a human expert reply this time to the human. After asking the human in the loop to reply, the model needs to learn the

answer to this user query, so they use incremental learning to do so, as shown in Figure 2. Their approach consists of mainly 3 modules:

1. Dialogue Embedding Module: at which the user utterance is embedded using Gated Recurrent Unit (GRU) based bidirectional Recurrent Neural Network (bi-RNN), and on top of it, they use self-attention layer to improve the encoding.
2. Uncertainty Estimation module: at which the confidence level between the user's utterance and the system response(s) is calculated.
3. Online Learning module: this module is only used when the confidence level from the second module is low, and none of the candidate responses are relevant to the user utterance. In this case, a human expert is involved to respond on this utterance, and the system should be updated with the given utterance and its proper response using incremental learning.

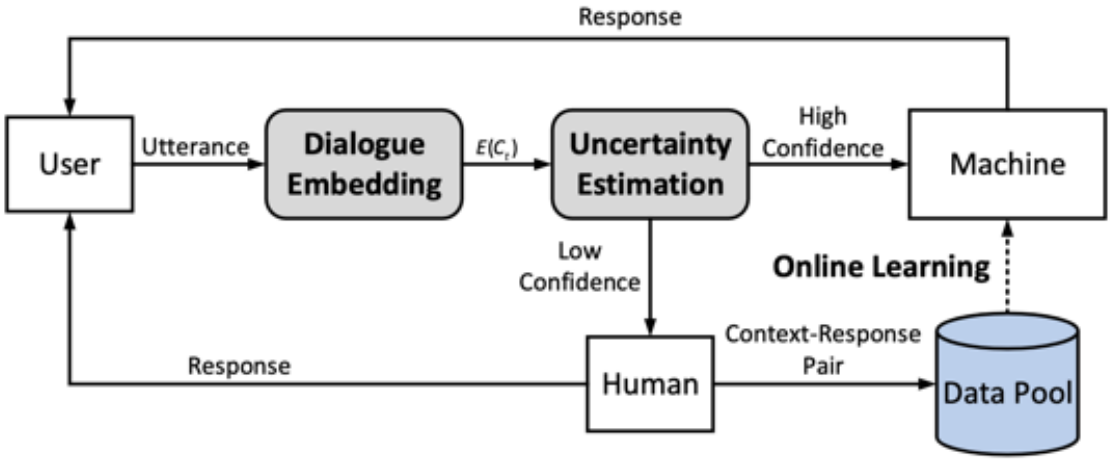


Fig. 2. Overview of IDS [27]

The same idea of adding a human-in-the-loop was discussed with a different approach by Rajendran et al. [28]. Their goal was to maximize task success in task-oriented dialogue systems while minimizing the involvement of a human expert in the loop. This is very similar to the previous approach except that here they used reinforcement learning instead of incremental learning.

The system has mainly three elements: the model M, a neural dialogue model which is trained for goal-oriented dialogues. The classifier C, which is a neural classifier that uses reinforcement learning to learn, and the human H, which is the expert in the loop. As discussed earlier, since it's very unlikely to get users' utterances to be similar to the ones in the training dataset, therefore the model's response might not be correct. The idea is that with every user utterance, the classifier gets to pick between the model and the human; this way, the classifier learns with trial and error, using a reward and punishment system.

If the classifier picked the model and the model answered correctly, the classifier gets rewarded with a high reward, but if the model answered incorrectly, then the punishment is high. Finally, if the classifier picked the human to answer, in this case, we always assume that the human response is correct, then the classifier is rewarded with a low reward. With trial and error, the classifier will try to maximize its reward by increasing the task success and minimizing the need for human.

Our incremental learning approach is mostly inspired by [27,28], both of them worked with the motivation of improving the neural network task-oriented dialogue system by adding a human in the loop to respond whenever the system outputs an invalid/incorrect answer to the user, and takes this expert's response and feeds it back to the system so it would learn the correct answer.

3 Proposed Method

Our approach consists mainly of four modules; *Knowledge Graph Extractor Module*, *Knowledge Graph Embedding Module*, *Incremental Learning Module*, and *Selection Module*. The framework takes place over two phases, the first phase is the training phase, in which we construct the KG and train the model with the first two modules, and the second phase is the deployment phase. First, we construct the KG from a question-answer dataset for the training phase by taking each question and answer and converting them into statements. Then we pass these statements to OpenIE[29], which extracts the facts/triplets from each statement. And finally, we use the constructed KG to train our model.

For the deployment phase, there are two scenarios; the first is that new knowledge needs to be added, so we use the *Incremental Learning Module* directly. The second is when interacting with the user; the user would ask a question, which we pass to the first module to convert into a statement to get the missing entity (*(h,r,?)* tail prediction or *(?,r,t)* head prediction), then we pass this required prediction to the model to retrieve the missing entity. Here we have the following cases;

- if the model could predict a link:
  - This link prediction is valid, so we output it to the user.
  - This link prediction is invalid, so we ask the human in the loop to answer and learn this new link.
- if the model could not predict a link, which might happen if the source entity is not in the entities list that the model is trained on (new entity). In that case, we redirect this task to the human in the loop and feed the model the new fact to update it.

3.1 Knowledge Graph Extractor (KGE) Module

This module aims to extract the KG facts from a Question-Answer dataset. Figure 3 illustrates an overview of this module. Having a question Q and answer A, we first pass Q to lexicalized PCFG parser[30] to extract the parse tree of the grammatical structure of the question.

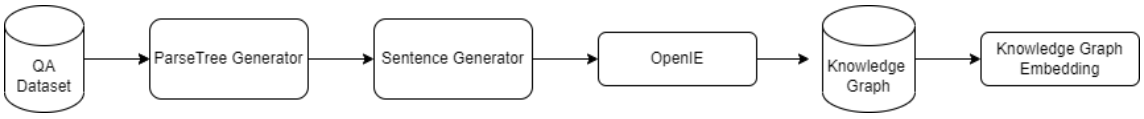


Fig. 3. Extracting Facts from QA dataset

Generating a parse tree transforms a natural language sentence/question into its equivalent syntactic tree form representing the grammatical structure. This process includes identifying groups of words (phrases), part of speech tags of these phases/words, and dependency labels. This is done in an unsupervised manner by using a pre-trained parser

provided by the Stanford Natural Language Processing Group[4], which follows the English Penn Treebank[5]. Example is shown in Figure 4.

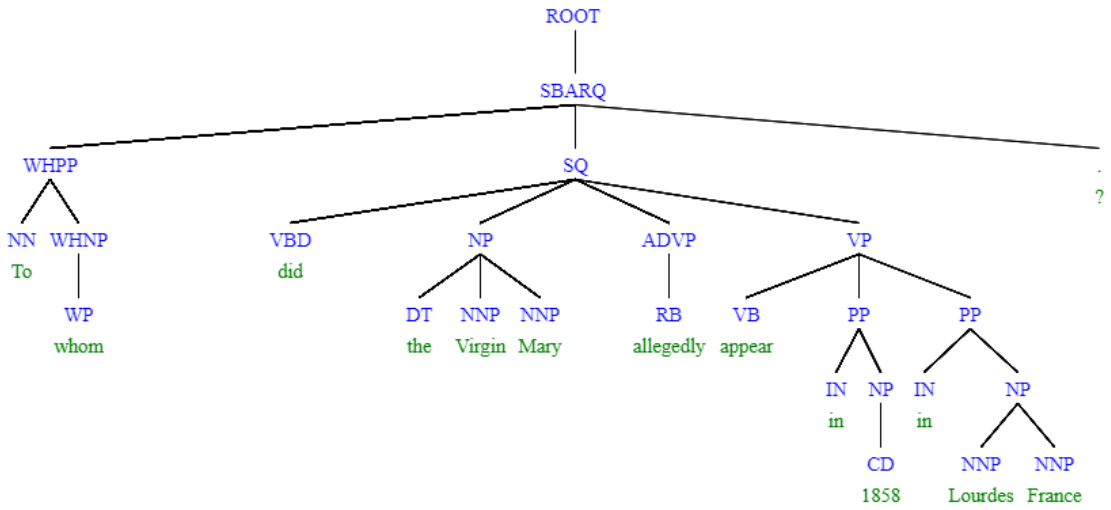


Fig. 4. Output parse tree for "To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France?"

After generating the parse tree, we use it to rewrite the question in the form of sentence S; by looping over the tree nodes. First, we decide where to place the given node in the sentence according to the node type. Then we remove the question header, and according to the type of question, we put the correct prepositions, if needed, before the answer.

This sentence S would contain a ***blank*** where the answer should be, so we simply take A and replace it with ***blank***. We use the same module for the user's question to extract the fact with the missing entity to pass to the model for the link prediction while leaving the ***blank*** placeholder to identify the prediction needed.

Once we have the sentence ready, we pass it to OpenIE[29] to extract all the possible fact triplets from the given sentence. The Open Information Extraction (OpenIE) is an unsupervised annotator that extracts the relation triples from a given system by splitting them into clauses. Each clause is used to generate a set of shorter sentence fragments. From these shorter sentences, it's easier to extract the triples [31,32,29,33]. One of the main advantages of OpenIE is that the extracted facts are humanly readable, which makes it easier to generate the answer from the knowledge graph. The result of this module should be a knowledge graph $KG=\{(e1,r,e2) \| \ e1,e2 \in \boldsymbol{E}$ *and* $r \in \boldsymbol{R}\}$ which contains the information presented in the question-answer pairs in the original dataset, which refers to the first contribution in the Introduction 1.

3.2 Knowledge Graph Embedding Module

We used TransE as our KG embedding model for our experiments. TransE constructs the embeddings for the entities $\boldsymbol{E}$ in $\|\boldsymbol{E}\|^k$, where k is the dimension of the embeddings which is passed as a hyperparameter to the model, and same for the relations $\boldsymbol{R}$ in $\|\boldsymbol{R}\|^k$. Initially, these embeddings are randomly initialized, and the model gets to learn and fine-tune these

[4] https://nlp.stanford.edu/software/lex-parser.html
[5] http://surdeanu.cs.arizona.edu//mihai/teaching/ista555-fall13/readings/PennTreebankConstituents.html

embeddings by minimizing the margin-based loss equation 2:

$$\mathcal{L} = \sum_{(e1,r,e2)\in S} \sum_{(e1',r,e2')\in S'_{(e1,r,e2)}} [\gamma + d(e1 + r, e2) - d(e1' + r, e2')]_+ \quad (2)$$

Where $\gamma > 0$ is a margin hyperparameter and the energy of a triplet $d(e1' + r, e2')$ is for some dissimilarity measured. The $[\gamma + d(e1 + r, e2) - d(e1' + r, e2')]_+$ donates only the resulted values that are positive. $S'_{(e1+r,e2)}$ is the set of corrupted triplets, where it takes triplets from the training set and replaces either the head or tail in the triplet by a random entity but not both at the same time. Shown in equation 3:

$$S'_{(e1+r,e2)} = \{(e1', r, e2)|e1' \in E\} \cup \{(e1, r, e2')|e2' \in E\}. \quad (3)$$

Because TransE is an energy-based model that uses a geometric interpretation of the latent space, it considers that a fact *(e1,r,e2)* holds when the embedding of the tail entity *e2* is close to the sum of the embedding of the head entity *e1* plus some vector that depends on the relationship *r*, i.e., *e1+r≈e2*. However, due to the nature of translation, TransE cannot correctly predict one-to-many and many-to-one relations; a selection mechanism is needed after the TransE prediction to help reduce this gap, further explained in module 3.4.

3.3 Incremental Learning Module

After training the model, we use this module in two scenarios; 1) if there is new information that should be added to the knowledge base, or 2) when the user asks the system a question and the system provides an invalid answer. An overview of this module is illustrated in Figure 5. In the second scenario, the system can not provide a correct answer; hence, we redirect to the human expert to answer this question. According to that answer, we also use it to update the model and KG.

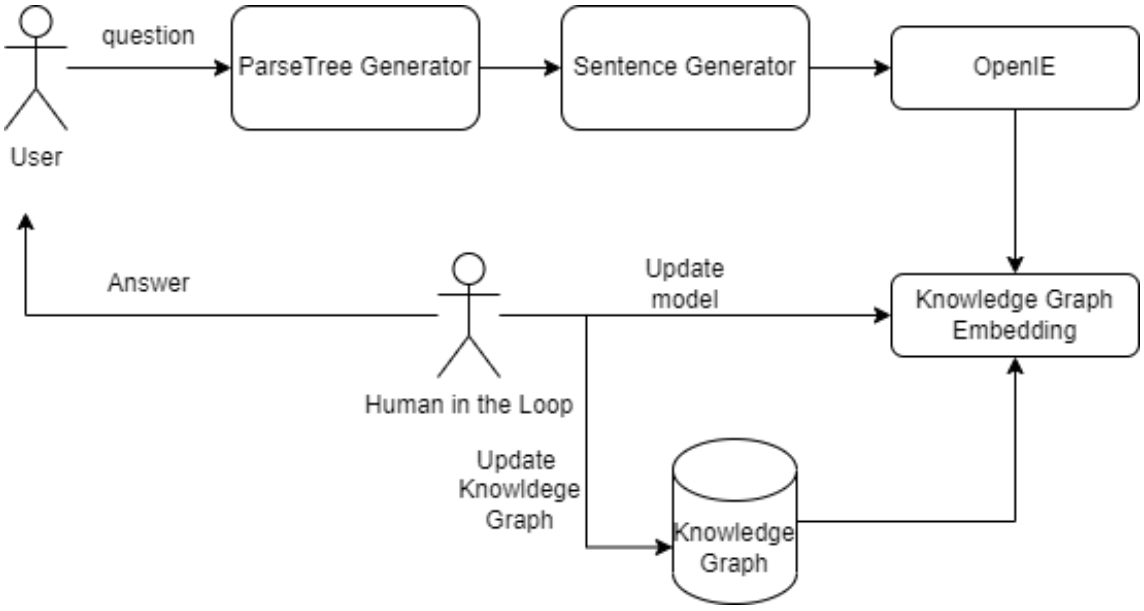


Fig. 5. Incremental Learning with HITL

As illustrated in Algorithm 1, first, we check if the 'new' fact contains any new entities. In case it includes an entity that is not in the entity list of the KG, we first add it to the list of entities by adding a new dimension to the model's entity parameter. Then, this new dimension is randomly initialized and updated by finetuning the model; by feeding it the new fact. The need for this step comes from the formal definition of the KG itself since the embeddings of the KG are dependent on the total number of entities and relations at the construction stage, as explained in subsection 3.2.

If the new fact does not contain any new entity, then we move directly to feeding the model the 'new'[6] fact. This way, the model could keep learning new knowledge without affecting the previously learned knowledge and without retraining from the beginning, which refers to the second contribution in the Introduction Section1.

Algorithm 1 Incremental Learning with the HITL feeding the model (e1,r,e2)

if $e1 \notin E$ **then**
    $E = E \cup \{e1\}$
    $embed(e1) = random(size = k)$ ▷ Randomly initializing embedding for the new entity
    $embed(E) = append(embed(E), embed(e1))$ ▷ Add randomly initialized vector for the new embedding in the embeddings
if $e2 \notin E$ **then**
    $E = E \cup e2$
    $embed(e2) = random(size = k)$
    $embed(E) = append(embed(E), embed(e2))$
train for (e1,r,e2)

3.4 Selection Module

This module is responsible for verifying the answer and selecting the correct answer output by the model since most KG embedding models mispredict the 1:n and n:1 relations, which makes sense because, without any provided context, all the facts are 'valid'. We can address this issue by passing the user's question to the KGE module 3.1 and excluding the common facts between the questions and the valid facts predicted by the model, explained in Algorithm 2. This step reduces the gap of having n:1 or 1:n relations, which refers to the third contribution.

Algorithm 2 Selection Module Algorithm

Input: Question q

$tree = ParseTree(q)$ ▷ Generate the ParseTree q
$sentence\_q = tree_to_sentence(tree)$ ▷ Generate the sentence of q without the answer
$facts = OpenIE(sentence\_q)$ ▷ Extract the facts and the missing fact to predict
$missing\_fact = facts.contain$
$top\_k\_predictions = model.predict(missing\_fact)$ ▷ save top k predictions with the highest score
$output\_fact = top\_k\_predictions - facts$ ▷ Eliminate the common facts between the two sets

Output: $output\_fact$ ▷ Output the fact with the highest score

[6] The reason we put new between " is that the fact that is being fed to the model might not be a new fact, but a fact that the model mispredict.

This way, we have fewer predicted facts; if only one remains, the system outputs that to the user and waits for the user's feedback. If it is not the correct fact that the user is looking for, then the system moves to HITL with the incremental module 3.3. If there is more than one valid fact after the filtration, the model picks the fact with the highest score as the correct output. If it is not the correct answer, move on to the next one until the user finds what they want. Finally, if none of the predicted facts were valid or no facts were left after the filtration, then we move to the HITL with the incremental module 3.3.

For example:*Q: To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France?*

- from Q we form the statement S:
  *S: The Virgin Mary allegedly appeared in 1858 in Lourdes France to **blank***
- from which we extract the following triplets:
  (The Virgin Mary, appear, in 1858)
  (The Virgin Mary, appear, in Lourdes France)
  (The Virgin Mary, appear, ?)
- Then we pass this tail prediction to the model to get the missing entity, which would give us all the relevant facts; technically, all of them are correct/valid.
  (The Virgin Mary, appear, in 1858)
  (The Virgin Mary, appear, in Lourdes France)
  (The Virgin Mary, appear, Saint Bernadette Soubirous)
- Since we know that the first two facts are already mentioned in the question, this is our context to pick the right answer; hence the system would output *(The Virgin Mary, appear, Saint Bernadette Soubirous)* which is the correct answer.

4 Experiments and Results

In this section, we first start by introducing the datasets that we used in subsection 4.1, then we present the evaluation metrics that we used in subsection 4.2, and finally, we discuss our experiments and their results on the given datasets in subsection 4.3.

4.1 Datasets

To validate our approach, we used the SQuAD[21] dataset, which is considered one of the benchmark datasets for the Question-Answering tasks. The SQuAD is a collection of question-answer pairs created from Wikipedia articles by humans through crowd-sourcing, which makes it very diverse. We used 70,000 question-answer pairs for our approach. After running the parser and the sentence generation step, we ended up with 68,445 sentences, which resulted in 100500 entities and 14783 relations, and 71194 facts after running the KGE module[7]. We used the cross-validation function by sklearn[8] to split our dataset into training, validation, and testing, each containing 56955 facts, 7119 facts, and 7120 facts, respectively. For the unseen experiments, we saved 50 entities away from the model among the corresponding 58 facts from the training, validation, and testing sets.

We compared our approach with other approaches using language models from the literature review to construct the KG. We used the same framework from the methodology to convert the question-answer pairs into sentences. Still, instead of adding the answer to the sentence, they added '[MASK]' for the language model to learn how to fill it with the correct answer. We added this step to achieve the same task: constructing the KG

[7] We will refer to the KG dataset extracted from SQuAD with SQuADKG.
[8] https://scikit-learn.org/stable/

from question-answer pairs, which is the main focus of our paper. We used the approach presented by[23] for this experiment with Roberta with their custom SpaCy and Textacy method to clean the KG. Each approach has its pros and cons, which we consider a designer choice for this step. Table 1 shows the results for running both methods on 57,355 statements as the parser and sentence generator output. On average, the LM approach took 03:33:52 to process 10,000 sentences, using GPUs offered by Colab Pro at the time we conducted this experiment which was Tesla T4, while for OpenIE, we used average processing power; Intel(R) Core i7-8750H CPU. Despite the difference in the computation power, the LM approach took a significantly longer time to process all the sentences and still extracted significantly fewer facts than the OpenIE approach; details about the LM experiment are discussed in Appendix section5.1.

Using The LM approach would provide a cleaner graph in terms of entities and relations. On the other hand, it requires multiple hops to reach the answer, which would affect the generation of the answer, as we have to keep track of the path taken to generate the answer. Although using OpenIE would make it easier to generate the answer, we might find facts like *(The National Archives, make strides towards making its holdings more widely available In 2006)*. These facts make it harder for the embedding model to understand the features and have many relations and entities with the same semantic meaning. However, this issue could be fixed by adding another layer of filtering the entities and relations before adding them to the graph.

The last point of comparison is the execution time and resource needed for each, which is presented in the last row in Table 1; the time difference between the two methods is noticeable.

POC	LM	OpenIE
count entities	6,778	87,498
count relations	5,174	14,118
count facts	12,251	59,636
Execution time hh:mm:ss	+20:00:00	02:10:27

Table 1. Constructing the KG with different approaches

We continued the rest of the experiments with the dataset generated by OpenIE, as we wanted to reduce having implicit data learned by the language model and work with a more classical approach. In addition, we don't see any improvement done by the LM, so we go with OpenIE as it requires less time and resources and still extracts more facts/knowledge.

To validate the Incremental learning module on its own, we used YAGO3-10[34] and FB15K237[35] since these are the datasets that were included in the OpenKE[36] results. The YAGO3-10[34] consists of 123182 entities and 37 relations; it contains 1,079,040 facts for the training set and 5000 facts for each validation and testing set. We picked this dataset because its entities are associated with at least ten different relations. These relations describe human attributes like association, profession, gender, etc.

The FB15K237[35] consists of 14,541 entities and 237 relations, which form 149678 facts for training, 3992 facts for testing, and 543 facts for validation.

To simulate the incremental learning with human-in-the-loop, we removed **100** entities from each of the datasets to save them for the unseen dataset along with their relevant facts from the training, validation and testing sets. These facts are then used to simulate the unseen queries for the model and apply the incremental learning approach. In addition to those, we saved any mistakenly predicted link while testing the model after training

so that we get to add these incorrect facts to the testing set. For YAGO3-10, the total number of unseen facts was 1935 and 4992 mispredicted facts. For FB15K237, the total number of unseen facts was 77, and 20319 mispredicted facts.

These three datasets are different in size, i.e., the total number of entities, relations, and the number of 1:n or n:1 relations. Since SQuADKG was constructed end-to-end, the number of relations is significantly larger than the number of relations of the other two datasets.

4.2 Metric

Mean Rank (MR): It is the average of the obtained ranks. Its range is between 1 and $\|\boldsymbol{E}\|$. As the value gets closer to 1 than $\|\boldsymbol{E}\|$, this means that the performance is improved. Because this metric is very sensitive to outliers, it's usually not used just by itself.

Mean Reciprocal Rank (MRR): It's value ranges between 0 and 1. It is the average of the inverse of the obtained ranks, and hence the higher the value is, the better the model results.

Hits@K (H@K): It is the ratio of predictions for which the rank is equal or lesser than a threshold K; its range lies between 0 and 1, where closer to 1 is better. Common values for K are 1, 3, 5, and 10. The higher the H@K, the better the model results. We mainly focus on K=10, which shows the proportion of correct entities ranked in the top 10.

4.3 Results and Analysis

In Table 2, the *TransE* row shows the results of training TransE on the YAGO3-10 dataset. After saving some entities and facts for the incremental learning part, the model was trained for 500 epochs. The results of running the model on the given test file were 56.4% for hits@10.

We saved the mispredicted facts to apply the incremental learning part to the model; this experiment aimed to simulate the human in the loop when the model mispredicted a particular fact, so we saved the facts that we knew that the model needed to 'learn.' As expected, the results after this type of incremental learning improved the model's performance on these facts. The results are shown in Table 2 in the *TransE+* row.

For the main experiment to feed the model entities that it has not seen before, we applied the incremental learning approach to this setting while adding a new dimension for this entity and initially randomizing its values till the model learns it. Then we went back to test the model's performance on the unseen facts, as shown in Table 2 in the *TransE++ unseen entities* row that the model could recognize these entities and facts correctly after this learning.

For the final experiment, we wanted to test if adding these new entities would affect the model's performance on the original data of the training phase. So we merged all the facts from the three experiments (the original test set, the mispredicted facts, and the unseen facts). Row *TransE++ all* in Table 2 shows the results of this experiment that learning new entities does not affect TransE's ability to recognize the original trained entities and facts.

We repeated those four experiments with the same order for the FB15K237 dataset, and their results are shown in Table 3. *TransE* refers to training the model with the training dataset after removing the unseen facts and showing the results of running on the test set. *TransE+* refers to the model after the incremental learning on the mispredicted facts, which was tested on those mispredicted facts after the incremental learning. One of the reasons why the performance did not reach high results for YAGO3-10 is the number

Model	MRR	MR	hit@10	hit@3	hit@1
TransE	0.342691	1704.469727	**0.564413**	0.402080	0.226445
TransE+	0.895744	2.000801	**0.988391**	0.960268	0.828263
TransE++ unseen entities	0.788238	9.513178	**0.928165**	0.848579	0.707494
TransE++ all	0.796323	16.453110	**0.940918**	0.887318	0.697302

Table 2. Results of TransE on YAGO3-10.

of 1:n and n:1 facts. *TransE++* refers to the model after incrementally learning the unseen entities and their relevant facts. Here, we show the results of testing the model on just the unseen data, shown in row *TransE++ unseen entities*, and testing the model on all the data, shown in row *TransE++ all*.

Model	MRR	MR	hit@10	hit@3	hit@1
TransE	0.279723	233.155823	**0.466185**	0.317886	0.184267
TransE+	0.489348	125.041397	**0.622388**	0.527722	0.412699
TransE++ unseen entities	0.744715	10.077922	**0.831169**	0.805195	0.681818
TransE++ all	0.452894	243.779419	**0.576355**	0.487343	0.382188

Table 3. Results of TransE on FB15K237 for the four scenarios.

The drop-off in the performance of the model after training the model on new entities in some cases, for instance, in Table 3 when it dropped from 62% (*TransE+*) to 57% (*TransE++ all*), is acceptable in our approach because initially, the model cannot predict a link of an entity it has not been trained on, but now it has this capacity for these new entities. The objective of our approach is to allow the model to learn new facts and new entities while still being able to recognize the original facts with acceptable accuracy. So this is a trade-off between being unable to output a fact if it contains an entity it hasn't seen and slightly affecting the prediction of existing facts.

After constructing its knowledge graph entities and facts, we conducted the same experiments on the SQuADKG dataset. Results are shown in Table 4. The first scenario *TransE* was to train TransE while keeping the unseen entities and their corresponding facts outside the training set, the loss while training was 0.007913. We tested the trained model on the test set, which it had not seen before, not during the training or validation. The accuracy for hit@10 was almost 2.9%, which was perfect to showcase the effect of our incremental learning effect. The model could not predict the test facts correctly because we constructed the KG manually, which means that the graph would be vulnerable to being unconnected, especially after removing some for the unseen scenario and some for the test and validation sets.

Model	MRR	MR	hit@10	hit@3	hit@1
TransE	0.015392	42327.921875	**0.029234**	0.017569	0.007309
TransE+	0.559917	62.463951	**0.911174**	0.865777	0.245678
TransE++ unseen entities	0.893739	1.844828	**0.982759**	0.956897	0.844828
TransE++ all	0.562405	62.002300	**0.911683**	0.865886	0.250244

Table 4. Results of TransE on SQuADKG for the four scenarios.

The second scenario *TransE+* was to teach the model these mispredicted/new facts and test it on the same set; here, we notice that the model was able to recognize these facts while still recognizing the other facts correctly as the score of the metrics got better, as the accuracy improved to 91% for these new facts (but not new entities). And finally, in the *TransE++* where we teach the model entities and facts they have not seen before during the training phase, the prediction score when examining the model on the unseen facts was 98%, and to make sure that learning these new entities still did not affect its overall performance we tested the model on a merged test set that combines everything, at which it is shown that the model was still able to predict the facts with 91%.

When using ComplEx on SQuADKG, we noticed that it performed poorly with an accuracy of less than 1% (0.009979) on the test set with a loss of 8.81. This could be due to the fact that the KG is constructed in an end-to-end unsupervised manner which highlights the problem with incompleteness, in addition to removing over 20% of the facts for validation and training, which makes the data significantly incomplete. Another reason is that ComplEx requires the relation r between the entities to be diagonal, as discussed in [7], which is not guaranteed in our case. Although TransE initially did not perform well either on this set, the difference in the scoring function and embeddings for each of the models made TransE more convenient to approach.

5 Conclusion and Future Work

In this paper, we start by introducing our problem statement, which consists of two parts; the first is constructing a knowledge graph from a question-answering dataset since QA datasets are more available and easier to collect, unlike KGs. However, QA representations lack the structural representation that exists in data representations such as KGs. Another advantage of the KG representation is its ability to explicitly store the data, unlike deep language models, where data is stored implicitly and appears as a black box, making it harder to manipulate.

For this problem, we propose our Knowledge Graph Extraction Module 3.1, which takes the question and extracts its grammatical structure using a parse tree generator. Then, from the generated tree, we rearrange the question in sentence form and add the answer in the correct position in the sentence. And the last step is to use OpenIE to extract the fact triples from the generated sentence. For the experiments, we used question-answer pairs from the SQuAD[21] dataset; from 70,000 question-answer pairs, we extracted 100500 entities and 14783 relations, and 71194 facts.

The second part of our problem statement addresses the incompleteness problem, which is present in most of the knowledge representations but mainly highlighted in knowledge graphs because it is shown as missing links in the graph, making it harder for the model to reach the correct answer. For this problem, we propose an incremental learning approach that allows the model to learn from a human expert in the loop that would feed the model new knowledge when needed. This Incremental learning module also allows the model to learn new entities, not in the original training entity vocab list. The need for this step is shown when the user asks a link prediction query with an unseen entity or when the human-in-the-loop tries to feed the model a new fact that contains an unseen entity; in both cases, the model would typically raise an error.

We conducted the same experiments on three different datasets to verify our framework. The first dataset is the extracted KG from the SQuAD dataset, which we refer to as SQuADKG. The second dataset is the YAGO3-10[34] dataset which is one of the benchmark datasets that contains 123,182 entities, 37 relations, and 1,179,040 triples, and the

third dataset is the FB15K237[35] which has 310,079 triples with 14,505 entities and 237 relations. Finally, for the embedding model, we used TransE[10]. We removed 100 entities with their related facts from the training, testing, and validation set to create the unseen set for the incremental learning experiments. For YAGO3-10, after training the results on the test dataset, TransE's performance was 56.44%. Then, after the incremental learning on the mispredicted facts, it reached 98.83%. Finally, for the last experiment on the unseen entities facts and testing on the original test set, the performance was 94.09%.

For FB15K237, the results were different due to the difference in the structure of the dataset itself. The experiments results on the test dataset were: initially 46.61%, after the incremental learning on the mispredicted facts, it reached 62.23%, and for the unseen entities facts with the Incremental Learning module, it achieved 83.11% on the unseen data. But on the complete test set, including the unseen data, the performance was 57.63%. The dropout from 62% to 57% is acceptable in our approach as it's a trade-off between not having the needed vocab and mispredicting some facts since the incremental learning approach can handle this issue when required.

Lastly, the SQuADKG dataset shows the most significant results, going from 2.92% in the first experiment to 91.16% in the final experiment on the unseen data. This improvement makes sense since we constructed the KG and then removed over 20% of the dataset for the test, validation, and unseen data, leaving the training dataset significantly incomplete. That's why the new knowledge fills the missing gaps in the constructed graph.

Our final contribution was to address the 1:n and n:1 link prediction problem that is known to be one of TransE's limitations, along with other models. We propose a selection method that eliminates the facts mentioned in the question to reduce the number of possible answers and supposedly output the response with the highest score. The objective was to try and reduce the complexity of each module, aiming to save resources while maintaining acceptable results.

As for the limitations of our approach that we aim to target in our future work, starting with the resources and time efficiency, we found that the parser takes the most time. Although we tested this module on limited resources, we still aim to find a more efficient way to achieve the same results with a somewhat optimized approach. Regarding the embedding model, we want to experiment with different models and verify the effect of incremental learning, with the required tuning, on each of them. The incremental learning approach would show promising results with reinforcement learning models because they naturally learn by interacting with the environment. One of the models that we are currently experimenting with is the dual agent approach[37] to navigate the KG.

Finally, we aim to expand our approach to handle multi-turn conversations by adding another layer that would keep track of the current dialogue state and use it to filter the possible answers. That would help navigate the KG better without significantly increasing the complexity.

References

1. Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10):2011–2027, 2020.
2. Xiaojun Chen, Shengbin Jia, and Yang Xiang. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, 141:112948, 2020.
3. Samuel Budd, Emma C. Robinson, and Bernhard Kainz. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis*, 71:102062, 2021.
4. Andreas Holzinger, Markus Plass, Katharina Holzinger, Gloria Cerasela Crişan, Camelia-M. Pintea, and Vasile Palade. Towards interactive machine learning (iml): Applying ant colony algorithms to

solve the traveling salesman problem with the human-in-the-loop approach. In Francesco Buccafurri, Andreas Holzinger, Peter Kieseberg, A Min Tjoa, and Edgar Weippl, editors, *Availability, Reliability, and Security in Information Systems*, pages 81–95, Cham, 2016. Springer International Publishing.

5. Yiwei Yang, Eser Kandogan, Yunyao Li, Prithviraj Sen, and Walter S Lasecki. A study on interaction in human-in-the-loop machine learning for text analytics. In *IUI Workshops*, 2019.
6. Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*, 2016.
7. Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(2):1–49, 2021.
8. Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.
9. Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.
10. Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.
11. Wen Zhang, Bibek Paudel, Wei Zhang, Abraham Bernstein, and Huajun Chen. Interaction embeddings for prediction and explanation in knowledge graphs. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 96–104, 2019.
12. Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.
13. Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
14. Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. A novel embedding model for knowledge base completion based on convolutional neural network. *arXiv preprint arXiv:1712.02121*, 2017.
15. Jill Burstein, Christy Doran, and Thamar Solorio. Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.
16. Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, Dinh Phung, et al. A capsule network-based embedding model for knowledge graph completion and search personalization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2180–2189, 2019.
17. Lingbing Guo, Zequn Sun, and Wei Hu. Learning to exploit long-term relational dependencies in knowledge graphs. In *International Conference on Machine Learning*, pages 2505–2514. PMLR, 2019.
18. A. H. Miller P. Lewis A. Bakhtin Y. Wu F. Petroni, T. Rocktäschel and S. Riedel. Language models as knowledge bases? In *In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019*, 2019.
19. Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
20. Robyn Speer, Catherine Havasi, et al. Representing general relational knowledge in conceptnet 5. In *LREC*, volume 2012, pages 3679–86, 2012.
21. Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.
22. Chenguang Wang, Xiao Liu, and Dawn Song. Language models are open knowledge graphs. *arXiv preprint arXiv:2010.11967*, 2020.
23. Vinitra Swamy, Angelika Romanou, and Martin Jaggi. Interpreting language models through knowledge graph extraction. *arXiv preprint arXiv:2111.08546*, 2021.
24. Shibo Hao, Bowen Tan, Kaiwen Tang, Hengzhe Zhang, Eric P Xing, and Zhiting Hu. Bertnet: Harvesting knowledge graphs from pretrained language models. *arXiv preprint arXiv:2206.14268*, 2022.
25. Tarun Garg, Kaushik Roy, and Amit Sheth. Can language models capture graph semantics? from graphs to language model and vice-versa. *arXiv preprint arXiv:2206.09259*, 2022.
26. Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. A review on language models as knowledge bases. *arXiv preprint arXiv:2204.06031*, 2022.

27. Weikang Wang, Jiajun Zhang, Qian Li, Mei-Yuh Hwang, Chengqing Zong, and Zhifei Li. Incremental learning from scratch for task-oriented dialogue systems. *arXiv preprint arXiv:1906.04991*, 2019.
28. Janarthanan Rajendran, Jatin Ganhotra, and Lazaros C. Polymenakos. Learning End-to-End Goal-Oriented Dialog with Maximal User Task Success and Minimal Human Agent Use. *Transactions of the Association for Computational Linguistics*, 7:375–386, 07 2019.
29. Mausam Mausam. Open information extraction systems and downstream applications. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence*, pages 4074–4077, 2016.
30. Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st annual meeting of the association for computational linguistics*, pages 423–430, 2003.
31. Swarnadeep Saha et al. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299, 2018.
32. Swarnadeep Saha, Harinder Pal, et al. Bootstrapping for numerical open ie. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 317–323, 2017.
33. Harinder Pal et al. Demonyms and compound relational nouns in nominal open ie. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 35–39, 2016.
34. Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
35. Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, pages 57–66, 2015.
36. Xu Han, Shulin Cao, Lv Xin, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. Openke: An open toolkit for knowledge embedding. In *Proceedings of EMNLP*, 2018.
37. Denghui Zhang, Zixuan Yuan, Hao Liu, Xiaodong Lin, and Hui Xiong. Learning to walk with dual agents for knowledge graph reasoning. *arXiv preprint arXiv:2112.12876*, 2021.

Appendix

5.1 Using Language Model to Generate KG

In this section, we discuss the experiment of constructing the KG using a language model (LM). The objective of this experiment was to compare and explore other approaches for constructing the KG. Starting with running the experiment on the approach proposed in [23] for which they proposed using masked sentences and passing them to the LM to allow the model to learn the features of the given sentence and accordingly extract the fact triplets. Then they filter those facts using a hybrid layer of SpaCy and Textacy to verify the extracted facts.

In Table 5, we present the results of each phase for this experiment. The *GT* column refers to the Ground Truth facts extracted using the SpaCy and Textacy libraries, the *LM* column refers to the facts predicted by the language model, the *Missed GT* column refers to the facts missed by the LM and captured in the ground truth. The *New LM* refers to the facts captured by the LM but not by the ground truth, and finally, the *Intersection* column is the facts captured by both the LM and the ground truth.

From observing the results of each phase, by manually evaluating samples of each, we noticed that the relevant facts are the intersection between the LM and the ground truth. However, the number of extracted facts was too little compared to the time and resources taken to extract them.

When experimenting with the SpaCy and Textacy layer to extract the facts to see if this filtering step could improve the quality of the facts extracted from OpenIE, we took a

POC	GT	LM	Missed GT	New LM	Intersection
count entities	11,666	6,778	9,593	4,077	4,161
count relations	6,901	5,174	4,907	2,682	3,085
count facts	14,885	12,251	11,330	7,470	5,878

Table 5. Extracting Facts Using Language Model

sample of 10,000 sentences from the question-answer processed pairs. The execution time was 1 hour, 30 minutes, and 27 seconds to generate 3048 facts that contain 3060 entities and 1840 relations. This filtering step could be an extra cleaning and verification step. Still, within our project, we didn't notice any significant improvement after applying this step relative to the time taken to process the whole dataset.

Authors

Mervat Abu-Elkheir Associate Professor and vice-dean at German University in Cairo. She has over 20 years of experience in academic teaching. Her current research interests are in the areas of interpretable machine learning and AI, AI for software engineering, data management, and engineering, and natural language understanding. She has been an IEEE member since 2014.

Mayar Osama received a Bachelor of Science in Media Engineering and Technology from German University in Cairo. Currently, she is pursuing her Master of Computer Science while working as a teaching assistant. Her research interests include Natural Language Processing, and Knowledge Reasoning and Representation

PROCEDURAL GENERATION IN 2D METROIDVANIA GAME WITH ANSWER SET PROGRAMMING

John Xu[1], John Morris[2]

[1]Harvard-Westlake High School, 3700 Coldwater Canyon Ave,
Studio City, CA 91604
[2]Computer Science Department, California State Polytechnic University,
Pomona, CA 91768

ABSTRACT

Video game designers often find themselves at a crossroad when designing levels; namely, many have a difficult time balancing the amount of control they want to have over what their levels look like [1]. If too little control is given, like in the case of pure perlin-noise generation, levels can end up with too much variation and unideal generations [2]. Softlock is an example of unideal generation in the case of metroidvania games, if the placement of keys cannot be easily controlled and end up being placed behind gates, the players can get permanently stuck [3]. Developers may usually hand-make all levels in order to try and prevent this from happening, however they risk spending too much effort and time on designing levels, resulting in a general lack of quantity in levels. Objectively speaking, both methods have their strengths and work well in specific genres of games, but limiting oneself to the boundaries of these methods does not fundamentally achieve both quantity and accuracy. This paper proposes an unique solution to this dilemma, providing automated generation of levels while also giving developers much more control over the overall output. Our method uses Answer Set Programming (ASP) to verify generation based on restrictions we place, guaranteeing the outcome to be what we want [4]. To demonstrate our method, we applied our solution to a 2D metroidvania game made in the Unity game engine and conducted quantitative tests to assess how well our method works as a level generator [5].

KEYWORDS

Procedural Generation, Answer Set Programming, Video Game Application

1. INTRODUCTION

Answer set programming excels at searching for multiple solutions (facts) that satisfy constraints (rules) that are desired. Its strengths are very unique and can work extremely efficiently in situations where different solutions that satisfy certain constraints are needed. In other words, ASP is able to declaratively search for its solutions, and thus represent its output as only the rules that are fed into it [6]. There is no need in ASP to individually dictate each solution when the solutions can be represented by the restrictions that imply what they can and cannot be. This lessens the need for manual implementation and control, which makes things a lot more labor intensive and time consuming. In addition, ASP is extremely powerful at solving computationally intensive and difficult search problems (NP-hard). Its efficiency and power lends itself well to real-world applications in topics that would otherwise cause huge headaches. For example, complicated problems like industry-level scheduling and planning can be handled efficiently with

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 123-136, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130511

ASP. Unfortunately, ASP has not been frequently applied to video game development, which is what this paper focuses on [7]. However, this paper will prove that ASP's powerful and efficient constrained search works well with the delicate generation of video game levels.

The programming language in which the rules are written is called AnsProlog (short for: Answer Set Programming in Logic) [8]. These constraints are solved using answer set solvers, which compute the final answer sets, such as clasp, assat, cmodels, and clingo. Our experimentation with ASP involves the use of clingo as an ASP solver. It was rare for previous endeavors to produce game-related projects with ASP in the first place, and it is extremely rare to see it done with clingo [9]. Many of those who tried ASP as a way of generating levels did not go far with the extent of application, often opting for a proof of concept and very small scaled designs that did not consider larger scale implementation of ASP into games. Many parts of the ASP application into video games remain unexplored and a lot of potential problems in ASP implementations like loading time and loading behavior are not as considered in the rare instances where ASP is used. However, our endeavor strives to create a practical and full-scale game, which would demand our implementation to be efficient in areas previously mentioned. Another point that expands on the lack of practicality is the fact that answer set solvers such as clingo need to go through complicated installation steps on any local machine in which it is used to function properly. Therefore, it becomes difficult for other implementations to work for the wider public. However in our implementation, we have the option for clingo to be hosted in a server so that it is able to send its answer set output to any local machine without the user needing to install anything extra. The online feature will not be the focus of this paper. It is also important to note that our approach to generate metroidvania-style levels is very unique to begin with [10]. Many of previous implementations had their focuses on other aspects, and ASP played varying sizes of roles in these projects

In this paper, we will be using clingo as our ASP solver to solve for rules we can manually input and adjust. The overall level is split into rooms, where each room and its relevant information is determined by the solver first represented by a graph, and then each room is iteratively solved based on the information from the graph. The solution is sent as a dictionary to the game engine, where we've set up scripts to read these dictionaries and build it into an actual level. Compared to a perlin-noise procedurally generated level, our method has strength in its ability to both generate and verify the map structure / geometry in the same step, which guarantees that the level that is generated satisfies any valid constraint we give it. ASP allows for easy prevention of softlock, which traditionally game designers solve by adding work-around solutions, like giving players the ability to selectively destroy or place certain tiles. With ASP, there is no need to create work-around solutions and lessens the burden on game designers to shape game play around the faults of level generation. Lastly, our method also has strength in its flexibility thanks to our incorporation of a powerful game engine. We are able to tweak generation rules and the tiles' sprites and artwork can also be easily modified and changed with a powerful game engine. This flexibility allows for even more potential variation in the levels that can be generated and thus supplying even more interesting combinations of content for players to explore.

We conducted two experiments to prove the function of our method. The first experiment involves measuring the runtime of generation of different types of rooms. All combinations of doors and directionality in rooms were tested, and average runtime was tabulated across a graph of time vs room type using the raw data of each type of room. The types are listed in an increasing order of number of doors, where bidirectional and directional door room types are clearly separated. This was done to see if our method was able to generate levels at good speeds relative to other more popular automated generation methods such as perlin noise generation, therefore determining our method's ability to be applied to more mainstream and real world examples. The second experiment involves testing the traversability of our generated levels to see

how effective they are at preventing softlocking and maintaining playability. A pathfinder was used to determine this across a 2D "letter representation" of the map where all important features of the map are represented by individual characters. The pathfinder would traverse through the level, finding the key first, and then going to the end. The pathfinder will not be able to reach the end if the key is after a gate, thus rendering the end unreachable since the end is always gated by a gate. We used this experiment to see how compatible ASP was with level generation and if it could "replicate" its strengths in a metroidvania level.

There are 5 more sections in this paper, each presenting different aspects of the experiment process and other relevant details. In the next section (Section 2), details are provided on challenges we had during the experiment and designing process. Meanwhile in Section 3, focus is put on the details of our solutions corresponding to these challenges mentioned prior. Section 4 focuses on the relevant details about the experiment we did, followed by related works in Section 5. Finally, Section 6 presents the conclusion remarks as well as pointing out the future of this project.

2. CHALLENGES

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

2.1. Directionality in Rooms

One of the important features developed using this ASP method is the implementation of directionality within rooms and determining a viable path in which the player can traverse the level. This specific feature is crucial for our method to generate traversable levels and prevent players from going where they are not supposed to. Though it may seem simple at first, its implementation presents itself as an extremely difficult one since the generation must take in account all the possible moves of the player and determine which way the room can lead players based on those moves. Our solution to this problem was to provide an effective way of determining direction, where we first determined all the possible tile locations the player could be, and then split the room into distinct platforms, which were defined in the code as floor tiles that are connected and within the player's jump height of each other. We then were able to determine directionality of the room based on these platforms but checking to see if the player's possible positions / movements were able to reach these platforms.

2.2. Long Solve Time

ASP is a power tool to be able to use, however we encountered downsides regarding the solve time that the solver is taking in order to generate valid solutions. Generally speaking, ASP solve times are prolonged when there is a big workload or if an excessive amount of constraint is placed. Knowing this, we were able to effectively counteract the factor we believed was giving us the biggest trouble, that being the big workload we were giving to the solver. The most obvious thing that added to the solver's workload was the fact that we were requiring the solver to solve for the entire level, so we came up with a "divide and conquer" solution where we split the generation workload into individual same-sized rooms to be solved iteratively. In this way, each call to the solver will have a room size that is a lot smaller to solve, exponentially decreasing the runtime of a singular room, and therefore decreasing the runtime of the entire level.

2.3. Learning ASP

The syntax and actual writing of AnsProlog was a challenge that was prominent in the early stages of development. Personally speaking, AnsProlog as well as its logic was very unique from other programming languages and difficult to understand at first, and thus it seemed like a far-fetched idea that any playable levels were able to be produced using AnsProlog. The way we set up the solver meant that we had to code the rules as a one whole long string, which were then passed into the solver, meaning that there was no color differentiation, auto-correct nor auto-indentation. This unfortunate circumstance made typos hard to detect, and in combination with our unfamiliarity resulted in a very difficult start to our research. However, we were able to familiarize ourselves with the workings of ASP by testing its behavior tiresomely with a Jupyter Notebook that simulated tile placements until we were able to grasp and utilize AnsProlog for our research.

3. SOLUTION

In this paper, we will be using Answer Set Programming to generate metroidvania-style levels. The levels will contain keys that can unlock their respective gates, where the ASP guarantees the keys to appear before the gate before the player's path. Our method views a single level as separate rooms that have a constant size, and we go through a two step process to ensure the specific room geometry satisfies our parameters and constraints as well as making sure the general pathway is connected throughout the level. The graph contains information on where the room may lead the player (directionality) and whether the room contains a key, a gate or no special objects (traversal rooms). Each room within the graph is iteratively solved based on these information from the graph. To ensure that a constant pathway still remains through the level, the edge geometry of neighboring built rooms were passed in to the solving process of the current room, however this still caused problems because it could be the case that no solution exists under the additional consideration of neighboring edge tiles. For the case of a no solution / unsatisfiable output, an algorithmic fix was applied to this problem involving the removal of neighboring rooms. Our level generator will output a dictionary of all tile, room, key, and gate information after all the rooms are iterated through and all unsatisfiable conflicts are resolved to the Unity game engine, where it is interpreted and built as an actual playable level. Based on the locations of the tiles, the artwork for tiles are determined accordingly to stylize the level.

The first step in our method is to define the parameters of the overall level, those being the size of the level, the type of gates we want it to have, the maximum and minimum number of gates and the size of each room. These parameters are passed in to the solver, which solves based on the ASP rules we've given to the solver as well, and returns to us a graph that contains information about the rooms. The graph shows the rooms' directionality and what they contain (key, gate, or nothing), and based on this graph we are able to iterate through each individual room to generate it. To make the rooms fit together, all the solved neighboring rooms' adjacent tiles to the original room (edge geometry) is mirrored over to the current room-to-build. The above mentioned fix is applied when an unsatisfiable case occurs for a room. The final solution after all rooms have been iterated are outputted as a dictionary to be read in Unity for the actual level to be built in-game.

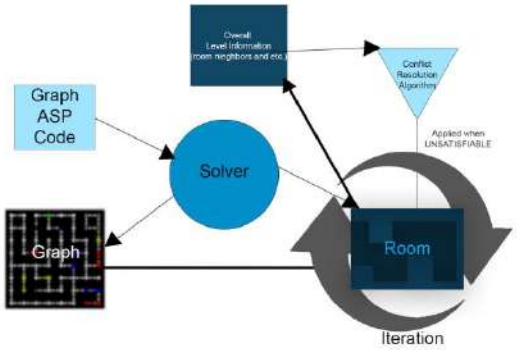


Figure 1. Overview of the solution

The first main step in our method is the generation of the graph that acts as a guideline to the more specific generation later on. The graph is a 2D representation of separate nodes representing rooms and connecting each room are either no arrows, a bidirectional arrow, or an one directional arrow representing the directionality of the room [11]. ASP guarantees solutions that always have a path between any two rooms. The nodes themselves can either be a circular shape, a diamond shape or a rectangular shape. White circular nodes mean that the room does not contain any keys or gates. Gray circular nodes are boss rooms while the green circular node is the start room, which is picked randomly from the first row. Any other colored circular nodes represent areas that are gated by a gate corresponding to its color. The colors of rectangular and diamond nodes represent the type of key/gate that the room contains, and the shape of the node itself determines whether it is a key or a gate. Rectangular nodes mean that there is a gate in that node / room, and the diamond nodes mean that there is a key in that room. The types of key/gates we want to have in our level, how big we want the entire level to be, how big we want each room to be, and the maximum and minimum number of gates we want there to be must all be defined in the code, which will be passed into the clingo solver to be solved. It's important to note that keys are always placed in an area that is before the gates so that there is no way the player can get stuck. It's solution is sent as a dictionary to C# scripts that store this information for further generation.

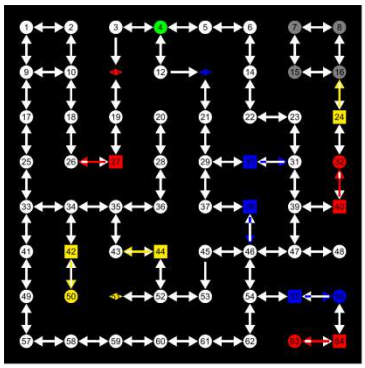


Figure 2. 2D representation of separate nodes

We utilized a mix of C# code and ASP to generate the graph. Different rules that are written in ASP are stored as strings in a separate script, and are combined here to form the complete ASP code. The parameters we pass in are slotted into their corresponding locations in the ASP code, and everything gets passed in to the solver to be solved.

```
public void BuildGraph(int worldWidth, int worldHeight, int gateKeyCount, int maxGatePerKey, int minGatePerKey, int bossGateKey,
int startRoom, int timeout, int cpus)
{
    string aspCode = WorldMap.bidirectional_rules + WorldMap.graph_rules + WorldMap.gate_key_rules;
    string path = ClingoUtil.CreateFile(aspCode);
    ClingoSolver solver = FindObjectOfType<ClingoSolver>();
    solver.maxDuration = timeout + 10;
    solver.Solve(path, $" -c max_width={worldWidth} -c max_height={worldHeight} -c start_room={startRoom} " +
        $"-c key_count={gateKeyCount} -c max_gate_type_count={maxGatePerKey} " +
        $"-c min_gate_type_count={minGatePerKey} -c boss_gate_type={bossGateKey} --parallel-mode {cpus} --time-limit={timeout}");
}
```

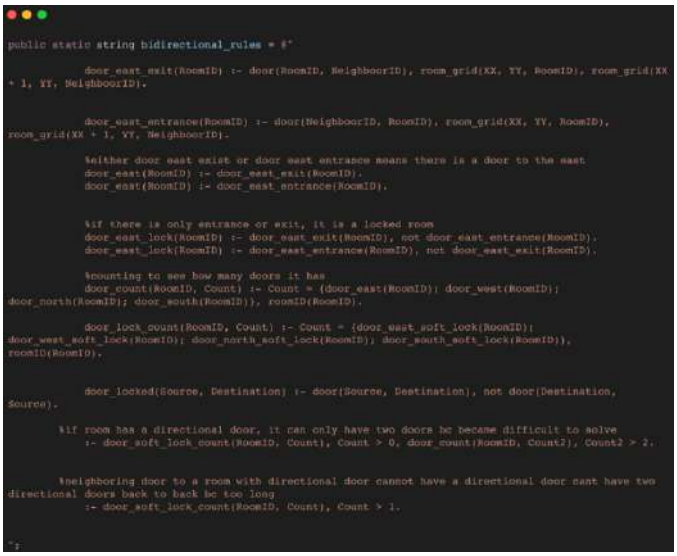
Figure 3. Screenshot of code 1

Below are two examples of ASP rules chunks that are added to the final graph asp code. The following code chunk defines the roomID based on where a room is in the level, and a grid of all the rooms is also defined based on the X position, Y position and a room ID. A door is defined as possibly occurring between any room and its immediate horizontal or vertical neighbors. The start room is also defined as any room in the top row of the level.

```
public static string graph_rules = @"
  #const max_width = 4.
  #const max_height = 3.
  #const start_room = 1.


  width(1..max_width).
  height(1..max_height).
  roomID(1..max_width*max_height).
  1{room_grid(XX,YY, ID)}1 :- width(XX), height(YY), ID = (YY - 1) * max_width + XX.
  room(ID) :- room_grid(XX,YY,ID).

  {door(RoomID1, RoomID2)}1 :- room_grid(XX,YY, RoomID1), room_grid(XX+1, YY, RoomID2).
  {door(RoomID1, RoomID2)}1 :- room_grid(XX,YY, RoomID1), room_grid(XX-1, YY, RoomID2).
  {door(RoomID1, RoomID2)}1 :- room_grid(XX,YY, RoomID1), room_grid(XX, YY+1, RoomID2).
  {door(RoomID1, RoomID2)}1 :- room_grid(XX,YY, RoomID1), room_grid(XX, YY-1, RoomID2).



  1{start(RoomID): room_grid(_,YY,RoomID), YY == 1}1.


";
```

Figure 4. Screenshot of code 2

This next chunk of ASP code is responsible for bidirectional constraints. The code is simplified to only include east doors for the sake of better viewing since all the other doors of other directions all behave similarly. An east door exit and an east door entrance is defined as must being east one room of the original room. A door to the east, whether entrance or exit, is considered an east door to the code. If there is only an entrance or only an exit, the room is considered locked to the east. We then count how many doors the room has as well the number of locks (this would originally contain all four directions). When you can only travel one way through a door, it is directional. When a room has a directional door, there can only be two doors. There can also not be two directional doors back to back in neighboring rooms. Anecdotally, we have observed higher runtimes when these restrictions are not in place, so these restrictions were added for maximum efficiency with solve time.



```
public static string bidirectional_rules = @"

        door_east_exit(RoomID) :- door(RoomID, NeighboorID), room_grid(XX, YY, RoomID), room_grid(XX
+ 1, YY, NeighboorID).


        door_east_entrance(RoomID) :- door(NeighboorID, RoomID), room_grid(XX, YY, RoomID),
room_grid(XX + 1, YY, NeighboorID).

        %either door east exist or door east entrance means there is a door to the east
        door_east(RoomID) :- door_east_exit(RoomID).
        door_east(RoomID) :- door_east_entrance(RoomID).


        %if there is only entrance or exit, it is a locked room
        door_east_lock(RoomID) :- door_east_exit(RoomID), not door_east_entrance(RoomID).
        door_east_lock(RoomID) :- door_east_entrance(RoomID), not door_east_exit(RoomID).

        %counting to see how many doors it has
        door_count(RoomID, Count) :- Count = {door_east(RoomID); door_west(RoomID);
door_north(RoomID); door_south(RoomID)}, roomID(RoomID).

        door_lock_count(RoomID, Count) :- Count = {door_east_soft_lock(RoomID);
door_west_soft_lock(RoomID); door_north_soft_lock(RoomID); door_south_soft_lock(RoomID)},
roomID(RoomID).


        door_locked(Source, Destination) :- door(Source, Destination), not door(Destination,
Source).

    %if room has a directional door, it can only have two doors bc became difficult to solve
        :- door_soft_lock_count(RoomID, Count), Count > 0, door_count(RoomID, Count2), Count2 > 2.


    %neighboring door to a room with directional door cannot have a directional door cant have two
directional doors back to back bc too long
        :- door_soft_lock_count(RoomID, Count), Count > 1.


";
```

Figure 5. Room Iteration and Iterative Unsatisfiable Conflict Resolution Algorithm

The graph output dictionary is used to generate each room iteratively based on an ID assigned to it based on its location in the level. The solver is used here again to determine the specific terrain geometry within the room as well as special cases such as implementing the gates and keys into actual tiles. As mentioned earlier an issue of an unsatisfiable case in rooms emerges as we try to fit these rooms back together. Our solution involves removing neighboring rooms in order to get rid of what's causing the unsatisfiable error. Our original approach was to randomly remove neighboring rooms, solve for the current room, then placing the removed room at the top of the stack of all rooms-to-build to be solved next. The random selection of neighboring rooms was not effective enough, because multiple neighboring rooms could be causing this unsatisfiable nature of the current room. Due to this lack of effectiveness, we modified our solution so that it generates all the possible combinations of neighboring tiles from smallest number of neighbors to biggest, and iterated through them to find the minimum set of neighboring rooms that caused the room to be unsolvable. By removing this minimum set, solving for the current room, and putting the removed rooms back to the stack to solve, the unsatisfiable case is able to be completely solved.

The following method gets a random smallest permutation out of the permutation 2D list generated by another method. The code represents its immediate neighbors as a 3 by 3 grid with numberings from 1 through 9, so the permutation returns the to-be-removed neighbors this way.

```
public static List<T> GetSmallestRandomPermutation<T>(List<List<T>> permutations, bool remove)
{

    int smallest = int.MaxValue;
    foreach (List<T> permutation in permutations)
    {
        if (smallest > permutation.Count) smallest = permutation.Count;
    }

    List<int> smallestIndices = new List<int>();
    for (int i = 0; i < permutations.Count; i += 1)
    {
        List<T> permutation = permutations[i];
        if (smallest == permutation.Count) smallestIndices.Add(i);
    }
    int rand = Random.Range(0, smallestIndices.Count);
    List<T> smallestPermutation = permutations[smallestIndices[rand]];
    if (remove) permutations.Remove(smallestPermutation);

    return smallestPermutation;
}
```

Figure 6. Screenshot of code 3

The next piece of code shows how the output of the previous method is used. The list of integer output is stored as the removed Neighbors of the current room, which is later interpreted into the actual room inside the world/level, which can then can be easily removed.

```
if(newRoom.buidStatus == ClingoSolver.Status.UNSATISFIABLE)
{
    List<int> killRooms = newRoom.removedNeighbors;
    foreach(int killID in killRooms)
    {
        Room killRoom = world.GetRoom(killID);
        DestroyRoom(killRoom,newRoom.lastBuildTime, roomID, ClingoSolver.Status.UNSATISFIABLE);
    }
}
```

Figure 7. Screenshot of code 4

This is a cool gif of the iteration algorithm in action.

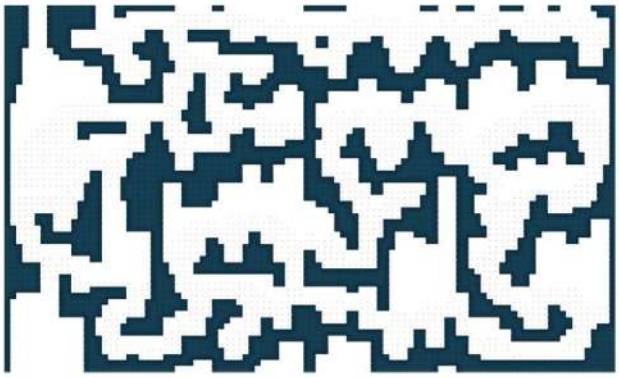


Figure 8. Screenshot of the gif

4. EXPERIMENT

4.1. Experiment 1

Our solution solves the problem because we were able to effectively use ASP to generate rooms in a relatively adequate amount of time. A level generator isn't really effective if it struggles with its runtime, and thus fails to bring a playable map to players within a short period of time. This experiment involves measuring the average runtime in seconds of solving 20 x 20 no gate and no neighbor rooms. The varying runtimes for all combinations of bidirectional and directional passageways are accounted for in this experiment. The fact that our ASP method was able to quickly generate its rooms in most passageway types shows that our method of applying ASP to level generation is viable to create 2D metroidvania maps.

Runtime is an effective way of quantifying map generation, which can otherwise be a very hard subject to measure and compare. Our sample size for this experiment is at least data from 30 generated levels, which could be improved. However, generating large amounts of levels still takes quite a long time, so the sample size is unfortunately limited.

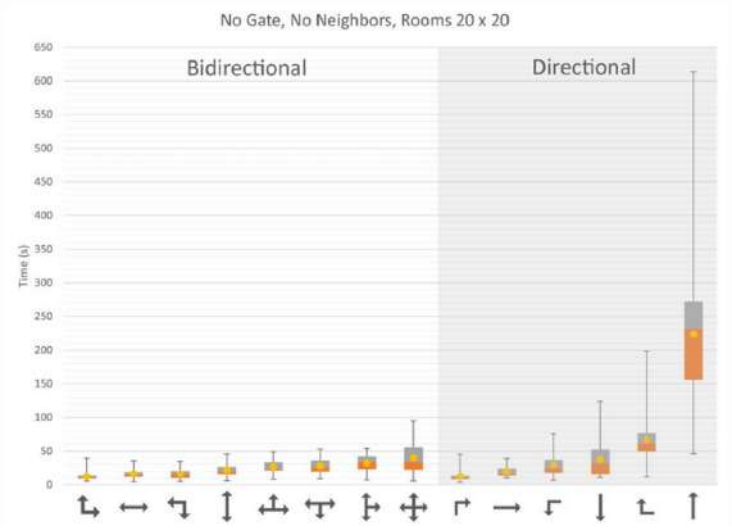


Figure 9. Graph of experiment 1

In a 20 by 20 no gate no neighbor scenario, all passageway types except directionally up rooms solve rapidly (within a minute or two at most extreme cases). It also appears that the amount of doors is directly related to the solvetime for rooms. The first 8 plots in the graph only have bidirectional doors and are ordered based on average runtime, with the first four having 2 bidirectional doors, followed by the 3 door rooms, and then the four door room. The rest of the plots in the graph in the gray section are rooms with one directional door and one bidirectional door. They are ordered from left to right in increasing average runtime. The directionally up room types solve on average 4 times slower than the second biggest average, and go as high as 600 seconds in some cases. Considering the fact that a 20 by 20 room gives plenty of room for players to explore, the generation speed is very solid. By limiting the directionally up rooms in the future, even faster runtime can be achieved.

4.2. Experiment 2

Our methodology is able to not only work efficiently but also work accurately to guarantee the outcomes we want it to achieve. One of the crucial goals for using ASP in level generation is to guarantee the player's path from point A to point B, whether that be to a key or to a boss area. We are able to prove that our solution completely guarantees the player to reach the key before a gate that leads to an endpoint by using a pathfinding algorithm not designed but modified by us.

We used around 100 maps to guarantee that the map is traversable everytime. More could be used for testing because the level was represented as a 2D map consisting of ascii characters that represented different objects and tiles in the level. The pathfinding algorithm was also able to work quickly through multiple files and what we were looking for was simple: a true or false for traversability.

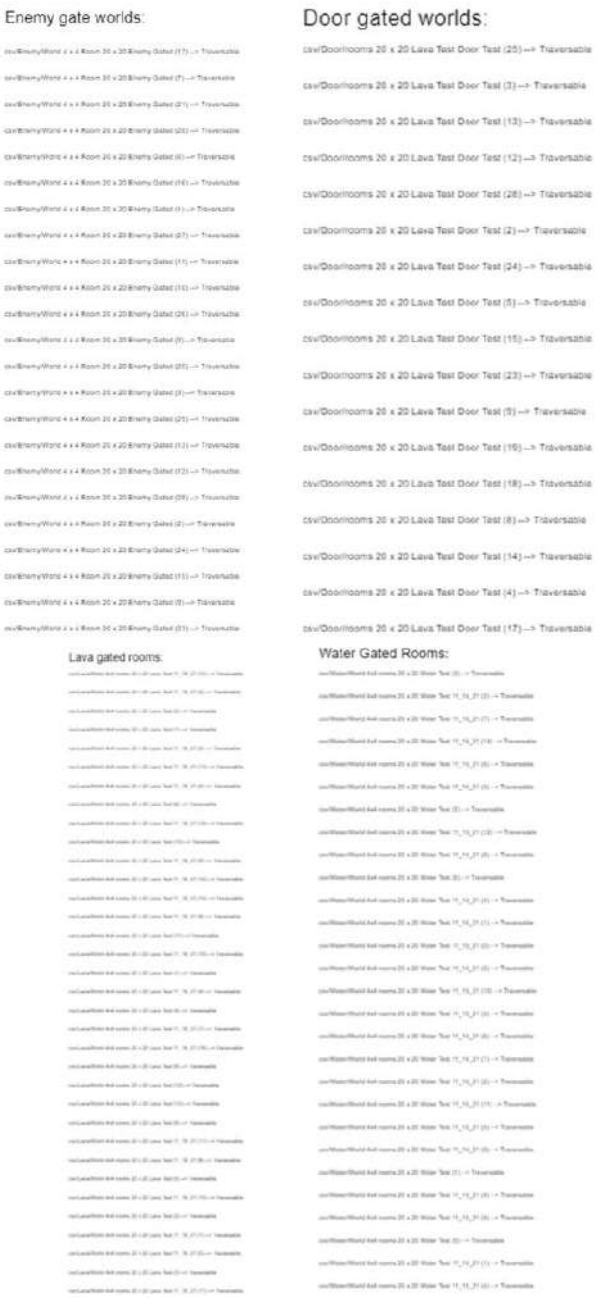


Figure 10. Result of experiment 2

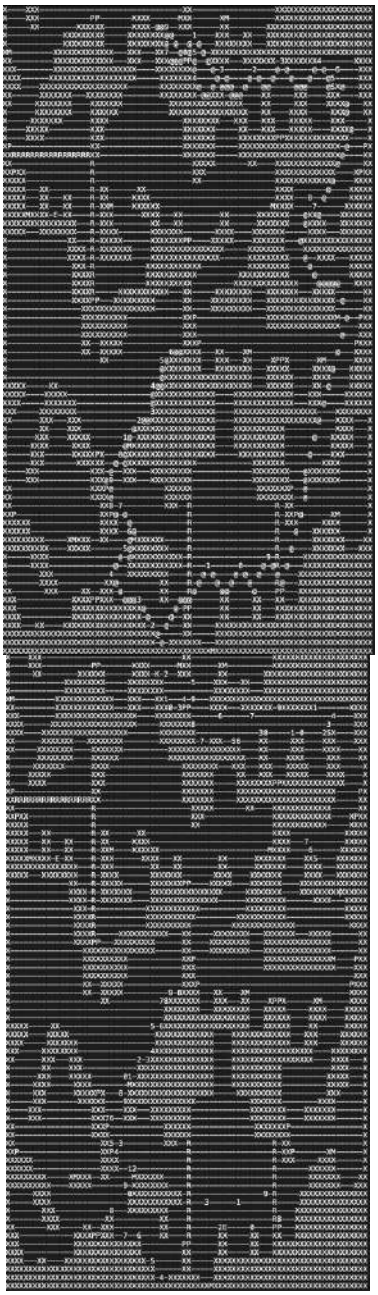


Figure 11. Example Ascii Maps for an enemy type of gate for better reference

The first list is very long, however it represents the guaranteed traversability of all the tested worlds. The pathfinding does its calculations and only outputs a “traversable” if the map that it traced was traversable, otherwise only the name of the map would appear. To visualize what the pathfinder is doing, a specific example is provided to show how the pathfinder starts at its start point (represented by ‘S’) traverses itself through the map to reach the end (‘B’). There are two maps that have either tiles (‘X’) or empty (‘-’) as its base structure, one representing a graph that has not been traversed yet, with all the key, start, and end visible. The other represents the pathfinder’s path (‘@’), and it is only able to go through the enemies (‘R’) when it has the key. The pathfinder always prioritizes getting the key first, so we see that the pathfinder

gets the key first before backtracking to get to the end in the shortest distance. Other ascii keys such as numbers as other letters are for previous testing that will not be covered in this paper.

From the first experiment, we can tell that the runtime for rooms does not usually exceed a one minute to two minute generation at most, meaning that generation for rooms will usually happen fast, thus levels in general will not take a long time to generate. There are limiting edge cases to the runtime performance, so the solution still needs to be better refined to produce more consistent results for overall level generation. In general, the room generation runtime is not bad for so many restrictions we put on it, however there is definitely room to grow. The second experiment can be considered a bigger success since the traversability of the levels are proven to be working for all test cases. This meets the demand of having a level generator where game designers can guarantee the playability of a level, or in more general terms, the ability to fully constrain certain parts of generation.

Considering how many restrictions we have put on the code for the solver to solve in addition with the iterative conflict algorithm, solve time was not expected to be too low. The actual solvetime of rooms did match our expectations, and proved to be an overall okay speed. We had high hopes of traversability on the other hand, since the guaranteeing aspect of ASP was one of the strong points we looked forward to utilizing. After testing, the traversability of ASP is indeed guaranteed in all cases, taking more time to solve for a more complete and accurate solution.

5. RELATED WORK

This paper focuses on using Computational Tree Logic to prevent softlocking in Super Metroids [12]. Path determination is also studied closely in this paper as the basis to solving softlocking as well. Overall there are many similarities between this paper's approach and our approach since we also view path determination as a crucial part of determining possible softlocks. This paper mostly differs in the aspect of what specific tool we chose to use to prevent softlocking, we used ASP while they used CTL.

This paper provides a very in depth "skeleton" of how to generate correct key-gate structures in video games using Answer Set Programming [13]. The paper does not specifically apply to metroidvania games, but games in general. Both our paper and this paper uses answer set programming as its tool of generating the placement of keys and gates since both want to guarantee keys before their corresponding gates. This paper differs from ours in the sense that we had more practical application to games since we used the output solution to go one step further and generate real 2D metroidvania levels. In other words, both papers' approach to generate the overall structure of the map is similar but differs in the actual implementation of the level.

This paper focuses on a very well-rounded approach to the procedural generation of metroidvanias [14]. Similar to our paper, this paper recognizes the difficulty of automating level creation, and strives to outline their process of creating an entire metroidvania, however it includes much more details than our paper in details such as player controls, animations and other visualizations while we tended to focus on the level generation aspect of the entire game. The paper also has a different approach to determining the key gate order, opting for a branching approach where node position represents its priority in the overall order of gates.

6. CONCLUSIONS

We have proposed a method to generate 2D metroidvania levels with ASP that prevents softlocks and guarantees traversability [15]. We generated an outline structure of a level first, then were

able to iteratively solve and render each room in the level. Each room and its neighbors were checked for their ability to connect together and any possible unsatisfiable errors were accounted for in the process using our iterative conflict resolution algorithm. We made sure that only the minimum permutation of rooms causing unsatisfiability was removed. To quantitatively test that our method works consistently and effectively, we applied our method to two experiments. The first being an experiment measuring the runtime of different types of room passageway generation. The second solution being an experiment measuring the traversability of each level by using a pathfinding algorithm. Runtime turned out to not be extremely fast, but in most cases short enough for level generation to take place in a brief time. Traversability was guaranteed in all cases in which levels were tested, showing that ASP can be very effective at verifying the playability of levels. Though it may take longer times to solve, ASP has complete insurance of the level working. This “trade-off” for verification over time is a good enough cost to address the issue of many issues plaguing traditional procedurally generated metroidvania levels.

One of the glaring problems with our method of applying ASP is the fact that actual users need to install clingo on their local computers before being able to actually generate maps with our method. This makes the idea of applying ASP to games a bit counter-intuitive. In addition, the runtime also seems to be a big issue in some cases, so more constraints have to be placed to avoid long solve time spikes for certain rooms that could prolong the generation of the entire level, leaving players to wait a long time to start playing a level.

We plan to solve the issue of installing clingo by hosting clingo in an online server, so any local computer may just request a solution from the server through local scripts setup in the game. Solve time can be reduced if we limit the amount of up directional passageway rooms that can exist in the map, which could speed up runtime significantly.

REFERENCES

[1] Dondlinger, Mary Jo. "Educational video game design: A review of the literature." Journal of applied educational technology 4.1 (2007): 21-31.

[2] Jakes, David, et al. "Perlin noise generation of physiologically realistic patterns of fibrosis." BioRxiv (2019): 668848.

[3] Guo, Yunfan, et al. "Soft-lock drawing of super-aligned carbon nanotube bundles for nanometre electrical contacts." Nature Nanotechnology 17.3 (2022): 278-284.

[4] Brewka, Gerhard, Thomas Eiter, and Mirosław Truszczyński. "Answer set programming at a glance." Communications of the ACM 54.12 (2011): 92-103.

[5] Oliveira, Bruno Pinheiro, et al. "A framework for metroidvania games." Proceedings of SBGames (2020): 836-844.

[6] Erdem, Esra, Michael Gelfond, and Nicola Leone. "Applications of answer set programming." AI Magazine 37.3 (2016): 53-68.

[7] McAllister, Graham, and Gareth R. White. "Video game development and user experience." Game user experience evaluation (2015): 11-35.

[8] Mellarkod, Veena S., Michael Gelfond, and Yuanlin Zhang. "Integrating answer set programming and constraint logic programming." Annals of Mathematics and Artificial Intelligence 53.1-4 (2008): 251-287.

[9] Gebser, Martin, et al. "Multi-shot ASP solving with clingo." Theory and Practice of Logic Programming 19.1 (2019): 27-82.

[10] Morris, John. DESIGNING A PROCEDURALLY GENERATED METROIDVANIA STYLE VIDEO GAME USING ANSWER SET PROGRAMMING. Diss. California State Polytechnic University, Pomona, 2021.

[11] Hou, Ji, et al. "Pri3d: Can 3d priors help 2d representation learning?." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[12] Mawhorter, Ross, and Adam Smith. "Softlock Detection for Super Metroid with Computation Tree Logic." Proceedings of the 16th International Conference on the Foundations of Digital Games. 2021.

[13] Smith, Thomas, Julian Padget, and Andrew Vidler. "Graph-based generation of action-adventure dungeon levels using answer set programming." Proceedings of the 13th International Conference on the Foundations of Digital Games. 2018.
[14] Stalnaker, Trevor Wayne. "Procedural generation of metroidvania style levels (thesis)." (2020).
[15] Claiche, Benjamin Elliot. Haiku, the Robot. Sistema de mapas para 2D Metroidvania. Diss. Universitat Politècnica de València, 2021.

AN ADVANTAGEOUS AND USER-FRIENDLY MOBILE PROGRAM TO BENEFIT STUDENTS IN SEEKING THEIR SUITABLE COLLEGES THROUGH THE USE OF WEB SCRAPING, MACHINE LEARNING, AND FRONTEND DESIGN

Sibo Tao[1], Yu Sun[2]

[1] Yorba Linda High School, 19900 Bastanchury Rd,
Yorba Linda, CA 92886
[2] Computer Science Department, California State Polytechnic University,
Pomona, CA 91768

ABSTRACT

The abstract describes a research project on the importance of college decision-making for high school students, and how college reviews and reflections can help students make informed decisions. The project focuses on developing an online platform that allows students to view reviews and reflections of former and current college students, covering not only academics, but also other aspects of college life such as culture, geographical location, campus, and tuition. The project includes two experiments, one on the accuracy of a sentiment analysis model and one on the user experience of the software frontend design. The results show that the sentiment analysis model is able to accurately predict positive and negative sentiment comments, but struggles with neutral comments due to data imbalance. The user experience experiment identifies several areas for improvement in the app's UI design. Overall, the research project seeks to provide high school students with a valuable resource for making informed college decisions.

KEYWORDS

Data, Analysis, Convenience, Reflect

1. INTRODUCTION

The growing pursuit of education among high school students has steadily grown in the past centuries [1]. Educational opportunities offered by a university have increased in value as they transfer to essential skills and productive work in society. Therefore, college decisions that students face during their senior year become increasingly important; students not only have to spend at least two years of their lives there but the style and quality of the university also directly influence students' future success. Choosing a college that suits the individual is crucial; individuals who attend a college that doesn't match their work style, culture, or expectations may find it difficult to endure their time there, leading to inefficient learning. Thoroughly understanding all aspects of a college before making a decision allows one to avoid obstacles and challenges that one will meet. Consequently, my research and project topic — College Reviews

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 137-147, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130512

and Reflections — is vital, as it will lead the student population down the correct route. Upcoming college students can view former and current students' opinions that describe various aspects of the colleges before making final decisions. In this way, students can fully evaluate a specific college and determine if it is the most favorable path based for them.

College websites and other services provide information to prospective students, particularly regarding academic abilities and credibility, allowing users to consider their college options based on academic qualities [2]. However, academics are not the only important characteristic, as other components of college make up a student's years of college life: culture, geographical position, campus, and tuition. It is necessary to look at student reviews to create a reputable and accurate reflection of a college, as they provide insight into its extracurricular qualities. Although the internet offers some resources for students' reviews and reflections, most resources lack sufficient data, which is the key for students to grasp what a college is like accurately. The opinions of a few students cannot accurately represent the entire population, creating a flawed representation of a college. A sufficiently-sized sample would better show where a college excels and fails in certain aspects. However, most resources online collect their data by interviewing students, requiring them to post a comment, and giving the rating of their college, which would not be efficient due to the volume of data and the time and resource costs.

The first experiment was designed to test the accuracy of the sentiment analysis model [3]. The experiment was set up so the model would predict fifteen comments of each type of sentiment, and its accuracies would be analyzed to verify whether it can precisely analyze each type of sentiment. We found out from the experiment results that the model perfectly predicted positive and negative comments but failed to predict neutral comments accurately. We analyzed that this stems from the imbalance of data for each type of sentiment – there are significantly more positive and negative comments than neutral ones. The second experiment was designed to test the user experience of the software frontend design: whether the app is user-friendly enough and whether experiment participants can provide beneficial feedback for the program. For this experimental setup, we asked ten participants to try the app and provide feedback regarding their experience. As a result, the users consistently highlighted three possible improvements: more information for each college, a search feature for the reviews, and more filters for the recommendation system on the main page. These comments signal that the UI design was not convenient enough for the users and later work is needed to remedy this problem.

2. CHALLENGES

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

2.1. The Difficulty of Data Collection

The difficulty of data collection was the first challenge I confronted as I was initially developing my app [4]. Finding useful data sources that satisfy my project's goal took work [5]. For my app, the comments I seek to gather are students' reflections and evaluations of the college they have gone to. Many of my early attempts to gather data failed due to irrelevant information. For instance, a flaw occurred as I attempted to gather data from social media, including Twitter and Reddit [6]. At first, I saw social media as an extensive database of valuable student opinions because millions of conversations are going on every hour on social media platforms. However, most comments there could have been more beneficial for my project. Students' comments are mostly unrelated to their college's positive and negative sides. Instead, these comments are filled with memes, gossip, and complaints, which are liked and commented on the most. Consequently, another route for my data collection – web scraping – met difficulties. The websites protect some

online data resources: their HTML code would be intentionally altered by the website owners or is set to be uninterpretable by web scraping methods [7][8]. Specific lines of web scraping code could circumvent this issue occasionally; however, an optimal solution for me is to look for alternative sites with available information.

2.2. My App's Operation on A Large Quantity of Data

Moreover, another potential obstacle I need to address is my app's operation on extensive data. There are thousands of colleges and universities in the United States and potentially millions of gatherable reviews on the internet. Storing a majority of them would undoubtedly meet issues: a sizable amount of data would slow the running speed of my app and would not be supported by much app-building software. Upon resolving this problem, the app developer must first support this quantity of data; others, such as Thunkable, would not function for my project [10]. An app developer well suited for storing extensive data is FlutterFlow, which also has optimal functions for filtering comments and colleges. In addition, I could implement ways to minimize the quantity of the data, including splitting the colleges based on the first letters of their names or their states.

2.3. The Artificial Intelligence Model

Consequently, the app's functionality is strongly impacted by the artificial intelligence model I am using. The accuracy of the model is crucial to the effectiveness of the app. As the model needs to analyze the sentiments of the comments – whether they are positive, negative, or neutral – it undoubtedly needs training on various data to ensure its ability of accurate predictions. The sentiment analysis model relies on the words – usually related to particular sentiments – that they were previously trained on to predict the sentimentality of the whole comment. Therefore, an adequate amount of manually-labeled data is required for training the model. After sufficient data has been collected, I will use a minimum of one thousand manually labeled comments to train the data. I would use the model to predict another one-thousand comments and verify its accuracy, securing its precision on further comments my app used and presented.

3. SOLUTION

The main structure of my program could be evaluated in three components: a web scraping backend, analytics system, and software developing frontend. All the data the app uses is stored within a database that would be processed by a sentiment analysis AI model [9]. For this program, I used Jupyter Notebook and multiple Python libraries: BeautifulSoup, Pandas, and Json for web scraping of data; Google Colab and other analysis-related Python libraries: sklearn and nltk to operate my analytic model; and Flutterflow for frontend user interactions with the program. With the failed attempts to develop my app using Thunkable, I eventually settled on FlutterFlow due to its convenience for designing components and extensive capacity to accumulate data. In addition, I decided to use Python to web scrape and program my sentiment analysis AI because numerous existing Python libraries provide a more straightforward and more advantageous way to accomplish these goals. My program design is based on convenient access to college students' reflections on their college. The program holds an immense amount of data, collected through web scraping and APIs, which is stored in a database that would eventually contain all the reviews, college names, tuition numbers, standardized test score ranges, and other characteristics of a college. All the reviews in the database would be labeled with their sentimentality utilizing a trained artificial intelligence analytic model, a key app feature. This feature would demonstrate the distribution of sentiments for all the comments to a college, allowing users to understand the ratio of students' sentimentality toward their college.

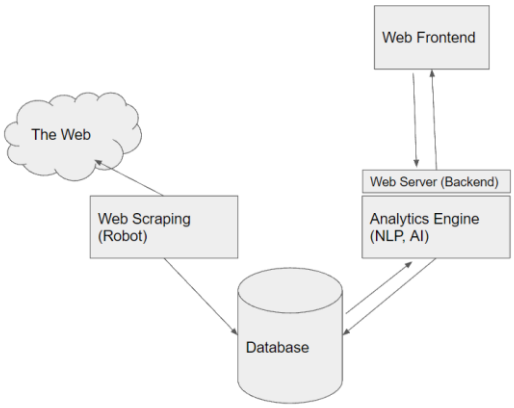


Figure 1. Overview of the solution

An essential part of my program is the web scraping backend. It relies on programming and APIs to gather an immense collection of students' reviews, their colleges, and other useful characteristics of colleges, including tuition and standardized testing score ranges. My program relies on this data to demonstrate the users' aspects of colleges through direct opinions of previous students, which is my primary purpose for designing this app.

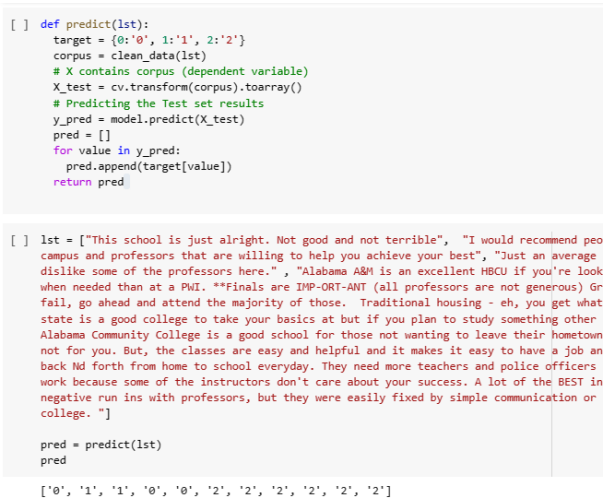
```
In [6]: print(len(specific_college_urls))
reviews = []
stars = []
college_names = []
sat_lower_end = []
sat_higher_end = []
act_lower_end = []
act_higher_end = []
acceptance_rates = []
tuitions = []
sat_ranges = []
act_ranges = []
for i in specific_college_urls:
    counter = 0
    page = requests.get(i)
    soup = BeautifulSoup(page.text, "html.parser")

    res = soup.findAll("div", class_= "overall-college-user-review-container review_div")

    for i in res:
        if counter < 30:
            var = i.findAll("i", class_= "fa fa-star")
            stars.append(len(var))

            if(i.find('p').text):
                reviews.append(i.find('p').text)
                #print(counter)
                counter += 1

            college_names_res = soup.find("div", class_= "college-header-banner-content")
            college_names.append(college_names_res.find("h1").text.strip())
            table = soup.find("table", class_= "table table-bordered")
            acceptance_rates.append(table.findAll("tr")[0].text)
            #print(acceptance_rates)
            sat_ranges.append(table.findAll("tr")[3].text)
            act_ranges.append(table.findAll("tr")[4].text)
            tuition_res = soup.find("div", class_ ="costs-container")
            print(college_names_res.find("h1").text.strip())
            tuitions.append(tuition_res.findAll('strong')[0].text)


print(college_names)
```

Figure 2. Screenshot of code 1

This screenshot illustrates the process of web scraping data from different colleges at one site, and this code runs in my program after storing each college's URL into an array. It collects all the necessary data from a specific college's page and stores them in different arrays, which would later be organized in separate columns in a CSV file. Multiple variables represent the fundamentality of my program: reviews array and college_name array — the two fundamental variables — and others like tuitions and acceptance_rates would be helpful for the recommendation system — which, based on the users' input of their characteristics, would provide them with recommended colleges in the main page — that would be implemented later in my program. Overviewing the process, this code first stores the URLs of all colleges and then webscrapes these pages by emphasizing the section that holds the vital components, including student reviews, SAT/ACT ranges, acceptance rate, and tuition [11]. Consequently, the array variables that were initialized would be filled with information from the different colleges, and later these arrays would be organized in a CSV file that would be uploaded to firebase.

Moreover, another fundamental part of my app is the analytics system. This system is requisite for the program's performance because it uses machine learning to perform the analysis action to

predict the sentiment of each comment. This analytic model is based upon Python-imported libraries like Natural Language Toolkit and Scikit Learn, which compute the human language into data and recognize human sentiments from the provided data [15].



```
[ ] def predict(lst):
      target = {0:'0', 1:'1', 2:'2'}
      corpus = clean_data(lst)
      # X contains corpus (dependent variable)
      X_test = cv.transform(corpus).toarray()
      # Predicting the Test set results
      y_pred = model.predict(X_test)
      pred = []
      for value in y_pred:
        pred.append(target[value])
      return pred
```

```
[ ] lst = ["This school is just alright. Not good and not terrible",  "I would recommend peo
    campus and professors that are willing to help you achieve your best", "Just an average
    dislike some of the professors here." , "Alabama A&M is an excellent HBCU if you're look
    when needed than at a PWI. **Finals are IMP-ORT-ANT (all professors are not generous) Gr
    fail, go ahead and attend the majority of those.  Traditional housing - eh, you get what
    state is a good college to take your basics at but if you plan to study something other
    Alabama Community College is a good school for those not wanting to leave their hometown
    not for you. But, the classes are easy and helpful and it makes it easy to have a job an
    back Nd forth from home to school everyday. They need more teachers and police officers
    work because some of the instructors don't care about your success. A lot of the BEST in
    negative run ins with professors, but they were easily fixed by simple communication or
    college. "]

    pred = predict(lst)
    pred

    ['0', '1', '1', '0', '0', '2', '2', '2', '2', '2', '2']
```

Figure 3. Screenshot of code 2

The image above contains the sentiment analysis model's prediction method and demonstrates its use. This code runs at the end of the analytics model portion – after the data has been resampled due to the imbalance between each type of comment and after the data has been split into training data and testing data for the model to be trained. The "predict" function initially cleans the data of the parameter "lst" by removing punctuations and lowering cases, and it sets the cleaned data to the corpus variable. Afterward, the model is applied to the data and predicts the sentiment of the data based on the words contained in each phrase. The sentiment of each data is then added into the "pred" array in the order of the input "lst," and the function returns the predicted-sentiment array. The cell below the predict demonstrates how the sentiment analysis process is performed through code: an array of sentences is provided and inputted as a parameter of the "predict" function. The predict function then would return an array of the predicted sentiments of each comment.

Consequently, the front-end would serve as the program's last major component. This component consists of designing a user interface that directly influences users' experience, and different features are implemented for users' convenience. The frontend component was initially implemented using Thunkable — which was not optimal for my program — before it was switched to FlutterFlow. As the program is designed for student usage, the software aspect of the project would be crucial since we want to offer a user-friendly and attractive application for them to use.

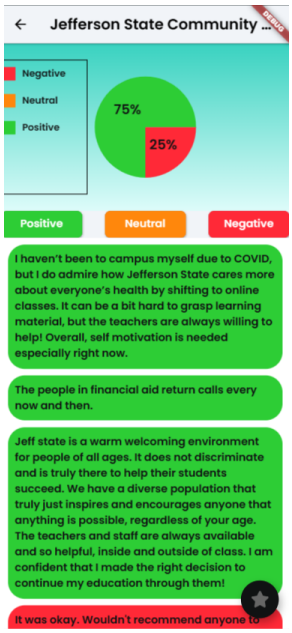


Figure 4. The reviews page of a specific college

The screenshot shows a specific college's reviews page, demonstrating various components designed for users' convenience. The pie chart at the top informs the reader of the proportion of positive, negative, and neutral reviews for the college. The positive, negative, and neutral buttons – denoted by a specific color – filter the comments by sentiment. Specifically, highlighting a type of sentiment means that only comments of that sentiment, whether positive, negative, or neutral, can be viewed. For instance, if only the green "Positive" button is highlighted, then only positive comments are filtered for users to perceive. Each comment has a background color based on its sentiments, which helps users quickly identify it. Furthermore, the star button on the bottom right corner stores a college to the "favorites page," where students can go through a shorter list of colleges more reflective of their interests. Consequently, it illustrates how the program is developed for users' convenience.

4. EXPERIMENT

4.1. Model Accuracy Experiment

An essential experiment I would conduct is testing the accuracy of the sentiment analysis model, which is vital to my program because the model's accuracy determines whether my app achieves its goal of illustrating a correct representation of the student population's sentiment toward a specific college.

We must test comments with diverse diction for the experiment to reflect whether the model is accurate. With this in mind, we came up with fifteen positive, negative, and neutral comments that would be used to test the model's accuracy. This ensures that all three possible outcomes – positive, negative, and neutral – are tested to see if the model can precisely predict them. After computing the comments into the model, we would compare the predicted sentiment with manually-labeled sentiment to examine the reliability of the analysis through its accuracy in predicting the correct sentiment.

Comment	Actual Sentiment	Predicted Sentiment
1	Neutral	Positive
2	Neutral	Neutral
3	Neutral	Negative
4	Neutral	Neutral
5	Neutral	Positive
6	Neutral	Positive
7	Neutral	Neutral
8	Neutral	Neutral
9	Neutral	Neutral
10	Neutral	Negative
11	Neutral	Neutral
12	Neutral	Positive
13	Neutral	Positive
14	Neutral	Neutral
15	Neutral	Neutral

Comment	Actual Sentiment	Predicted Sentiment
1	Positive	Positive
2	Positive	Positive
3	Positive	Positive
4	Positive	Positive
5	Positive	Positive
6	Positive	Positive
7	Positive	Positive
8	Positive	Positive
9	Positive	Positive
10	Positive	Positive
11	Positive	Positive
12	Positive	Positive
13	Positive	Positive
14	Positive	Positive
15	Positive	Positive

Comment	Actual Sentiment	Predicted Sentiment
1	Negative	Negative
2	Negative	Negative
3	Negative	Negative
4	Negative	Negative
5	Negative	Negative
6	Negative	Negative
7	Negative	Negative
8	Negative	Negative
9	Negative	Negative
10	Negative	Negative
11	Negative	Negative
12	Negative	Negative
13	Negative	Negative
14	Negative	Negative
15	Negative	Negative

Figure 5. Table of experiment 1

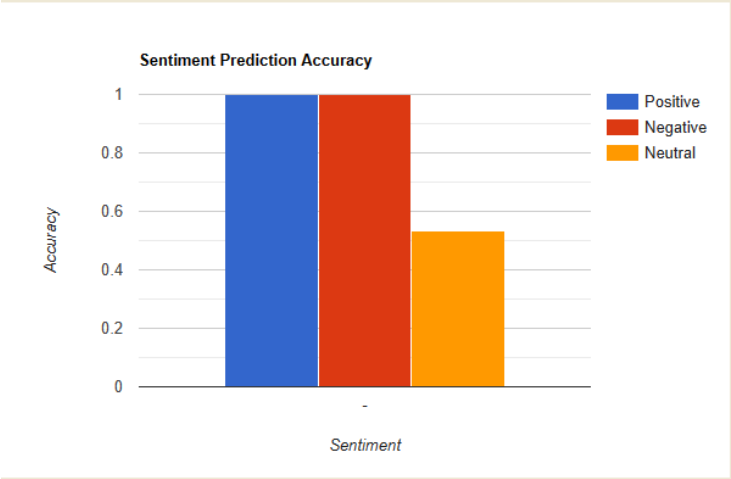


Figure 6. Sentiment Prediction Accuracy

From this experiment, the overall prediction accuracy is 84%, with a positive prediction accuracy of 100%, negative prediction accuracy of 100%, and neutral prediction accuracy of 53%. This statistic demonstrates that my model can accurately predict positive and negative comments but is ineffective for neutral ones. The model's accuracy in predicting neutral comments did not meet my expectations, as its accuracy drastically differs between positive and negative comments. Reflecting on this data, I believe it turned out this way because there was a lack of neutral sentiment comments from the training data and the internet compared to the positive and negative ones. This issue leads to the model's incapability to accurately predict neutral comments; the model is more likely to predict a comment as positive or negative than neutral. Therefore, this explains how the model can accurately predict positive and negative sentiment comments while predicting neutral comments inaccurately.

4.2. User Experience Experiment

Another potential blind spot that needs to be tested and fixed to improve the program is user experience. Although several components are integrated for the app to be more convenient to use — for example, recently searched colleges, favorite college page, and college recommendation system on the main page — different users could have different suggestions on which the app could be improved for convenience. This aspect is crucial because it shows the professionality of the program and attracts the users.

The experiment would allow users to test the app and offer feedback on potential improvements. Ten users would try out the app and navigate through all its features, and they would reflect and fill out a feedback form on ideas for new components that could make the program more convenient to use. Consequently, the feedback would be analyzed, and the program would be adjusted accordingly based on this feedback. In this way, we would acquire the indicative reflection from those who had the user experience on the program; therefore, this potential blind spot could be detected and addressed optimally.

User	Feedback (paraphrase)
User 1	For users' convenience, the program could implement a searching feature that could search and highlight words from specific comments, and the users could skip through comments and go to ones with these specific words.
User 2	For the filtering feature on the main page, the app should have more components that could be filtered; besides SAT score and tuition, other ones could be GPA, acceptance rate, state of the college/university, private schools only, etc.
User 3	On the college page, it could show more components of a specific college, such as the ranking of the college, its exceeding majors, and students enrolled each year.
User 4	The number of reviews for each college could be more for users to reflect on. Besides that, a searching tool for keywords in the comments would be beneficial.
User 5	I noticed that the filtering feature for the recommendation system on the main page only has "SAT score and tuition" on there. For the app to be more user-friendly, I believe adding more components to the filtering aspect would be great.
User 6	My suggestion for this program is that more features could be added; right now the program is limited to reviews of colleges, and I believe it would be better for the users to see more information about each college.
User 7	Going through the app, a feature that I thought would be helpful is another filtering feature in the reviews section for each college. I noticed that the app has a filtering feature based on sentiment, which is very helpful, but it could also filter something else such as food-related comments, campus-related comments, or housing-related comments.
User 8	Something that could enhance my experience of using this app is more reviews for each college. In this way users can understand more of each aspect of a college.
User 9	Everything seems fine with this app; I believe it could be especially useful for high school students who are about to make the college-decision. One thing that could improve the app is more filters for the recommending colleges component on the main page.
User 10	In order for the app to be more user-friendly, more useful descriptions of each college should be provided, such as its 75 percentile SAT score, its average GPA of accepted students, and its best majors.

Figure 10. Table of experiment 2

After analyzing users' feedback, three significant improvements became clear: including more reviews and information for each college, adding a searching or filtering feature on the college reviews, and making more available filters for the recommendation system. These reflections are beneficial as they demonstrate areas of the project that require enhancement. In the first version of this app, we expected there to be insufficient aspects. Indeed, the next step of this project is to resolve the issues above and refine the system to be additionally advantageous for users. More information would be gathered from the internet – including reviews and other vital details of a college. A search feature would be implemented to assist users in finding what meets their intentions. Lastly, the recommendation system would include more practical filters such as GPA, acceptance rate, and location.

5. RELATED WORK

The College Finder & Recommender Web Application System research paper provided a solution involving a web-based application that allows users to efficiently find universities that they can apply to and match users' demands [14]. Improving upon the existing solution, which is searching each independent college manually, this paper provides an effective solution of utilizing a filtering system, recommendation system, and a wish list. The filtering system helps users find colleges that match the required properties. The recommendation system helps users by recommending colleges that match the input of users' profiles. The wish list provided easier and quicker access to a shorter list of desired colleges. Overall, the system is effective and influential as it provides increased efficiency in searching for suitable colleges; however, there are crucial aspects that the system ignores. For instance, the app focuses on academic suitability for a student and ignores most of the other aspects of a college. My app improves on what they tried as it provides students' reviews on colleges that reflect other perspectives of colleges. With this improvement, the app's users could better visualize their lives in the college they are looking for and would further assist high school students in making the college decision.

6. CONCLUSIONS

The College Finder & Recommender Web Application System research paper provided a solution involving a web-based application that allows users to efficiently find universities to apply to and match users' demands [14]. Improving upon the existing solution, which is searching each independent college manually, this paper provides an effective solution of utilizing a filtering system, recommendation system, and a wish list. The filtering system helps users find colleges that match the required properties. The recommendation system helps users by recommending colleges that match the input of users' profiles. The wish list provided easier and quicker access to a shorter list of desired colleges. Overall, the system is effective and influential as it provides increased efficiency in searching for suitable colleges; however, there are crucial aspects that the system ignores. For instance, the app focuses on academic suitability for a student and ignores most of the other aspects of a college. My app improves on what they tried as it provides students' reviews on colleges that reflect other perspectives of colleges. With this improvement, the app's users could better visualize the college life they are looking for and would further assist high school students in making college decisions.

Building on the intention of informing students about colleges alongside their academic capabilities, various features, and ideas are implemented to help students understand the different perspectives of colleges. As technology advances and researchers make innovations, students later on will more easily find the colleges that suit them and will not regret the years of life they spent there.

REFERENCES

[1] Brint, Steven, and Charles T. Clotfelter. "US higher education effectiveness." RSF: The Russell Sage Foundation Journal of the Social Sciences 2.1 (2016): 2-37.

[2] Samuelowicz, Katherine, and John D. Bain. "Revisiting academics' beliefs about teaching and learning." Higher education 41 (2001): 299-325.

[3] Ullah, Mohammad Aman, et al. "An algorithm and method for sentiment analysis using the text and emoticon." ICT Express 6.4 (2020): 357-360.

[4] 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.

[5] Ohno-Machado, Lucila, et al. "Finding useful data across multiple biomedical data repositories using DataMed." Nature genetics 49.6 (2017): 816-819.
[6] Anderson, Katie Elson. "Ask me anything: what is Reddit?." Library Hi Tech News 32.5 (2015): 8-11.
[7] Samant, Nishank, et al. "College Finder & Recommender Web Application System." (2017).techniques." 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT). IEEE, 2019.
[8] Khder, Moaiad Ahmad. "Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application." International Journal of Advances in Soft Computing & Its Applications 13.3 (2021).
[9] Birnbaum, Lawrence, Margot Flowers, and Rod McGuire. "Towards an AI model of argumentation." Proceedings of the First AAAI Conference on Artificial Intelligence. 1980.
[10] Lim, Soo Ling, and Peter J. Bentley. "How to be a successful app developer: Lessons from the simulation of an app ecosystem." Acm Sigevolution 6.1 (2012): 2-15.
[11] Coyle, Thomas R., and David R. Pillow. "SAT and ACT predict college GPA after removing g." Intelligence 36.6 (2008): 719-729.
[12] Emeakaroha, Vincent C., et al. "A cloud-based iot data gathering and processing platform." 2015 3rd International Conference on Future Internet of Things and Cloud. IEEE, 2015.
[13] Kaplan, Andreas M., and Michael Haenlein. "Users of the world, unite! The challenges and opportunities of Social Media." Business horizons 53.1 (2010): 59-68.
[14] Samant, Nishank, et al. "College Finder & Recommender Web Application System." (2017).
[15] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." The Journal of machine Learning research 12 (2011): 2825-2830.

ROBUST OUTDOOR POSITIONING VIA RAY TRACING

Vladislav Ryzhov

Department of Multimedia Technologies and Telecommunications, Moscow Institute of Physics and Technology, Moscow, Russia

ABSTRACT

Positioning problem can be solved with several approaches in 5G. This paper introduces RT (Ray Tracing) technique for small-cell Outdoor positioning and tracking for systems with distributed architecture with multipath processing gain. Proposed approach exploits high-resolution angular spectrum estimation and known radio propagation environment. It solves positioning problem robustly even in NLOS cases via CoMP (Cooperative Multi Point) reception and joint processing at cloud CPU. The aim of this paper is to disclose proposed algorithm for NR and compare it to GNSS measurements.

KEYWORDS

Ray Tracing, positioning, angular domain, CoMP, channel modelling, GPS measurement

1. INTRODUCTION

Positioning is an old task for wireless communications. The roughest and low-complex methods include identification by Cell-ID and RSS (Received Signal Strength). With high synchronization requirements [1] GPS and TDoA (Time Difference of Arrival) measurements increase positioning accuracy up to dozens of meters. Nowadays, base stations are equipped with large antenna arrays (e.g., 32T32R) and current trend is to split panel into several ones or use single panel with increased number of elements. Under these assumptions angular domain represents the most accurate method for positioning [2]. At the same time the most of classic algorithms suffer from multipath propagation, LOS blockage and variable channel conditions. Another future concept is cooperative MIMO, which is expected to improve spectral efficiency and coverage mostly by increasing number of degrees of freedom and channel diversity [3].

Proposed RT (Ray Tracing) positioning approach exploits multipoint reception and angular spectrum estimation. It is robust to complex propagation conditions and partly benefits from large scattering and multipoint reception. Due to joint RT processing of distributed AoA measurements proposed method is the most suitable for C-RAN (Cloud Radio Access Network) architecture. Also, such approach can be used for real-time CSI (Channel State Information)/ precoder prediction or provide more information for dynamic scheduling for operators.

RT requires high precision maps for accurate localization. In this paper we omit issues related to accumulation data for accurate map reconstruction. With a multiple of automotive sensors, satellite images, cameras, etc. this map can be found. Although our approach does not require additional distributed synchronization, it assumes only intra-panel synchronization and calibration for angular spectrum estimation. Problems related to imperfect calibration,

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 149-164, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130513

interference directions and inaccurate map are solved via probabilistic weighting and clustering. Anyway, the map inaccuracy is also covered in this paper.

The contribution of this paper is that it proposes a novel approach for precise and robust positioning with a knowledge of propagation conditions and use of distributed RAN architecture. It covers the main aspects of developed framework and, among other things, compares simulations accuracy it with real GNSS measurements.

2. SYSTEM MODEL

2.1. Channel Model

Uplink channel is modelled via RT in certain outdoor environment, obtained from parsed 3D OSM (Open Street Map) with typically located UEs (User Equipment) and RRUs (Remote Radio Unit) from open-source maps with mobile operator towers and their frequency bands.

2.2. Layout

Cooperating RRUs (with 120^0 sector antenna) are placed to satisfy ISD ≈ 150 m, tilt ≈ 10^0, height ≈ 15-25 m. UEs are evenly spaced in the street at height of 1.5 m. For simplicity each UE is equipped with one antenna with uniform pattern, while RRU has (4x8) horizontal antenna array with 3GPP antenna element radiation pattern [4].

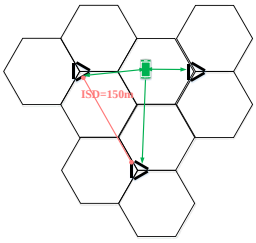


Figure 1. Layout with 3-sector antenna.

2.3. Signals

Each UE is configured [5] to send SRS (Sounding Reference Signal) in TDD mode with granularity K_TC = 2 and N_RB = 272 PRBs.

2.4. Overall Positioning Model

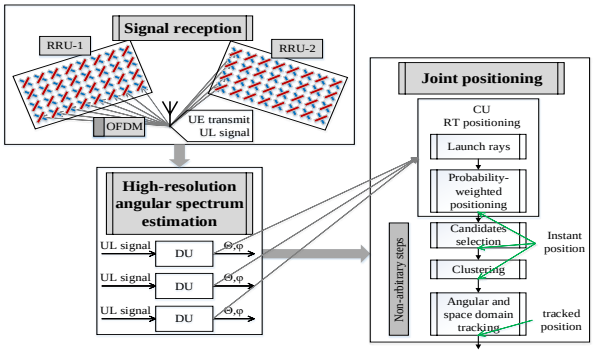


Figure 2. System model for positioning.

Multiple distributed RRUs cooperatively receive SRS, passed through RT channel. After that each DU (Distributed Unit) independently finds AoAs (Angles of Arrival) from estimated angular spectrum. These estimates are reported to CU (Central Unit), which contains precise outdoor model. CU localize UE from current instantaneous measurement via RT search in a direction of received angles. At this step position represents probabilistic estimate. Discarding and clustering of estimated positions is done to cope with outliers from imperfect angle estimation and map errors. Also, CU track measurements in angular domain and expectation of positions for each cluster in space domain in order to additionally discard false positions and smooth UE trajectory. To keep trade-off between required accuracy and complexity we are able to exchange additional clustering or tracking complexity for higher accuracy.

3. RT CHANNEL MODEL

RT channel modelling has a good accuracy [7] and papers [8-10] describes RT positioning. Developed C++ RT simulator has two main operating options: channel modelling and positioning. In both cases it contains the following propagation patterns:

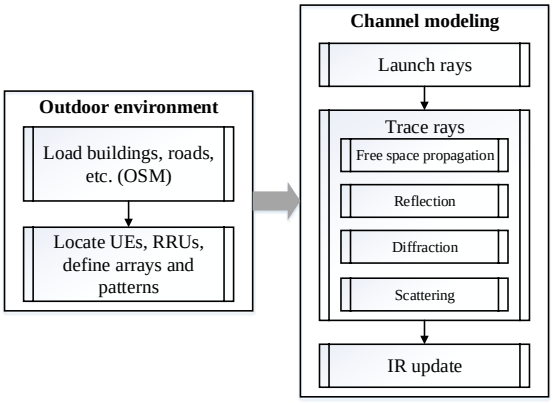


Figure 3. Concept of developed channel modelling via RT.

- Free space propagation

Solution of Helmholtz wave equation for point source in far field is a spherical wave, which defines phase and amplitude of propagating ray:

$$\mathbf{E}(\mathbf{r},\boldsymbol{\theta},\boldsymbol{\varphi}) = \frac{\mathbf{E_0}(\boldsymbol{\theta},\boldsymbol{\varphi})\mathbf{e}^{-\mathbf{ikr}}}{\mathbf{r}} \quad (1)$$

- Reflection

When interfered with by building, ground, etc. ray is reflected according to geometrical optics and reflection coefficients. From a Fresnel's equation reflection coefficients can be found from angles of incidence and reflection and electrical properties of material:

$$\mathbf{R}_{\perp} = \left|\frac{n_2\cos(\theta_i) - n_1\cos(\theta_t)}{n_2\cos(\theta_i) + n_1\cos(\theta_t)}\right|^2$$
$$\mathbf{R}_{||} = \left|\frac{n_2\cos(\theta_t) - n_1\cos(\theta_i)}{n_2\cos(\theta_t) + n_1\cos(\theta_i)}\right|^2 \quad (2)$$

- Scattering

Due to the roughness of surface, ray is scattered around the direction of reflection. Authors of [6] performed measurement campaign for scattering pattern of typical buildings and concluded that single-lobe directive model was the closest:

$$|\overline{E_S}|^2 = E_{S0}^2 \left(\frac{1+\cos\psi_R}{2}\right)^{\alpha_R}, \alpha_R = 4 \quad (3)$$

- Diffraction

Paper [11] defines the most common Knife Edge Diffraction (KED) model:

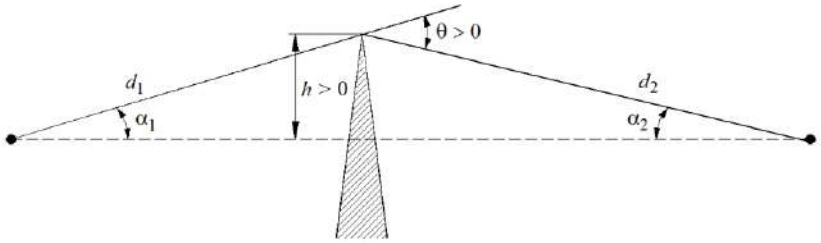


Figure 4. For definition v [11].

$$v = h\sqrt{\frac{2}{\lambda}\left(\frac{1}{d_1}+\frac{1}{d_2}\right)} \quad (4)$$

where electric field can be found from:

$$F(v) = \frac{E_{diffracted}}{E_0} = \frac{1+i}{2}\int_v^{\infty} e^{-\frac{i\pi t^2}{2}} dt \quad (5)$$

and approximation ($v > -0.7$) for diffraction loss in dB:

$$J(v) = 6.9 + 20\log\left(\sqrt{(v-0.1)^2+1}+v-0.1\right) \quad (6)$$

For detailed maps prediction accuracy of RT channel modelling is high [12]. The main hardship is complexity, which can be reduced with minimum storage and computations techniques. For these reasons the following methods were used for acceleration of tracing in C++ simulator:

- BVH (Bounding Volume Hierarchy) is a method [13] used for compact storage and reduced intersection testing. In order to decrease number of objects to compare, each object and UE are mapped to a spatial grid, so that ray is interfered only with objects in current grid cell.
- Parallelepiped wrappers are utilized to perform fast initial intersection test and discard objects without complex surface processing.
- Fast implementation of 3D geometry for ray tracing [14-15] with an extensive application of fast ray-triangle intersection [16].

4. POSITIONING

4.1. Independent AoA estimation

Firstly, each RRU estimate independently high-resolution angular spectrum from received SRS, e.g., via MUSIC algorithm [17].

$$\mathbf{S}(\boldsymbol{\varphi},\boldsymbol{\theta}) = \frac{\mathbf{1}}{\mathbf{a}^{\mathbf{H}}(\boldsymbol{\varphi},\boldsymbol{\theta})\mathbf{V}\mathbf{V}^{\mathbf{H}}\mathbf{a}(\boldsymbol{\varphi},\boldsymbol{\theta})} \quad (7)$$

$$\mathbf{a}(\boldsymbol{\varphi},\boldsymbol{\theta}) = \boldsymbol{\Delta}_{\mathbf{x}}^{\mathbf{i}}\cos(\boldsymbol{\theta})\sin(\boldsymbol{\varphi}) + \boldsymbol{\Delta}_{\mathbf{y}}^{\mathbf{i}}\cos(\boldsymbol{\theta})\cos(\boldsymbol{\varphi}) + \boldsymbol{\Delta}_{\mathbf{z}}^{\mathbf{i}}\sin(\boldsymbol{\theta}) \quad (8)$$

$$\begin{pmatrix}\boldsymbol{\Delta}_{\mathbf{x}}^{\mathbf{i}}\\ \boldsymbol{\Delta}_{\mathbf{y}}^{\mathbf{i}}\\ \boldsymbol{\Delta}_{\mathbf{z}}^{\mathbf{i}}\end{pmatrix} = \begin{pmatrix}(\mathbf{i}_{\mathbf{h}}-\mathbf{1})\mathbf{d}_{\mathbf{h}}\cos(\boldsymbol{\varphi}_{\mathbf{h}}) + (\mathbf{i}_{\mathbf{v}}-\mathbf{1})\mathbf{d}_{\mathbf{v}}\sin(\boldsymbol{\varphi}_{\mathbf{h}})\sin(\boldsymbol{\theta}_{\mathbf{v}})\\ (\mathbf{i}_{\mathbf{h}}-\mathbf{1})\mathbf{d}_{\mathbf{h}}\sin(\boldsymbol{\varphi}_{\mathbf{h}}) - (\mathbf{i}_{\mathbf{v}}-\mathbf{1})\mathbf{d}_{\mathbf{v}}\cos(\boldsymbol{\varphi}_{\mathbf{h}})\sin(\boldsymbol{\theta}_{\mathbf{v}})\\ (\mathbf{i}_{\mathbf{v}}-\mathbf{1})\mathbf{d}_{\mathbf{v}}\cos(\boldsymbol{\theta}_{\mathbf{v}})\end{pmatrix} \quad (9)$$

where V – matrix whose columns are noise eigenvectors, $a(\varphi,\theta)$ – steering vector of antenna panel with $\boldsymbol{\theta}_{\mathbf{v}}$ – tilt and $\boldsymbol{\varphi}_{\mathbf{h}}$ – rotation of antenna array in horizontal plane. AoAs are estimated from spectrum $\mathbf{S}(\boldsymbol{\varphi},\boldsymbol{\theta})$ via CA-CFAR (Cell Averaging Constant False Alarm Rate) detector:

$$\mathbf{I}[\{\boldsymbol{\varphi}_{\mathbf{n}},\boldsymbol{\theta}_{\mathbf{n}}\}] = \mathbf{I}\left[P(\varphi_n,\theta_n) > \frac{C_{threshold}}{N_\varphi N_\theta}\sum_{\substack{i=-\frac{N_\varphi}{2}\\ i\neq 0}}^{\frac{N_\varphi}{2}}\sum_{\substack{k=-\frac{N_\theta}{2}\\ k\neq 0}}^{\frac{N_\theta}{2}} P(\varphi_{n-i},\theta_{n-k})\right] \quad (10)$$

4.2. Joint Estimation of Spatial Probability Density

Cloud CU receives noisy estimates of angles $\boldsymbol{\Omega}^{\mathbf{est}} = \{\boldsymbol{\varphi}_{\mathbf{n}},\boldsymbol{\theta}_{\mathbf{n}}\}$ from each DU and search for positions by launching rays in close directions $\boldsymbol{\Omega}^{\mathbf{est}} + \boldsymbol{\Omega}^{\mathbf{bias}}$. Intersections of multiple rays from several RRUs define possible emitter locations. Each of positions $\{\mathbf{P}_{\mathbf{x,y,z}}^{\mathbf{i}}\}_{\mathbf{i=1}}^{\mathbf{N_{pos}}}$ is associated with probabilities of received rays and the expectation maximization leads to:

$$\mathbf{P}_{\mathbf{x,y,z}} = \frac{\mathbf{1}}{\sum_{\mathbf{i}}\mathbf{w}_{\mathbf{i}}}\sum_{\mathbf{i}}\mathbf{w}_{\mathbf{i}}\,\mathbf{P}_{\mathbf{x,y,z}}^{\mathbf{i}} \quad (11)$$

Bayesian approach can be used to cope with angle estimation errors caused by interference and residual inter-RRU calibration errors:

$$\mathbf{P}(\boldsymbol{\Omega}_{\mathbf{bias}}|\boldsymbol{\Omega}_{\mathbf{est}}) = \frac{\mathbf{P}(\boldsymbol{\Omega}_{\mathbf{est}}|\boldsymbol{\Omega}_{\mathbf{biased}})\mathbf{P}(\boldsymbol{\Omega}_{\mathbf{biased}})}{\mathbf{P}(\boldsymbol{\Omega}_{\mathbf{est}})} \quad (12)$$

where $\boldsymbol{\Omega}_{\mathbf{est}}$ – biased estimation of angle (azimuth or elevation) received from DU, $\boldsymbol{\Omega}_{\mathbf{bias}}$ – small angle deviation from received direction. $\mathbf{P}(\boldsymbol{\Omega}_{\mathbf{biased}})$ – unique for certain RRU antenna panel and does not depend on UE (e.g., $\mathbf{P}(\boldsymbol{\Omega}_{\mathbf{biased}})\sim\mathbf{N}(\mathbf{0},\frac{|\Omega_{\mathbf{biased}}^{max}|}{\mathbf{3}})$), $\mathbf{P}(\boldsymbol{\Omega}_{\mathbf{est}}|\boldsymbol{\Omega}_{\mathbf{biased}})$ – related to estimation from angular spectrum (peak value and peak width) for each found ray.

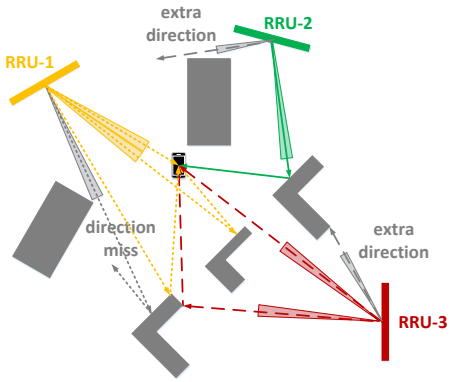


Figure 5. Ray search at CU via RT and known propagation environment.

$$\mathbf{w_i} = \sum_{\mathbf{r=1}}^{\mathbf{N_{rays}}} \mathbf{P}(\boldsymbol{\Omega}_{\mathbf{est}}|\boldsymbol{\Omega}_{\mathbf{biased}})\mathbf{P}(\boldsymbol{\Omega}_{\mathbf{biased}}) \quad \textbf{(13)}$$

At the end of this step each found position $\mathbf{P^i_{x,y,z}}$ is described with its probability $\mathbf{w_i}$. From formula (11) we may find single state maximum expected position. For some cases (e.g., UE has connection to several RRUs) this estimate will be enough to produce the only dense cluster with weighted centre very close to real position.

4.3. Optimal Position Selection

At this stage development of algorithms for joint processing of positions and RT model outputs has a high prospects and research possibilities. RT outputs may include number of rays, number of reflections per each ray, closeness ray to reflecting/diffracting edges, LOS/NLOS recognition, number of BSs, power level indicators, specific hardware limits of antennas and others. Processing of parameters above requires much fewer computing resources than application of RT model. At the same time such algorithm can enhance selection of the most probabilistic positions via recognition of typical patterns. Performance gain can be achieved by taking into account correlations between algorithm positioning errors and physical propagation conditions which are in general related to side RT outputs. As you can see in section 5, even the simplest parameters such as number of BSs or LOS/NLOS indicator are good predictors of errors.

4.4. Clustering of Probabilistic Estimates of Position

In general case erroneous angle acquisition may cause tracking extra directions (e.g., in music algorithm it is caused by interference) or direction miss (e.g., small search area). These errors lead to decomposition of all positions into several remote clusters. Despite probability of fake clusters is typically low, it slightly decreases accuracy of estimation. In order to increase accuracy, we need to perform simple clustering of all positions and exclude fake clusters from consideration.

Clustering algorithm

1. Estimate table of distances. Initialize minimum distance between clusters $d_{cluster}$.
2. Perform DF (Depth-First search) of connected subgraphs with condition of connectivity: $d_{i,k} < d_{cluster}$
3. Find clusters' centers from formula (11) and corresponding variances.

First step of proposed clustering approach has complexity $\mathbf{O(N_{pos}^2)}$, while the second and third steps are linear. In order to exclude storage of large table (memory ~ $\mathbf{O(N_{pos}^2)}$) of distances this algorithm can be replaced with similar with storage of logical vector of added positions (memory ~ $\mathbf{O(N_{pos})}$).

If the following tracking is not being exploited the most likely cluster is used for selection of optimal position (weighted with probabilities / max posterior position / combining / more sophisticated with RT outputs).

4.5. Tracking in Angular Domain

AoA knowledge is equivalent to knowledge about position for accurate 3D maps. Tracking of direct measurement (angles) is more appropriate than mediated because the error distribution for the first one is more understandable, while the second one may produce biased trajectory. Due to calibration residuals, interference and scattering estimate for azimuth/elevation is close to mixture of uniform and normal/cauchy distributions or contaminated ones. For these reasons L2 optimization is less appropriate for tracking [18] rather then Lp (1<p<2)

AoA tracking algorithm

Do for each TTI with new AoA estimate:

1. Start trend from new angle estimate.
2. Check whether new estimate can be used to update current trends.
3. Fit each single trend to a new estimate.
4. Check for redundant trends.
5. Erase noisy, redundant and outdated trends, shrink buffers.
6. Predict angles with corresponding trend deviations.

which is closer to median (L1). Another reason to choose L1 were simulations, resulted in better error reduction mainly due to ability to cope with outliers.

$$\cos(\Omega_{TTI_i} + \Delta\Omega) \approx \cos(\Omega_{TTI_i}) - \sin(\Omega_{TTI_i})\Delta\Omega \quad (14)$$

Figure 6 represents assumptions for developed AoA tracking system. It includes linear change of noised AoA between TTI (from steering vectors (9) and long distance to tracking object (14)), AoA estimation miss, false AoA detection, sudden AoA change (UE changed travelling direction or got around the corner of the building) and irregular TTI structure. AoA tracking is done for each antenna array and azimuth/elevation independently. Lp fit for single trend is described in [19].

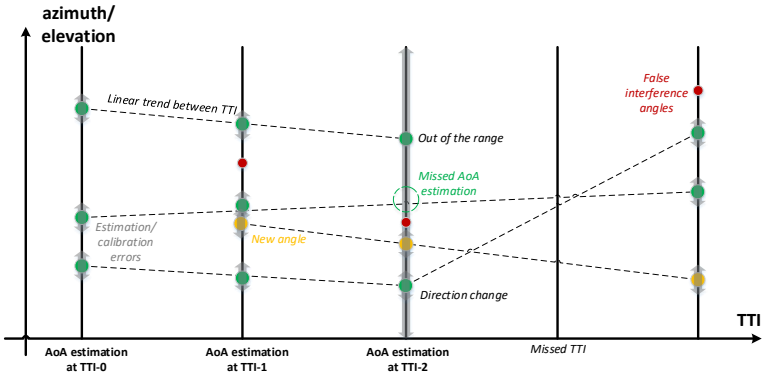


Figure 6. Assumptions for angle tracking model.

Angle predictions are used then for RT positioning with probabilistic weights (13) and additional weights from trend deviations from azimuth and elevation:

$$\mathbf{w_{ray}} = \frac{\mathbf{N_{buffer}}}{\sum_{\mathbf{i=1}}^{\mathbf{N_{buffer}}} |\Omega_i - \mathbf{a_{trend}} \cdot \mathbf{TTI_i} - \mathbf{b_{trend}}|^{\mathbf{p}}} \quad (15)$$

The same tracking system without separation for several trends is used for main cluster tracking. Figure 7 reveals the idea for cluster tracking.

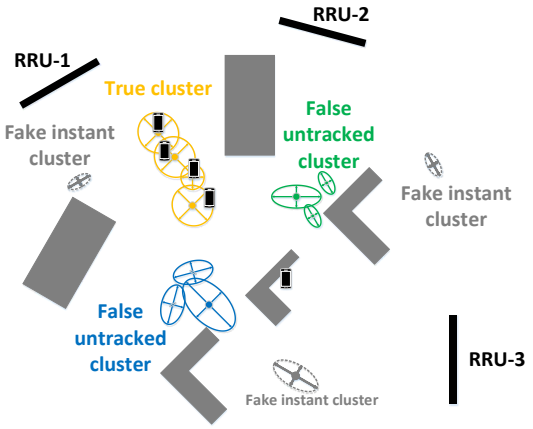


Figure 7. Representation of clusters tracking.

4.6. Accuracy Control

Estimation both position and predicted error is important for combining RT with other positioning approaches. For RT great errors in instant position estimation can be controlled by comparison with measurements for received power and time differences (not taken into account in this paper). Another way is to exploit RT outputs as described in section 4.3. E.g., miss can be estimated via variance of maximum likelihood points:

$$\mathbf{I}\{\mathbf{P_{x,y,z}} = \mathbf{miss}\} = \mathbf{I}\{\mathbf{D}[\mathbf{P^i_{x,y,z}}] > \boldsymbol{\varepsilon_0}\} \quad (16)$$

5. Simulation Results

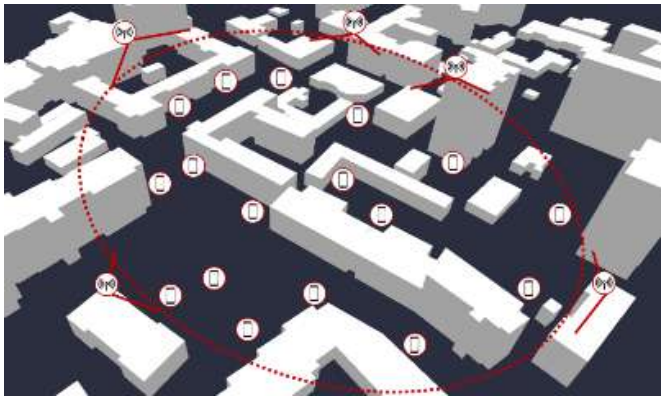


Figure 8. Example of studied Outdoor Environment.

5.1. Positioning Accuracy

Figure 8 represents simulated Urban Environment, parsed from OSM map. Base stations' antenna arrays were placed at the top of the buildings. Figure 9 shows how RT localization principle works:

- base station estimates angular spectrum and detects 3 directions;
- CRAN performs RT approach and detects position of UE

In case of accurate localization estimated CIR might be improved. Very close taps (e.g., LOS and ground reflection) might be distinguished. This result might be also used for more accurate beamforming calculation for open-loop systems.

Figure 10 shows CDF of positioning error of single-state position estimation. It is noticeable that impairments that lead to erroneous angle acquisition also cause dramatic decrease of positioning accuracy. Second graph demonstrates that low accuracy of phase calibration of antenna panel for NLOS cases makes RT positioning without clustering and position selection not robust.

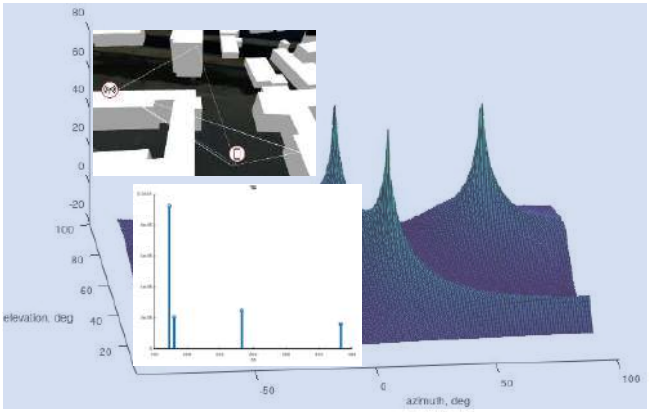


Figure 9. Example of high precision angular spectrum estimation and ray tracing for positioning and CIR reconstruction.

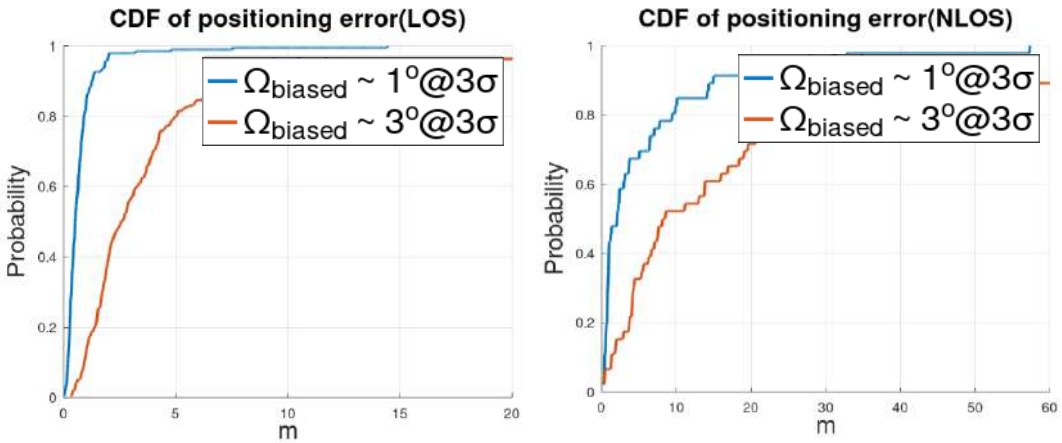


Figure 10. CDF of error of joint estimation of the most likely position (after step 4.2).

Table 1 demonstrates dependence of positioning error on the number of base stations used for SRS acquisition. The high positioning error occur primarily due to the points of poor radio coverage, where the use of several base stations is limited. If we take into account that such points are known in advance, then it is possible to control this error. Research has shown that such areas are related to reception of signal from single collocated antenna array with very low SNR and sparse NLOS channel.

Table 1. Localization accuracy with dedicated maximum number of base stations.

LOS&NLOS +/- 1^0@3σ, without tracking					
Nbs	1	2	3	4	5
90% percentile for absolute error	15m	2m	1.6m	1.3m	1m

Figure 11 represents CDF of positioning error (+/- 2^0) with clustering and rejection of position candidates by the only flexible threshold for number of rays. Previously it was mentioned that impairments can cause fake remote position candidates to appear and decrease accuracy significantly. Such misdetection was solved with selection of the most likely cluster (e.g., in figure 12). Inside cluster several approaches can be used:

- Position with maximum posterior probability
- Most expected position, averaged in cluster

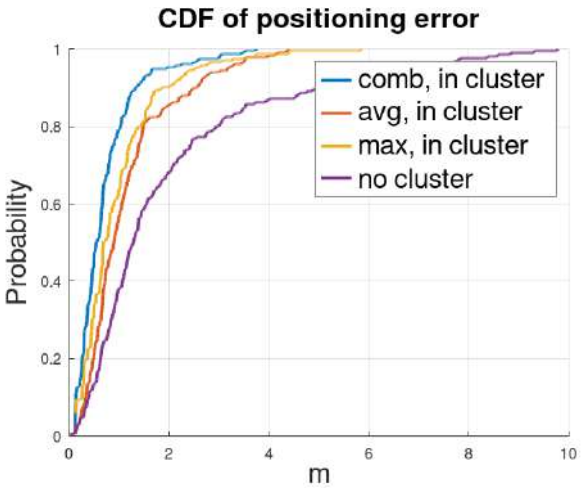


Figure 11. CDF of error of joint estimation of expected position with main cluster selection (after step 4.4).

Both of these strategies are presented in graph as well as their combination. Optimal clustering provides positioning error with Pr = 0.9 less than 1.3m for typical mixed LOS/NLOS Urban Outdoor scenario while without clustering 1.3m is reached with Pr = 0.5. In case of more accurate phase synchronization (+/- 1^0@3σ) at the same probability level localization accuracy reaches 0.9m, which perfectly fits 5G NR requirements for accuracy [20-21].

True positioning error is highly correlated to interim assessment of position variance among a set of possible positions with Pearson correlation coefficient = 0.7, which means that the most of predicted positions can be estimated with very close errors. Better error control as mentioned earlier can be achieved by matching positions with exact patterns from RT outputs.

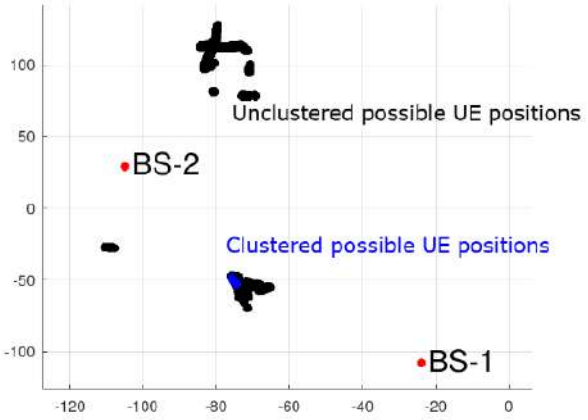


Figure 12. Example of clustering of candidates for UE position

Figures 13-14 represent example of UE trajectory in NLOS and corresponding tracking error. Highest errors (above 1m) happen during direction change in sparse NLOS conditions. After UE enters LOS area (after 310 TTI, TTI = 10 frames) tracking adapts much faster and error mostly does not exceed 1m after convergence.

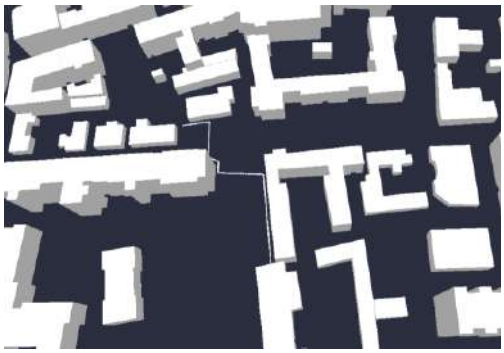


Figure 13. Example of UE trajectory.

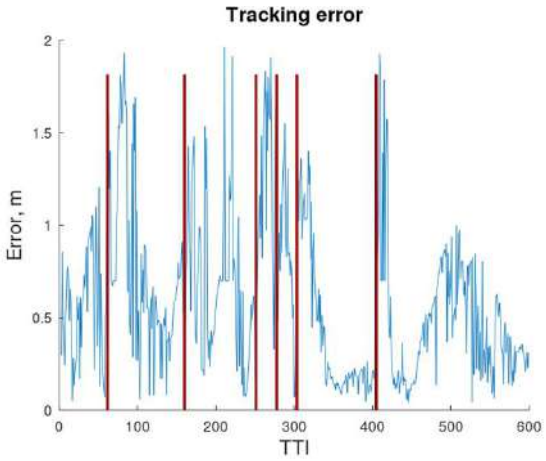


Figure 14. Absolute tracking error across the track (Figure 13). Vertical lines correspond to the moments when UE turns in another direction.

Figure 15 shows UE trajectory in azimuth azimuth/time plane from single base station. Predicted (green) have low level of outliers. Such approach can also be useful for management of frequent UE tracks to enhance positioning via storage of previous measurements and use them to predict optimal multiuser beamforming.

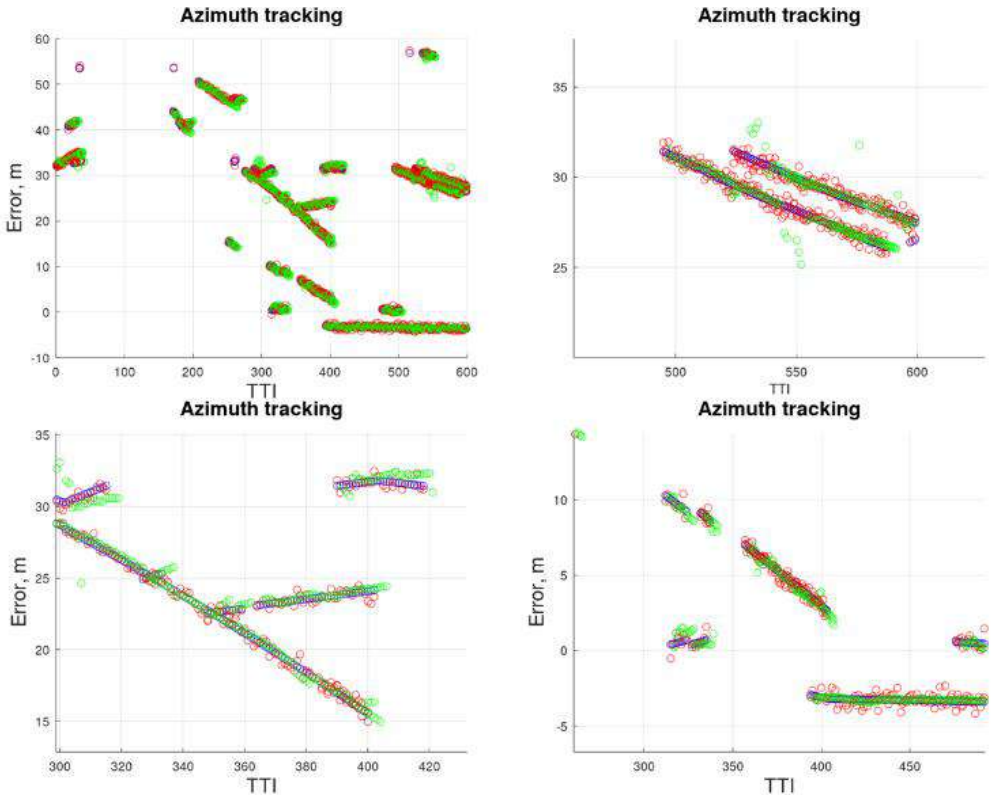


Figure 15. Azimuth tracking of UE trajectory. Blue – true AoA, red – measurement (interference not shown), green – predicted.

5.2. Impact of Inaccurate Maps

Initially the most concerns about positioning accuracy were related to the accuracy of the digital map. These were resolved with RT positioning in the map, where properties of materials (reflection coefficients, scattering model), buildings sizes and positions are pre-set with errors. Results are summarized in table 2. These results are very close to estimate by Monte Carlo method if simulated with known average number of reflections and rays to each UE in 2D space.

Table 2. Contribution of map inaccuracy to localization error.

	Deviation from true value	median	90% percentile
	no	0.7 m	1.3 m
Transmission coefficient	$Uniform(50\%, 200\%)$	1 m	1.7 m
	$Uniform(80\%, 150\%)$	0.8 m	1.5 m
Building, walls position	$N(0\ m, 1\ m)$	1.2 m	2.1 m
	$N(0\ m, 2\ m)$	1.5 m	2.5 m

The most of increase of position error from inaccurate map came from UE with a little number of rays or served by the only base station. Also ray search area has to be increased to partly mitigate error.

5.3. Comparison of Measurements and Simulations

Simulation results above were compared with real GPS measurements in the same outdoor positions as simulated. About 1000 measurements were done both via Samsung Galaxy Note 8 (*GPS Test* software) and iPhone SE (*GPS Data Smart* software). The following metrics were collected for each measurement:

- Horizontal accuracy
  Internal software estimate of positioning accuracy corresponding to 68% confidence interval.
- Measured accuracy
  Absolute difference of positions in Google or Yandex maps and real environment (compared to surrounding buildings)

Table 3 provides results of experiment and compares with simulations (with combined cluster selection, but without tracking). On average proposed approach has a several times higher accuracy than GNSS one.

Table 3. Comparison between accuracy of measurements and simulations.

		median	90% percentile
Note 8	Horizontal accuracy	5.2 m	8.9 m
	Measured accuracy	4.7 m	8.6 m
iPhone SE	Horizontal accuracy	5.0 m	7.9 m
	Measured accuracy	7.6 m	11..2 m
Simulated	1^0@3σ	0.5 m	1.2 m
	3^0@3σ	0.7 m	1.5 m

Figure 16. Example of usage GPS Test and Yandex Maps for single measurement.

6. CONCLUSIONS

This paper covered RT positioning technique for Urban Outdoor scenario and supplemented previously studied positioning algorithms. It was shown that impairments and interference cause pure RT positioning to fail in NLOS. Developed clustering approach for probabilistic candidates for positions mostly mitigated this error and provided almost the same accuracy as for LOS case. Proposed positioning framework was compared to measurements and demonstrated much better accuracy. For 90% Outdoor cases RT positioning may provide ~0.9-2.5m accuracy, which

is enough for the most 5G use cases. Also, it was shown that that proposed algorithm is robust and able to work even in low precision maps.

Developed low complex tracking system in angular domain demonstrated ability to efficiently cope with angle acquisition outliers. This can be used to decrease allocated resources per UE or improve scheduling.

RT positioning has a good internal assessment of error, which enables to combine it with other positioning systems.

7. FUTURE RESEARCH

In the future it is expected to develop prediction model for probabilistic SU/MU-MIMO hybrid precoder with use of RT and angle tracking. Preliminary simulations have shown that such approach is possible to save up to dozens of percent of sum rate for moving pedestrian in urban scenario.

8. DISCUSSION ABOUT IMPLEMENTATION AND PROBLEMS

Proposed RT approach for positioning is beneficial for future RAN at least because it produces new independent estimates for joint filter exploiting GPS, sensors, TDoA from base stations, etc. It does not burden edges with additional large computational resources of traffic and transfer the main load to server.

As mentioned above, RT positioning in general case requires detailed maps. These may be used from external reliable sources. Anyway, below are some ideas for inaccurate map calibration:

- RT approach can be used to shift biased buildings in digital maps. Some areas of maps are covered by several base stations. For multiple UE positions corresponding to the same reflectors for several distributed antenna panels multiple biases in high-resolution FIR taps can be used. Synchronization errors are decreased by multiple pairs of reflectors for different UE positions. Shifting of object borders seems to be solved with GAN models.
- Another approach is offline and exploit RTK receivers for digital map calibration. In this case sets of low cost UEs are replaced with single accurate reducing computation complexity.

Another important issue to mention is fast, low-complex and memory-efficient RT model. For the only purpose of positioning, it can be effectively stored in database containing sets <angle, position, RT outputs> where RT outputs links to another database <index, object description>. RT outputs are any important parameters that can enhance optimal position selection stage or be used for other prediction needs. Many simulated cases have shown that via C++ from several seconds to several days (depends on accuracy, details, map size) may be required to simulate with a single thread. These seems to be opportunistic, because it should be done once to fill the database of several Gb. Also due to rays are independently propagate these computations can be easily parallelized and also some parts of database could be upgraded if propagation environment has changed without complete computation.

ACKNOWLEDGEMENTS

This paper will not be possible without support of my supervisor Anton Laktyushkin. His knowledge and critics let me avoid many common mistakes in this work. I would also like to thank all the colleagues from Huawei Moscow Research Center with whom I had the opportunity to work.

REFERENCES

[1] Sand, S.; Dammann, A.; Mensing, C. Positioning in Wireless Communication Systems; John Wiley & Sons Ltd.: Hoboken, NJ, USA, 2014.
[2] Y. Wang, Z. Shi, Y. Yu, S. Huang and L. Chen, "Enabling Angle-based Positioning to 3GPP NR Systems," 2019 16th Workshop on Positioning, Navigation and Communications (WPNC), 2019, pp. 1-7, doi: 10.1109/WPNC47567.2019.8970182.
[3] Marsch, P., & Fettweis, G. (Eds.). Coordinated Multi-Point in Mobile Communications: From Theory to Practice. 2011. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511783029
[4] 3GPP, TR 138 901, Release 14, Study on channel model for frequencies from 0.5 to 100 GHz, 2017-05
[5] 3GPP, TS 38.211, Release 15, Physical channels and modulation, 2018-07
[6] V. Degli-Esposti, F. Fuschini, E. M. Vitucci and G. Falciasecca, "Measurement and Modelling of Scattering From Buildings," in IEEE Transactions on Antennas and Propagation, vol. 55, no. 1, pp. 143-153, Jan. 2007, doi: 10.1109/TAP.2006.888422.
[7] A. W. Mbugua, Y. Chen, L. Raschkowski, L. Thiele, S. Jaeckel and W. Fan, "Review on Ray Tracing Channel Simulation Accuracy in Sub-6 GHz Outdoor Deployment Scenarios," in IEEE Open Journal of Antennas and Propagation, vol. 2, pp. 22-37, 2021, doi: 10.1109/OJAP.2020.3041953.
[8] Kong, Fanzeng et al. “A GDOP-Weighted Intersection Method for Ray-Trace Based Target Localization using AOA Measurements in Quasi-Specular Environment.” (2016).
[9] del Corte Valiente, Antonio & Gómez, José & Gutiérrez-Blanco, Oscar & Castillo-Sequera, Jose. (2019). Localization Approach Based on Ray-Tracing Simulations and Fingerprinting Techniques for Indoor–Outdoor Scenarios. Energies. 12. 2943. 10.3390/en12152943.
[10] Tayebi, Abdelhamid et al. “The Application of ray-tracing to mobile localization using the direction of arrival and received signal strength in multipath indoor environments.” Progress in Electromagnetics Research-pier 91 (2009): 1-15.
[11] Recommendation ITU-R P.526.11, Propagation by diffraction, The international telecommunication union, 10/2009.
[12] T. Fugen, J. Maurer, T. Kayser and W. Wiesbeck, "Verification of 3D Ray-tracing with Non-Directional and Directional Measurements in Urban Macrocellular Environments," 2006 IEEE 63rd Vehicular Technology Conference, 2006, pp. 2661-2665, doi: 10.1109/VETECS.2006.1683351.
[13] Fabianowski, Bartosz & Dingliana, John. (2009). Compact BVH Storage for Ray Tracing and Photon Mapping.
[14] Andrew S. Glassner, James Arvo, David Kirk. Graphics Gems, I, II, III, Academic Press, 1990-1992.
[15] Daniel Sunday. Practical Geometry Algorithms: with C++ Code, 2021
[16] Möller, Tomas & Trumbore, Ben. (2005). Fast, Minimum Storage Ray-Triangle Intersection. Journal of Graphics Tools. 2. 10.1145/1198555.1198746.
[17] R. Schmidt, "Multiple emitter location and signal parameter estimation," in IEEE Transactions on Antennas and Propagation, vol. 34, no. 3, pp. 276-280, March 1986, doi: 10.1109/TAP.1986.1143830.
[18] Narula, S.C. (1987). The Minimum Sum of Absolute Errors Regression. Journal of Quality Technology, 19, 37-45.
[19] Sposito, V.A., Kennedy, W.J., & Gentle, J.E. (1977). Algorithm AS 110: L p Norm Fit of a Straight Line. Applied statistics, 26, 114-118.
[20] Study on NR positioning support, 3GPP TR 38.855 V1600
[21] Service requirements for the 5G system, 3GPP TS 122 261 - V16.14.0

AUTHORS

Vladislav Ryzhov is a researcher and developer in the field of LTE/5G. He received B.S. degree in MIPT in the field of applied mathematics and physics in 2021 and continued his studies towards M.S. He has an industry experience in Huawei Technologies Co. Ltd. and Yadro.

A PERCEPTIVE PROGRAM TO ASSIST REMOTE LEARNING FOR STUDENTS WITH LEARNING DISABILITIES USING SCREEN AND BLUETOOTH OUTPUT TRACKING

Angelina Zhao[1], Jonathan Sahagun[2]

[1]Crean Lutheran High School, 12500 Sand Canyon Ave, Irvine, CA, 92618
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

In the United States, approximately 1 out of every 44 infants born fall under the autism spectrum, leading them through a life of hardship in the future [7]. Their parents or guardians must pay extra attention to them, as they are not the best at communication and voicing their opinions. As they grow up, their education can fall behind as they struggle in school. Our program allows students to stay on task since a teacher or an adult is always watching over them and guiding them into the right direction. Developed using Python, Thunkable, and Firebase, this program includes 3 major components. The UI component is responsible for the various interactive buttons and boxes [8]. The recording component does the bulk of the job by sending off the data gathered on the student's movements down to the third component, the database which stores all of their information under each individual user ID.

KEYWORDS

Python, iOS, Firebase

1. INTRODUCTION

With the number of students who require extra assistance in order to achieve what others around them are able to achieve, the utilization of assistive technology has increased [1]. Parents of these children have spent hours guiding their young ones to the right direction in succeeding in school, however, the burden that is growing upon their shoulders may eventually be too much to bear. It has been scientifically proven that parents of those whose children may be on the autism spectrum have suffered significantly larger amounts of stress in their everyday lives [2]. The management of family affairs added onto the time spent caring for the child would be the culprits of this additional stress. They must stay attentive to their child in order to correctly teach them the ways of life and success as they see fit. Since the COVID-19 pandemic outbreak in early 2020, many previous classroom based help such as additional teachers have not been able to assist these students in ways that they had before [3]. Their academic progressions have been put on pause since remote learning has proven to be a struggle amongst all students, not just ones who struggle with learning disabilities. Students tend to stray off track and let their mind wander while their teachers make an attempt to educate them through a device screen. In the long run, these COVID-19 students will be faced with difficulties when applying to higher education or landing jobs since they weren't taught the necessary academic skills.

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 165-173, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130514

The first methodology aimed to encompass all possible forms of various ways of learning into one physical sensory device to allow students to learn in the best way that they see fit. Some children may prefer audio over tangible, while others may enjoy sensory over visuals. This solution is extremely beneficial in in-person school, however, it may pose a struggle during online school.

Our proposed method of properly providing these struggling students with a means of focus during class is real time updating data of the student's keystrokes and mouse movements. This would allow teachers to check up on their students and what they're doing in order to ensure maximum participation and concentration. When a student is pressing keys or moving their mouse when they're not supposed to, the teacher or tutor can gently remind them to stay on task to learn the material. We believe that this is the beginning of an extremely effective solution as it allows the teacher to follow up on the child's interactions due to the rapidly updating data collection and the precision of the location of the mouse and the clicks. Due to the nature of those children who fall under the autism spectrum or struggle with school due to a learning disability, they can quickly become distracted or disengage in the lessons. Thus, the accuracy and feedback that the program delivers back to the teacher would reduce the time spent on nonacademic thinking [9]. We've also created a notes section to allow teachers to leave notes of where the child left off on their lesson or if any parents would like to check in on their student's engagement. Not only would this would prove to be efficient in advancing online education, but it would also significantly reduce the stress that would be placed on the parents' shoulders as they have one less thing to worry about. This ensures that students receive proper care and attention without needing to place additional pressure on their guardians.

2. CHALLENGES

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

2.1. The Child's Device Privacy After the Class Ends

A component that might pose as a challenge is the child's device privacy after the class ends. If the program continues to record private information outside of classroom hours, it would cause heavy conflict between the two parties. I could use a function that would allow the student to control when the program would be running, thus protecting their privacy. This function would include a pause button as well, incase emergencies or breaks might occur during classtime. This function could also be implemented on the teacher's side of the app in case a student accidentally turns it off during classtime.

2.2. Sorting the Students

Another component that would cause problems is sorting the students. Two students might have the same name or be born on the same day and so if we were to just use simple names or birthdays to identify and track each student, it would cause heavy confusion as one student may be able to login into another's database. Due to this, I could assign each student a unique, long user ID that would be unable to result in duplicates. This user ID would be assigned to them through the child's email and would show up on the teacher's side with their first and last name.

2.3. A Possible Error when Signing Up

The third challenge that would result from this program is a possible error when signing up. For example, if a student were to not enter an email or enter a password that they typed wrong, it

would be difficult to go back into the system and change their information as a user ID would already have been created [6]. Therefore, I could use a method of double checking. For example, I could implement a second "confirm password" box or I could make the code fail to run if the email box is left blank. Both of these features, if applied to the program, would help reduce the number of struggles when using this app.

3. SOLUTION

The program is split into 3 major components that all link together. The UI component, recording component, and the database. When the teacher first downloads the app, they will be met with various buttons and textboxes. They are able to create an account or log into an existing one through the first screen. Once logged in, they are able to add students or view ones that they had previously had and through each student, they will be able to see everything update in real time if the student is running the program. For the authentication component, we decided to use Firebase [10]. Each adding each student, they are linked with a unique user ID and that would be how each student is identified in the database. The database consists of each teacher that has created an account in the app along with each of their student's names, emails, and last known collected positions or keys pressed.

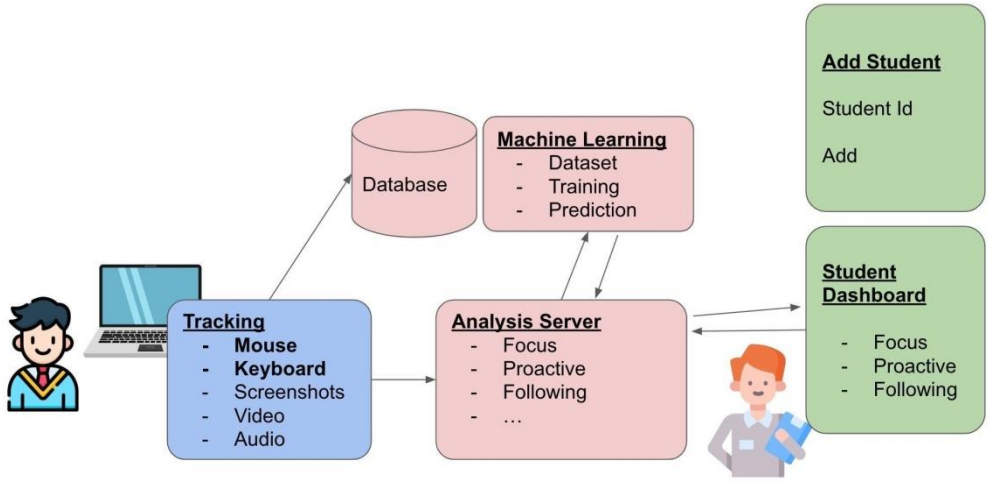


Figure 1. Overview of the solution

The UI component is what the user first sees when they download the application. They are met with a screen that has two buttons, one prompting the user to sign up and the other allows the user to sign in if they are already an existing user. We decided to use Thunkable to approach the UI component.

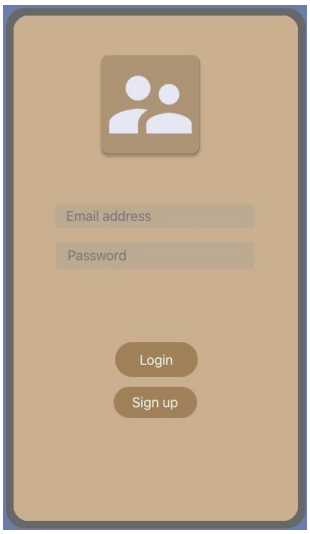


Figure 2. Screenshot of main page

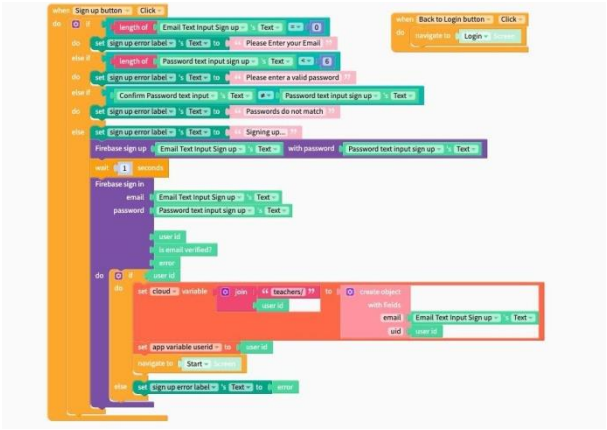


Figure 3. Screenshot of Thunkable page

Since we decided to use Thunkable for the creation of the UI component aspect of our program, the set up is quite simple [15]. At the very beginning of the program, the user is prompted to sign up or sign in. In the signup screen, the code makes sure that the user enters an email and that their entered passwords match. This takes them to the next screen which is their student screen. The buttons allow for the teachers or users to easily access each component for a smoother experience. In the login screen, each of the text boxes are matched with the proper login information in the database so the program can swiftly check if the entered information exists. If there is any mistake, there will be an error popping up on the screen. The verification is for security purposes and allows each teacher to manage each of their students separately and privately.

The recording component is the bulk of the program. We used Python to code the program out. This code tracks the student's movements and presses which then sends the data into the database for the teacher to view. There is a recording stop and start button on the student's end of the program that allows the program to stop tracking their movements once class ends.

Figure 4. Screenshot of tracking movements 1

```
def key_press(key):
    global upload_data
    now = str(int(time()*1000))
    k = str(key)
    upload_data["live"]["last_key"] = k
    upload_data["live"]["keystrokes"][now] = k
    print("you pressed {0}".format(
        key
    ))


def key_release(key):
    print("you released {0}".format(
        key
    ))


def on_mouse_click(x, y, button, pressed):
    global upload_data
    now = str(int(time()*1000))
    clickdata = {"x": x, "y": y}
    upload_data["live"]["clicks"][now] = clickdata
    print("you {0} at {1}".format(
        "pressed" if pressed else "released",
        (x, y)))


def on_mouse_scroll(x, y, dx, dy):
    print("you scrolled {0} at {1}".format(
        "up" if dy > 0 else "down",
        (x, y)))


def on_mouse_move(x, y):
    global upload_data
    upload_data["live"]["x"] = x
    upload_data["live"]["y"] = y
    print('you moved the mouse to {0}'.format(x, y)')
```

Figure 5. Screenshot of code

We used various functions to track the keyboard and mouse movements. Once the Start Recording button is pressed, all of these functions run simultaneously. The key press function not only identifies if a key was pressed or not, but it also reports what key was pressed. The key release function lists when the key previously pressed was released and what key was released.

The mouse click function constantly updates when and where the mouse clicks and when it gets released. The fourth function allows the program to report back when the user scrolls and the direction and placement of the scroll. The last function that the recording would deal with is the mouse direction movement. This would continuously update where the mouse moved until the user decides to stop the recording. All of these functions can be quickly stopped by pressing the enter key if the stop recording button fails.

The last major component of my program is the database aspect. After we made sure everything previously had worked, we needed a place to store all the data in order to ensure the users would be able to track the students using the program. Without a place to store the information, the movements and tracking would not be able to be written down. We decided to use Firebase to store all the information for not only the teacher accounts, but also the student accounts and each individual's movements.

Figure 6. Screenshot of update note

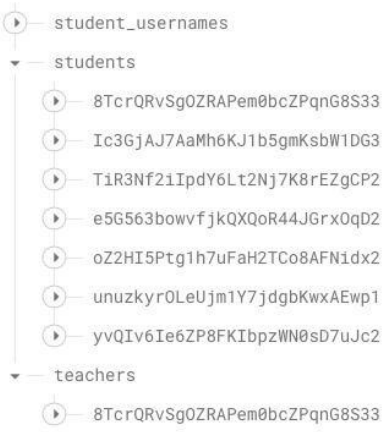


Figure 7. Screenshot of tracking movements 2

The code above tells the program that the data collected should update in to the firebase. The firebase would then show the results in the app that the teacher would use. The database stores all of the student's information so that the teacher can reflect on what the student has done throughout the duration of the class. This also prevents users from losing their information or students needing to create a new account as each user is registered into the database upon signing up. The students and teachers are each uniquely classified by their user ID and upon clicking the drop down arrow, we are able to see all the details that the student or teacher has. Firebase keeps all the data precise so we don't need to worry about error, as long as the program is functioning as expected. Then, the firebase would update the information into the app, allowing teachers to view it from their mobile devices.

4. EXPERIMENT

4.1. Experiment 1

An issue we face with this app is latency [11]. We want our app to be able to see what students are typing in real time. Lag can affect this so we want to minimize it.

We devised a test to measure latency. The test works by creating timestamps measured in milliseconds of when a keystroke is registered on the students' application. That timestamped is uploaded to our database hosted by Firebase [12]. On the instructor's application add code to record the current time again in milliseconds when the data is retrieved from Firebase. We subtract the time recorded on Firebase from the time recorded in the instructor's application to calculate the latency [13]. This process is repeated multiple times and we take the average of the latencies.

Trial	Time Data Uploaded (ms)	Time Data Retrieved (ms)	Latency (ms)
1	1670097143283	1670097144175	892
2	1670097550420	1670097551078	658
3	1670098731909	1670098732288	379
4	1670099004928	1670099006090	1162
5	1670099465699	1670099466577	878
Average			793.8

Figure 8. Table of experiment

Having a latency time of less than 1 second allows us to accurately measure the students' interactions with the application in real-time. This is important for providing feedback to the students on their typing and attention to the task at hand. This also ensures that the instructor has access to up-to-date information on the students' progress, allowing them to make adjustments to the instruction if necessary. Overall, the results of our experiment are encouraging, and we are confident that our application will provide valuable insights into students' focus.

To truly measure the effectiveness of our app, we need to see if students use it and if it can be a useful tool for instructors.

For this experiment, we gathered students and instructors to test our app to test our app. The experiment simply asked the student on a scale of 1-10 how easy the app was to set up and if they noticed the app while it was in use. For the instructor's side we asked how on a scale of 1-10 how useful information provided was and how likely they were to use the app again.

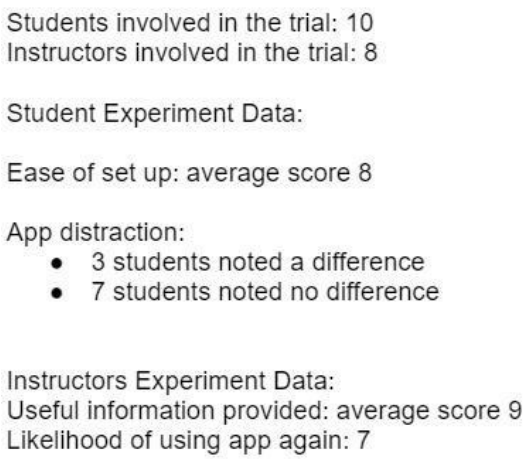


Figure 9. Result of experiment

From the preliminary results, we can conclude our app is on the correct path of being effective for monitoring students' focus. We realize we have a small sample size but we are positive in expanding our app and trials to more users to further gather and compile more data on the effectiveness of this type of student monitoring.

5. RELATED WORK

Raja, S. K. S., & Balaji, V. developed a sensory learning based device that would help children with learning disabilities. It allows for the child to intake learning information in different forms as some may prefer audio over visual and vice versa [4]. This is quite different from our application since it relies on the child to be present in person to be able to access this sensory device. This solution is a great approach, however, due to the state of the environment as of now, it's not in our best interest to expose these children to any more viruses than necessary which is why our believe my program covers the online aspect.

Purnama, Y., Herman, F. A., et al created an application that is meant to be installed and used from a tablet. It incorporates different activities such as identifying opposite words or matching images given a set example, with various difficulties to the student's liking [5]. We believe that this solution can be quite effective for the younger students, however it would pose a challenge to create such interactive activities for the older students who are learning more advanced things.

Our program is different as it monitors the movement of the students instead of providing the problems that the student would be completing.

6. CONCLUSIONS

We believe that a limitation to our project is the mouse trail. Hypothetically, if the mouse was moved from point A to point B, our program as of now, has no record of whether or not the mouse moved directly from point A to point B or if the mouse circled around before stopping. We would implement a mouse tracer that would map out the route that the mouse took before it landed in a specific spot. This would ensure that the teachers would be able to see exactly what their student is drawing or writing. Another limitation to my project would be the attention span of the student. If we had more resources and time, we would consider implementing a software that would report to the teachers if the student was paying attention to the screen or not. A camera

would be able to track the pupils and attention focus point and would allow for even higher rates of engagement.

The future of the children who lie the autism spectrum is always slightly improved every time someone believes that they are able to help those who need it [14]. Although our program has many aspects that can be improved for the future good of it, it strives to better these children's academic careers.

REFERENCES

[1] Raskind, Marshall H., and Eleanor L. Higgins. "Assistive technology for postsecondary students with learning disabilities: An overview." Journal of Learning Disabilities 31.1 (1998): 27-40.

[2] Bonis, Susan. "Stress and parents of children with autism: A review of literature." Issues in mental health nursing 37.3 (2016): 153-163.

[3] Frankova, Helena. "The impact of COVID-19 on people with autism, learning disabilities and mental health conditions." Nursing and Residential Care 22.6 (2020): 1-3.

[4] Raja, S. Kanaga Suba, and V. Balaji. "Sensor based learning device for children with autism." Materials Today: Proceedings 50 (2022): 307-311.

[5] Purnama, Yudy, et al. "Educational software as assistive technologies for children with Autism Spectrum Disorder." Procedia Computer Science 179 (2021): 6-16.

[6] Fergus, Paul, et al. "Interactive mobile technology for children with autism spectrum condition (ASC)." 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC). IEEE, 2014.

[7] Johnson, Chris Plauché, and Scott M. Myers. "Identification and evaluation of children with autism spectrum disorders." Pediatrics 120.5 (2007): 1183-1215.

[8] Sun, Xiaolei, Tongyu Li, and Jianfeng Xu. "Ui components recognition system based on image understanding." 2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C). IEEE, 2020.

[9] Zhang, Li-fang. "Validating the theory of mental self-government in a non-academic setting." Personality and Individual Differences 38.8 (2005): 1915-1925.

[10] Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app development-a study." International Journal of Computer Applications 179.46 (2018): 49-53.

[11] Iorio, Marco, Fulvio Risso, and Claudio Casetti. "When latency matters: measurements and lessons learned." ACM SIGCOMM Computer Communication Review 51.4 (2021): 2-13.

[12] Moroney, Laurence, and Laurence Moroney. "An Introduction to Firebase." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform (2017): 1-24.

[13] Moroney, Laurence, and Laurence Moroney. "The firebase realtime database." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform (2017): 51-71.

[14] Newschaffer, Craig J., et al. "The epidemiology of autism spectrum disorders." Annu. Rev. Public Health 28 (2007): 235-258.

[15] Klemisch, Kerstin, Ingo Weber, and Boualem Benatallah. "Context-aware UI component reuse." Advanced Information Systems Engineering: 25th International Conference, CAiSE 2013, Valencia, Spain, June 17-21, 2013. Proceedings 25. Springer Berlin Heidelberg, 2013.

AN FRFT BASED REAL-TIME ESTIMATION OF MOVING TARGET ACCELERATION METHOD FOR FMCW RADAR

Qingbo Wang

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

ABSTRACT

This paper considers the problem of real-time estimating the moving target acceleration in frequency modulated continuous wave (FMCW) radar. Based on the accelerated target FMCW radar echo signal model, after utilizing KeyStone transform to eliminate the effect of range migration on the signal parameters estimation. An improved fractional Fourier transform (FrFT) and optimized the best matching rotation angle search strategy is proposed to estimate the chirp rate of doppler dimension echo signal related to the target acceleration. Compared with the traditional FrFT, the approach in this paper has less computation and significantly reduced processing time while ensuring estimation accuracy. The proposed method is demonstrated with simulation and measurement data.

KEYWORDS

FMCW Radar, Acceleration Estimate, Range Migration, Fractional Fourier Transform, Linear Frequency Modulated, Chirp Rate

1. INTRODUCTION

Current-generation FMCW automotive radar can accurately measure the velocity of moving targets [1]. However, many targets with high acceleration exist in actual driving scenarios. The lack of acceleration information may cause the automated driving system to be unable to accurately predict the target state and make proper path planning and decisions, which leads to an increased probability of traffic accidents. Therefore it is necessary to implement real-time measurement of moving target acceleration in FMCW radar systems.

Suppose the moving target is accelerated. Its echo signal in the doppler dimension can be regarded as a linear frequency modulated (LFM) signal. The acceleration estimation of the target can be turned into an LFM signal parameter estimation problem [2]. In addition, the range migration will affect the doppler dimensional signal phase, resulting in the signal being divided into multiple parts in the time-frequency plane and affecting the parameter estimation of the signal. The KeyStone transform is needed to decouple the range and doppler dimension before range FFT.

The solutions of LFM signal parameter estimation can be divided into two categories: time-frequency representation (TFR) and frequency-chirp rate representation (FCR) [9]. Short Time Fourier Transform (STFT) [3] as a typical example of linear TFR is widely used because of its simple and convenient implementation. Wigner-Ville distribution (WVD) [4] as representative of quadratic class TFR has perfect time-frequency concentration when estimating the mono-

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 175-189, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130515

component signal parameter. However, WVD-based methods have cross-term problems when dealing with multi-component LFM signals. Although various methods have been proposed to reduce cross-term effects, such as Pseudo Wigner Distribution (PWD) [5], smoothed Pseudo Wigner Distribution (SPWD) [6]. These methods are realized at the expense of time-frequency resolution. In addition, TFR-based methods can not directly obtain the parameters estimation of the LFM signal. The line detection algorithms are additionally needed to get the signal chirp rate, such as the Radon-Wigner transform (RWT) [7], Wigner-Hough transform (WHT) [8]. This significantly increases the computational cost and limits the application in engineering. Lv's Distribution (LVD) [9] as the FCR class method proposed in recent years. It can directly obtain the chirp rate of the LFM signal without additional calculations. Compared to WVD, LVD can avoid interference by cross terms and is more easily implemented in engineering. However, the maximum chirp rate estimation of LVD is affected by the sampling rate, which causes the estimated acceleration to be ambiguous.

FrFT [10] as the generalization of the traditional Fourier transform, widely used in LFM signal detection and parameter estimation. FrFT converts the LFM signal from the time domain to the fractional Fourier domain by rotating the time-frequency plane. It has the best energy concentration in a specific fractional Fourier domain. Moreover, The discrete FrFT proposed by Ozaktas [11], Pei [12] makes FrFT can be implemented by FFT and chirp signal convolution. It is well suited for engineering applications.

This paper proposes a new algorithm for LFM signal chirp rate estimation based on the FrFT. It simplifies the traditional FrFT and has a more concise form. And based on the feature that the fractional domain spectrum of the improved FrFT is symmetric on both sides of the optimal rotation angle. An optimized search strategy is applied to avoid the global search of the rotation angle, which significantly reduces the processing time. After implementing the algorithm in a radar system, the actual environment tests proved that the algorithm could effectively and accurately estimate the acceleration of a moving target.

2. FMCW RADAR SIGNAL AND SYSTEM MODELS

In order to measure target range and velocity, the FMCW radar will transmit a series of LFM signals (or chirp pulses) with a total number of *N* through the transmitting antenna (Tx).

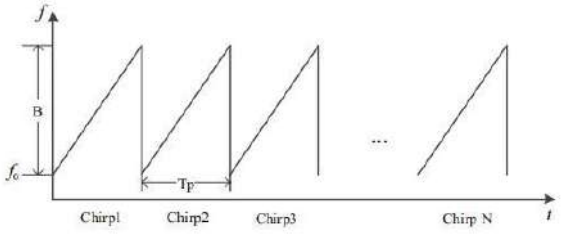


Figure 1. FMCW continuous transmit pulses time-frequency figure

As shown in Figure 1, the LFM signal with start frequency f_0 and its frequency increases linearly at rate of K over the time duration T_p [1].

2.1. IF Signal Model

The LFM signal transmitted by Tx can be expressed as:

$$S_t(t) = \exp[\, j(2\pi f_0 t + \pi K t^2)] \quad t \in (0, T_p) \tag{1}$$

When the transmitted signal strikes the target, it will reflect back and be received by the receiving antenna (Rx). The signal received by the Rx can be seen as a version of the transmitted signal after time delay τ. That is:

$$S_r(t) = \exp\{j[2\pi f_0(t-\tau) + \pi K(t-\tau)^2]\} \quad t \in (0, T_p) \tag{2}$$

When the radar system receives the signal reflected from the target, the echo signal is mixed with the transmitted signal through a mixer. The resulting signal is filtered low-pass to produce an intermediate frequency (IF) signal $S_I(t)$. The IF signal can be expressed as:

$$S_I(t) = S_t(t) \cdot S_r^*(t) = \exp\{j[2\pi(f_0\tau + K\tau t) - \pi K\tau^2]\} \quad t \in (0, T_p) \tag{3}$$

Assuming that there exists a moving target in front of the radar, the initial radial distance between radar and target is R_0. The target moves away from the radar with radial velocity v and constant radial acceleration a. Considering that the chirp pulse duration T_p is very short, the movement of the target within one chirp pulse is negligible. Only the movement between the two chirp pulses is considered. Therefore, the "Stop-and-Go" model is applied to establish for target echo signal. Based on the above assumption, one can get time delay of the nth chirp pulse:

$$\tau_n = \frac{2(R_0 + vnT_p + \frac{1}{2}an^2T_p^{\,2})}{c} \quad n \in [0, N-1] \tag{4}$$

where c is the speed of light. Substituting τ_n into IF signal $S_I(t)$ and neglecting the τ^2 term in equation (3) (because $\tau^2 \approx c^{-2}$, its value is very small), the nth IF signal can be derived as follows:

$$S_I(t,n) = \exp\left\{ j2\pi \left[\frac{2R_0 f_0}{c} + \left(\frac{2R_0 K}{c} + \frac{2KvnT_p}{c} + \frac{Kan^2T_p^{\,2}}{c} \right) t + \frac{2f_0 v}{c} nT_p + \frac{f_0 a}{c} n^2 T_p^{\,2} \right] \right\} \tag{5}$$
$$t \in (0, T_p) \quad n \in [0, N-1]$$

To perform digital signal processing, the IF signal $S_I(t)$ is sampled with sampling frequency f_s and sampling period $T_c = 1/f_s$. Total M points are sampled. The sampled IF signal can be expressed as:

$$S_I(m,n) = \exp\left\{ j2\pi \left[\frac{2R_0 f_0}{c} + \left(\frac{2R_0 K}{c} + \frac{2KvnT_p}{c} + \frac{Kan^2T_p^{\,2}}{c} \right) mT_c + \frac{2f_0 v}{c} nT_p + \frac{f_0 a}{c} n^2 T_p^{\,2} \right] \right\} \tag{6}$$
$$m \in [0, M-1] \quad n \in [0, N-1]$$

After sampling the IF signal for all periods, we can get an $M*N$ two dimensional discrete time signal $S_I(m,n)$. One of the dimensions is fast time dimension (or range dimension) mT_c and the another is slow time dimension (or doppler dimension) nT_p. It is clear in equation (6), the frequency and phase of IF signal is:

$$f_{IF} = \frac{2R_0 K}{c} + \frac{2KvnT_p}{c} + \frac{Kan^2T_p^{\,2}}{c} \tag{7}$$

$$\varphi_{IF} = 2\pi\left(\frac{2R_0 f_0}{c} + \frac{2f_0 v}{c} nT_p + \frac{f_0 a}{c} n^2 T_p^{\,2}\right) \tag{8}$$

In equation (7), due to the presence of $2KvnT_p$ and $Kan^2T_p^{\,2}$ terms, IF signal frequency is not a constant which is related to the chirp pulse period number n. So fast time and slow time coupled with each other. As a result, when the target is moving at high speed and acceleration, the central

of range spectrum envelope will be shifted significantly. This phenomenon leads to an additional phase error of the spectrum peak between the range cell shifts after the range FFT, affecting the estimation of the chirp rate along the doppler dimension signal.

2.2. Range Migration Analysis

Performing K points range FFT for discrete IF signal (6), one can get:

$$\begin{aligned} X(k,n) &= \sum_{m=0}^{M-1} S_I(m,n)\exp(-j2\pi k\frac{m}{M}f_s T_c) \\ &= \exp(j\varphi_{IF})P_M(\frac{k}{M}f_s - f_{IF}) \end{aligned} \tag{9}$$

where $P_M(\cdot)$ can be regarded as a discrete Fourier transform with a rectangular window of length M. The result is an asinc function, which is a discrete form of the continuous sinc function. The expression of $P_M(\cdot)$ is:

$$\begin{aligned} P_M(f) &= \sum_{m=0}^{M-1}\exp(j2\pi fmT_c) \\ &= \frac{\sin(\pi fMT_c)}{\sin(\pi fT_c)}\exp[-j\pi f(M-1)T_c] \end{aligned} \tag{10}$$

Based on equations (9) and (10), one can get the peak phase φ_{peak} of the range spectrum:

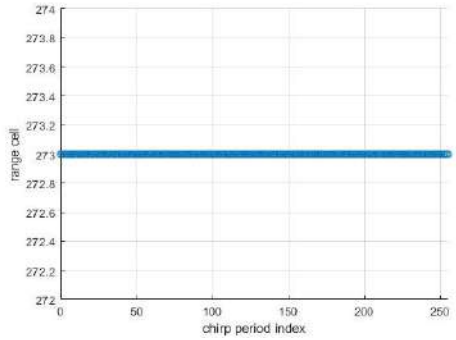
$$\varphi_{peak} = \varphi_{IF} - \pi(\frac{k_{peak}}{M}f_s - f_{IF})(M-1)T_c \tag{11}$$

where k_{peak} denotes spectrum peak index. Furthermore, the IF signal doppler frequency f_d can be viewed as the result of the derivation of phase φ_{peak} with respect to slow time nT_p. If the peak index k_{peak} is not dependent on variable nT_p, the f_d can be derived as:

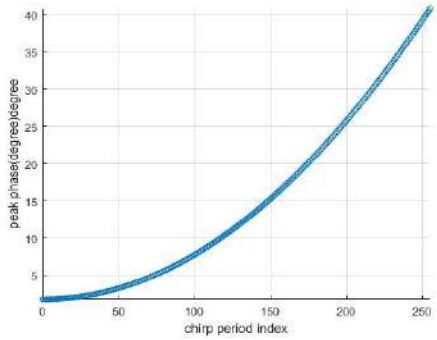
$$\begin{aligned} f_d &= \frac{1}{2\pi}\frac{d\varphi_{peak}}{dnT_p} \\ &= \left[\frac{2f_0 v}{c} + \frac{Kv(M-1)T_c}{c}\right] + \left[\frac{2f_0 a}{c} + \frac{Ka(M-1)T_c}{c}\right]nT_p \end{aligned} \tag{12}$$

Obviously, under the above assumptions, f_d is a linear function of variable nT_p, and the chirp rate is related to the acceleration of the target. By using the LFM signal parameters estimation algorithm, one can get the target acceleration estimated. But in practice, this is not a common scenario, the change of k_{peak} must be taken into account.

Consider a single TX radar system example. The chirp pulse parameters are as follows: f_0 = 77GHz; f_s = 5000sps; T_p = 110us; K = 40MHz/us; N = 256. The target initial velocity and acceleration are set to 0m/s and 30m/s^2. Performing range FFT for all chirp pulse periods. The index of range spectrum peak for each period is shown in Figure 2(a). All peaks appear at 273 range cell. Figure 2(b) is the result of unwrapping phase for spectrum peaks. Without range cell shift, the peak phase is a quadratic function of chirp periods index n. Compared to Figure 2, Figure 3 shows the spectrum peak index and phase for target acceleration 300m/s^2. There exist three shifts in range cell. Furthermore, the phase also have shifts at the same position.

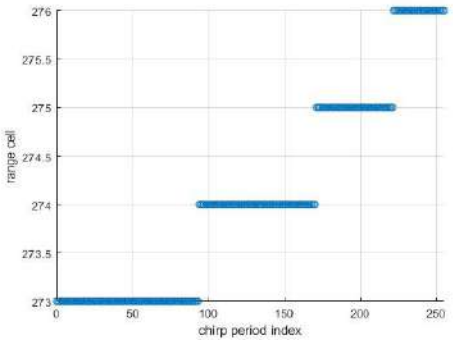


(a) peak index for all chirp periods

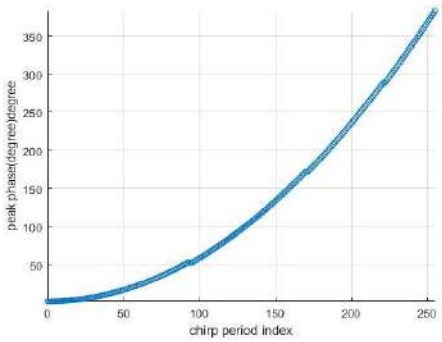


(b) peak phase without range cell shift

Figure 2. Range FFT results without range migration



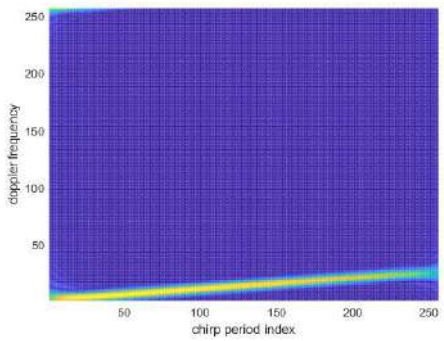
(a) peak index for all chirp periods



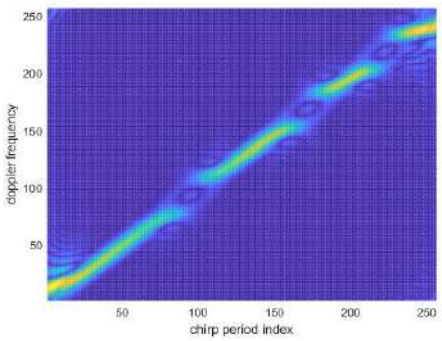
(b) peak phase without range cell shift

Figure 3. Range FFT results with range migration

If we represent the peak phase of the above two cases in the time-frequency plane, the result is shown in Figure 4. When range cell shift is not present, the doppler signal is linearly distributed in the time-frequency plane as shown in Figure 4(a). Comparatively, when range cell is shifted, the doppler signal time-frequency distribution is divided into four segments, which is depicted in Figure 4(b). Ideally, the chirp rate of each segment is still the same. However, when there is noise in the signal, the chirp rate of each segment may vary greatly. In this case, the estimation error is much larger.



(a) peak index for all chirp periods



(b) peak phase with range cell shift

Figure 4. Doppler dimensional signal time-frequency distribution

2.3. KeyStone Transform

The KeyStone transform is well known in the field of synthetic aperture radar (SAR) for its capability of eliminating moving target range migration. According to the introduction in the previous section, the FMCW radar also has similar problems and affects the acceleration estimation of moving targets. Therefore, the KeyStone transform is applied before range FFT to decouple the fast and slow time. Based on the principle of KeyStone transform, putting the slow time terms in equation (6) together, one has:

$$S_I(m,n)=\exp\left\{j\left[2\pi\frac{2R_0f_0}{c}+2\pi\frac{2R_0K}{c}mT_c+2\pi(f_0+KmT_c)\frac{2vnT_p}{c}+2\pi(f_0+KmT_c)\frac{an^2T_p{}^2}{c}\right]\right\} \quad (13)$$

Re-scale the slow time axis for each fast time sample point by follow transformation:

$$(f_0+KmT_c)nT_p=f_0n'T_p \quad (14)$$

Applying the scaling operator to (13), one can get:

$$S_I(m,n')=\exp\left\{j\left[2\pi\frac{2R_0f_0}{c}+2\pi\frac{2R_0K}{c}mT_c+2\pi f_0\frac{2vn'T_p}{c}+2\pi f_0\frac{an'^2T_p{}^2}{c}\right]\right\} \quad (15)$$

Equation (15) indicates that for the new slow time $n'T_p$, the coupling between the slow time and fast time has been removed. By performing range FFT on (15), the central signal spectrum envelope will be located at $2R_0K/c$, which is only related to the initial distance of the target R_0. That is, the range FFT for all chirp periods will peak at the same index. The range spectrum peak phase and doppler frequency can be re-expressed as:

$$\varphi'_{peak}=2\pi[\frac{2R_0f_0}{c}+\frac{2f_0v}{c}nT_p+\frac{f_0a}{c}n^2T_p{}^2-(\frac{k_{peak}}{M}f_s-\frac{2R_0K}{c})\frac{(M-1)T_c}{2}] \quad (16)$$

$$f'_d=\frac{1}{2\pi}\frac{\varphi'_{peak}}{dnT_p}=\frac{2f_0v}{c}+\frac{2f_0a}{c}nT_p \quad (17)$$

From the above equation, it can be seen that after KeyStone transform, without k_{peak} term influence, the doppler dimension signal changes to LFM signal that chirp rate only related to target acceleration.

The KeyStone transform can be implemented by the Chirp-Z transform and realization details can be referred to [13]. This paper will not discuss its implementation. In the following, we will analyze how to estimate the LFM signal chirp rate by improved FrFT.

3. THE LFM SIGNAL CHIRP RATE ESTIMATION VIA IMPROVED FRFT

3.1. Fractional Fourier Transform

Firstly, let us briefly reviews the definition of FrFT. Consider a mono-component LFM signal $f(t)$. For the convenience of derivation, the initial frequency of the $f(t)$ is set to 0.

$$f(t)=A\exp(j\pi Kt^2)\quad t\in(-T_d/2,T_d/2) \quad (18)$$

The formal FrFT of $f(t)$ at an arbitrary rotation angle α is given by equation (19):

$$F^{\alpha}[f(t)] = X_{\alpha}(u) = \int_{-\infty}^{+\infty} f(t) K_{\alpha}(t,u) dt \tag{19}$$

where $K_{\alpha}(t,u)$ is the transform kernel and defined by (20):

$$K_{\alpha}(t,u) = \begin{cases} \sqrt{1-j\cot\alpha}\, e^{j(\pi u^2 \cot\alpha - 2\pi u \csc\alpha + \pi^2 \cot\alpha)}, & \alpha \neq k\pi \\ \delta(t-u), & \alpha = 2k\pi \\ \delta(t+u), & \alpha = (2k+1)\pi \end{cases} \tag{20}$$

FrFT can be divided into the four steps:

(1) Perform de-chirp operation by modulate with a chirp $\exp(j\pi t^2 \cot\alpha)$.
(2) Perform scale Fourier transform of the de-chirped signal by the scale factor $\csc\alpha$.
(3) Perform another modulate operation with chirp $\exp(j\pi u^2 \cot\alpha)$.
(4) Multiply amplitude factor $\sqrt{1-j\cot\alpha}$.

The FrFT can be regarded as the process of signal projection on different chirp base functions. The energy intensity of signal in FrFT domain reflects the degree of similarity between the signal and chirp base functions. It will peak at the optimal rotation angle only when the parameters of the chirp base functions match the sought signal.

3.2. Improved Fractional Fourier Transform

Step 1 includes a chirp modulation, which makes FrFT adaptive to process the LFM signal. *Step 2* transforms the signal from the time domain to the FrFT domain. *Step 3* and *4* are performed to guarantee the FrFT additivity of rotation property and energy conservation property. The additivity property is important in some applications, but our purpose is to obtain the signal chirp rate. Whether *Step 3* is performed or not does not significantly affect the results. In consideration of saving hardware resources, we combine *Step 1*, *Step 2* and *4* to form an improved FrFT. The improved FrFT and its kernel can be expressed as:

$$\hat{F}^{p}[f(t)] = \hat{X}_{\alpha}(u) = \int_{-\infty}^{+\infty} f(t) \hat{K}_{\alpha}(t,u) dt \tag{21}$$

$$\hat{K}_{\alpha}(t,u) = \sqrt{1-j\cot\alpha}\, e^{j(\pi^2 \cot\alpha - 2\pi u \csc\alpha)} \quad \alpha \in (0,\pi) \tag{22}$$

Since the improved FrFT no longer has the rotational additive property after omitting *Step 3*, we set the rotation angle scope from *0* to π. This is sufficient to cover the possible scope of the unknown parameters LFM signal chirp rate.

The improved FrFT is a special form of LCT. If we set a special parameter matrix $A = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\csc\alpha & 0 \end{bmatrix}$, the LCT reduces to improved FrFT [14]. Using the properties of the LCT, we can derive some important properties of improved FrFT, which are essential to the subsequent analysis.

(1) The time-shift property

$$\hat{F}^{p}[f(t-\tau)] = \hat{X}_{\alpha}(u - \tau\cos\alpha) e^{j(\pi\tau^2 \cot\alpha - 2\pi\tau u \csc\alpha)} \tag{23}$$

(2) The frequency-shift property

$$\hat{F}^{p}\left[f(t)e^{jvt}\right] = \hat{X}_{\alpha}(u - v\sin\alpha) \tag{24}$$

(3) The invertibility property

$$f(t)=\int_{-\infty}^{+\infty}\hat{X}_{\alpha}(u)\hat{K}_{\alpha}^{*}(u,t)du \tag{25}$$

(4) The energy conservation property

$$\int_{-\infty}^{+\infty}|f(t)|^{2}\,dt=\int_{-\infty}^{+\infty}|\hat{X}_{\alpha}(u)|^{2}\,dt \tag{26}$$

Similar to FrFT, the improved FrFT can also be expressed in terms of the WVD. The WVD of signal $f(t)$ is defined as:

$$WVD(t,\omega)=2e^{j2\omega t}\int_{-\infty}^{+\infty}f(\tau)f^{*}(2t-\tau)e^{-j2\omega\tau}d\tau \tag{27}$$

Applying the time shift property to the $f^{*}(2t-\tau)$ term of the above equation:

$$WVD(t,\omega)=2e^{j2\omega t}\int_{-\infty}^{+\infty}\hat{X}_{\alpha}^{*}(-u+2t\cos\alpha)e^{-j(2t^{2}\cot\alpha-2tu\csc\alpha)}\int_{-\infty}^{+\infty}x(\tau)\hat{K}_{\alpha}(\tau,u)e^{-j2\omega\tau}d\tau du \tag{28}$$

Then use frequency-shift property to compute inner integral in equation (28) and make the change of variable $\varepsilon = u + 2\omega sin\alpha$, it comes to:

$$WVD(t,\omega)=2\int_{-\infty}^{+\infty}\hat{X}_{\alpha}(\varepsilon)\hat{X}_{\alpha}^{*}(-\varepsilon+2t\cos\alpha+2\omega\sin\alpha)e^{-j(2t^{2}\cot\alpha-2t\varepsilon\csc\alpha+4\omega t)}d\varepsilon \tag{29}$$

Finally, making following change of variables from (t,ω) to (u,v):

$$\begin{aligned}u&=t\cos\alpha+f\sin\alpha\\ v&=-t\csc\alpha\end{aligned} \tag{30}$$

The final result can be obtained as:

$$WVD(t,\omega)=2e^{j2uv}\int_{-\infty}^{+\infty}\hat{X}_{\alpha}(\varepsilon)\hat{X}_{\alpha}^{*}(2u-\varepsilon)e^{-j2v\varepsilon}d\varepsilon \tag{31}$$

From the perspective of axis rotation, the improved FrFT of signal can still be seen as the result of rotating the t - f axis to the u - v axis. Compared to the traditional FrFT, the difference is only reflected in the position of the v-axis. The rotation of the signal from the time t-axis to the u-axis is entirely the same. That is say, after performing improved FrFT, the spectral support interval of the signal in u-axis is the same as traditional FrFT. It can be represented in Figure 5.

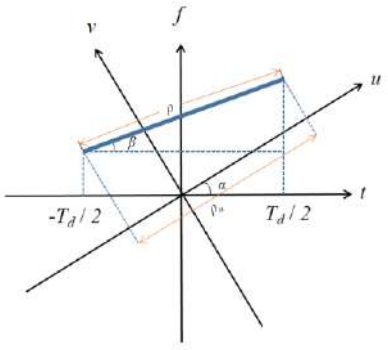


Figure 5. Representation of LFM signal in the t-f and u-v plane

The blue line in Figure 5 is the distribution line of the signal in the time-frequency plane, whose angle with the time axis t is β. The width of the signal in the time-frequency plane can be expressed as:

$$\rho = \frac{T_d}{|\cos\beta|} \tag{32}$$

Considering the improved FrFT with rotation angle α. ρ_u in Figure 5 is the projection length of the LFM signal in the u-axis after the transformation. According to the geometric relationship in Figure 5, the expression of ρ_u can be obtained as:

$$\rho_u = \left| \frac{T_d \cos(|\alpha - \beta|)}{\cos\beta} \right| \tag{33}$$

Since the angle between signal and the time axis is fixed, the magnitude of ρ_u is related to the rotation angle α. When $\alpha = \beta + \pi/2$, the width of the support interval becomes a dot where the energy of the signal is concentrated. Otherwise, the energy is divided equally by the support interval, which approximates a rectangle inside the support interval, and outside the rectangle, the energy of the signal drops sharply. The expressions for the energy magnitude in the support interval are derived following by definition of improved FrFT.

When the rotation angle α is the finite length LFM signal $f(t)$ optimal rotation angle, i.e. $\alpha = -\text{arccot}(-K)$. Equation (21) can be turned into:

$$\begin{aligned} \hat{X}_\alpha(u) &= A\sqrt{1-j\cot\alpha}\int_{-T_d/2}^{T_d/2} e^{j[\pi(K+\cot\alpha)t^2 - 2\pi u\csc\alpha]}dt \\ &= A\sqrt{1-j\cot\alpha}\,T_d \sin c(\pi T_d u\csc\alpha) \end{aligned} \tag{34}$$

From the equation (34), it can be seen that the transformation of the signal at the optimal rotation angle results in a sinc function. The peak is located at $u = f_0\csc\alpha$.

When $\alpha \neq -\text{arccot}(-K)$, after some derivations of equation (21), one can get:

$$\hat{X}_\alpha(u) = A\sqrt{\frac{1-j\cot\alpha}{2(K+\cot\alpha)}}\sqrt{[c(T_1)+c(T_2)]^2 + [s(T_1)+s(T_2)]^2}\, e^{j\left[\arctan\frac{s(T_1)+s(T_2)}{c(T_1)+c(T_2)} - \pi\frac{(u\csc\alpha)^2}{K+\cot\alpha}\right]} \tag{35}$$

where $c(T)$ and $s(T)$ can be expressed as:

$$\begin{aligned} c(T) &= \int_0^T \cos\left(\frac{\pi}{2}z^2\right)dz \\ s(T) &= \int_0^T \sin\left(\frac{\pi}{2}z^2\right)dz \end{aligned} \tag{36}$$

T_1 and T_2 can be expressed as:

$$\begin{aligned} T_2 &= \sqrt{2(K+\cot\alpha)}\left(\frac{T_d}{2} - \frac{u\csc\alpha}{K+\cot\alpha}\right) \\ -T_1 &= \sqrt{2(K+\cot\alpha)}\left(-\frac{T_d}{2} - \frac{u\csc\alpha}{K+\cot\alpha}\right) \end{aligned} \tag{37}$$

Compared with conventional FrFT, the spectral amplitude of the signal in the improved FrFT support interval is equal to FrFT. The difference between them is only in the phase term. But this will not affect our aim to search for the optimal rotation angle. The spectral amplitude of signal can be expressed as:

$$\left|\hat{X}_\alpha(u)\right| = A\sqrt{\frac{1-j\cot\alpha}{2(K+\cot\alpha)}}\sqrt{[c(T_1)+c(T_2)]^2 + [s(T_1)+s(T_2)]^2} \tag{38}$$

Consider the case where the spectrum amplitude is the maximum. Due to the assumption that the initial frequency of the signal is 0 in the above, it will peak at 0 in u-axis. Substituting $u = 0$ into the equation (38) and assume that $T_d\sqrt{K+\cot\alpha} >> 1$. The Fresnel integral in equation (38) have $c(T) = s(T) \approx 0.5$. Then we can obtain the maximum spectral amplitude expression:

$$\left|\hat{X}_\alpha(0)\right| = A\sqrt{\frac{1-j\cot\alpha}{2(K+\cot\alpha)}} \tag{39}$$

When the spectral support interval is $[-\rho_u/2, \rho_u/2]$, taking $u = \pm\rho_u/2$ into equation (38), one can get:

$$\left|\hat{X}_\alpha(\pm\frac{\rho_u}{2})\right| = \frac{A}{2}\sqrt{\frac{1-j\cot\alpha}{2(K+\cot\alpha)}} \tag{40}$$

From equation (40), we can obtain that when the support interval is ρ_u, the amplitude becomes half of the maximum value. Furthermore, the value of the Fresnel integral function fluctuatingly decreases with the increase of $T_d\sqrt{K+\cot\alpha}$. The shape of the spectrum is closer to a rectangle, and energy of the signal is mainly concentrated in the support interval. While outside the support interval, the amplitude of the signal spectrum is very small [15].

We have derived the spectral characteristics of the continuous signal above. But in practice, we are dealing with discrete signals. So it is necessary to consider the spectral characteristics of the discrete signals. The discrete improved FrFT can be implemented using the FFT-based algorithm proposed by Ozaktas [11]. This algorithm needs to normalize the dimensional before calculation. Since the time and frequency domain have different magnitudes. For the convenience of calculation, both of them are converted into a normalized domain. Define the normalization factor $S = \sqrt{T_d / f_s}$, the signal time domain interval $[-T_d/2, T_d/2]$ and frequency domain interval $[-f_s/2, f_s/2]$ is converted to $[-\Delta x/2, \Delta x/2]$, where $\Delta x = \sqrt{T_d f_s}$. In normalized dimensional the sample interval changes to $1/\Delta x$ and the number of samples is N, where $N = \Delta x^2$.

Based on the assumptions of method II in [11]. To satisfy the sampling theorem, the signal needs to be twice interpolated. After interpolating the signal using sinc interpolation algorithm and some algebraic manipulations similar in [11]. The discrete improved FrFT can be written as:

$$\hat{X}_\alpha(\frac{m}{2\Delta x}) = \frac{\sqrt{1-j\cot\alpha}}{2\Delta x} e^{-j\pi\csc\alpha(\frac{m}{2\Delta x})^2} \sum_{n=-N}^{N} e^{j\pi\csc\alpha(\frac{m-n}{2\Delta x})^2} e^{j\pi(\cot\alpha-\csc\alpha)(\frac{n}{2\Delta x})^2} f(\frac{n}{2\Delta x}) \tag{41}$$

where m is the discrete sampling point of FrFT domain, $m \in [-M, M]$. It can be recognized that equation (41) is the result of convolving the signal $f(t)$ with $\exp(j\pi t^2\cot\alpha)$ after modulated by $\exp[j\pi t^2(\cot\alpha-\csc\alpha)]$. The improved FrFT requires only one signal twice interpolation and extraction, one chirp signal multiplication and one signal convolution operation.
Replace $f(n/2\Delta x) = A\exp[j\pi K(n/2\Delta x)^2]$ in equation (41) and set $K = -\cot\alpha_0$:

$$\hat{X}_{\alpha_0}(\frac{m}{2\Delta x}) = \frac{A\sqrt{1-j\cot\alpha_0}}{2\Delta x} \sum_{n=-N}^{N} e^{-j\pi\csc\alpha_0\frac{2mn}{(2\Delta x)^2}} \tag{42}$$

When $m = 0$ in equation (42), we can obtain the maximum amplitude of the discrete signal spectrum:

$$\left|\hat{X}_{\alpha_0}(0)\right| = \frac{|A|\sqrt{1-j\cot\alpha_0}\,|(2N+1)}{2\Delta x} \approx \frac{|A|\sqrt{N}}{\sqrt{\sin\alpha_0}} \tag{43}$$

Denote the energy of the signal $f(t)$ by E and applying the energy conservation property (26):

$$E = \left|\hat{X}_{\alpha_0}(0)\right|^2 = \frac{AN}{\sin\alpha_0} \tag{44}$$

where α_0 is the optimal rotation angle and have $\alpha_0 = \beta + \pi/2$. After normalization, the geometric relationship in Figure 5 changes to:

$$\cos\beta = \frac{\Delta x}{\sqrt{(KT_d S)^2 + (\Delta x)^2}} \tag{45}$$

$$\rho = \frac{\Delta x}{\cos\beta} = \sqrt{T_d f_s + \frac{K^2 T_d^3}{f_s}} \tag{46}$$

$$\rho_u = \left|\rho\cos(\frac{\pi}{2} + |\alpha - \beta|)\right| = \rho\sin(|\alpha - \beta|) \tag{47}$$

In the normalized domain, the sample interval is $1/\Delta x$. And we can get the number of sampling points M of the signal in support interval $M = \rho_u \Delta x$. In the support interval, the signal energy can be approximated as being equally divided by the sampling points M. Hence we can obtain the following:

$$E = \left|\hat{X}_{\alpha}(m)\right|^2 M \tag{48}$$

Combining equation (44) with equation (48), one has:

$$\left|\hat{X}_{\alpha}(m)\right| = \frac{A}{\sqrt{\sin|\alpha - \alpha_0|}} \tag{49}$$

3.3. Optimal Rotation Angle Search Strategy

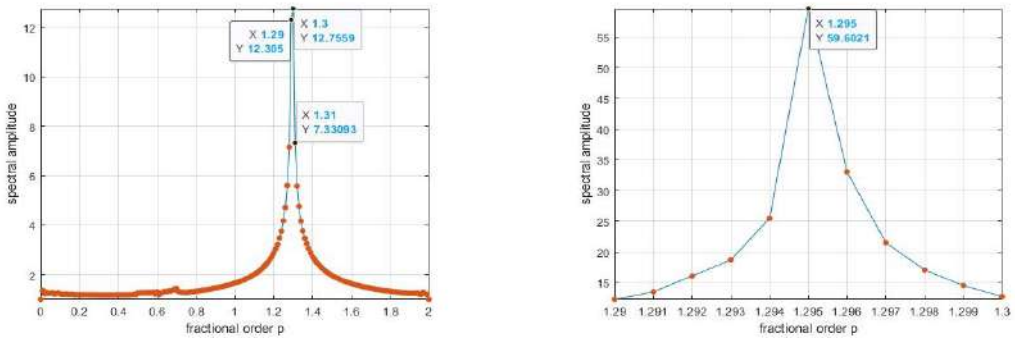
Similarly to FrFT, the improved FrFT also need to scan for all rotation angles and search for peaks in the formed α-u plane. However, this kind of method has two shortcomings. On the one hand, when the parameter estimation accuracy is high, it needs to set a smaller search step size, which leads to a very time-consuming calculation of improved FrFT and peak search in the two-dimensional plane. On the other hand, storing this data in the radar hardware system is very memory consuming. In order to avoid performing improved FrFT for each value of α, we employ an improved search strategy to reduce the computational cost while ensuring the accuracy of signal parameter estimation.

From equation (49), the spectral amplitude of the signal in the FrFT domain is symmetric on both sides of the optimal rotation angle. We can simplify the process using this feature by applying a two-level search. First, perform the improved FrFT with a larger step size. Search the location of the peak and compare the peak with the difference in amplitude between the left and right sides. According to the symmetry property, the optimal rotation angle will appear on the side with a smaller values of difference. Then perform the improved FrFT on this side with smaller step size. Repeat the above steps until the expected accuracy.

4. SIMULATION

Example 1: The process of estimating the LFM signal chirp rate of the improved FrFT algorithm is shown in this example. Consider an LFM signal $\exp(j\pi 2000t^2)$ which is sampled with

parameters f_s = 4000Hz, T_d = 1s. Set the step size of coarse search to 0.01, and the result is shown in Figure 6(a). It is clear from the figure that order 1.3 is the optimal order. Comparing the difference between the 1.3 order and its left and right side orders. We can determine that the optimal rotation order is in (1.29, 1.30). In this scope continue performing the improved FrFT with a smaller step size of 0.001. The results are shown in Figure 6(b).



(a) Coarse search result

(b) Concise search results

Figure 6. The improved FrFT two-level search

The maximum amplitude order in the Figure 6(b) is 1.295. From the relationship between the rotation order p and the chirp rate of the signal $K = -\cot(p\pi/2)f_s/T_d$. We can get the estimation of the chirp rate is 1998.7Hz/s. The estimation error is 1.3Hz/s.

Example 2: To evaluate the improved FrFT performance, we add Gaussian white noise to the signal in *Example 1*. The signal to noise ratio (SNR) varies from −10 to 5dB. For each SNR value, total 1000 trials are performed. Figure 7 shows that the improved FrFT and search strategy still has good estimation performance at -6dB noise condition.

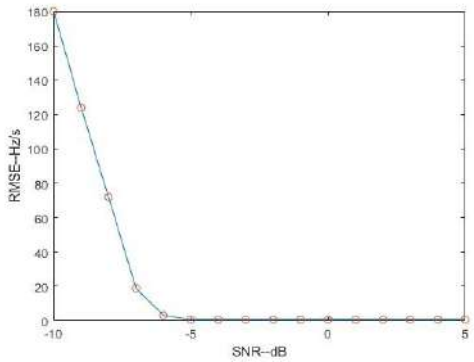


Figure 7. RMSE versus SNR for the chip rate

Example 3: This example is to demonstrate the efficiency of the improved algorithm. The same signal from *Example 1* is used and adds -5dB noise. Compare the improved FrFT algorithm with the conventional FrFT of 0.01, 0.001 and 0.0001 step sizes. The estimated errors and processing times are shown in the Table 1. Simulation results indicate that the improved FrFT can achieve the accuracy of traditional FrFT in 0.0001 step size, and the processing time is 1/3500 of the original.

Table 1. Comparison of the efficiency between improved algorithm and traditional FrFT

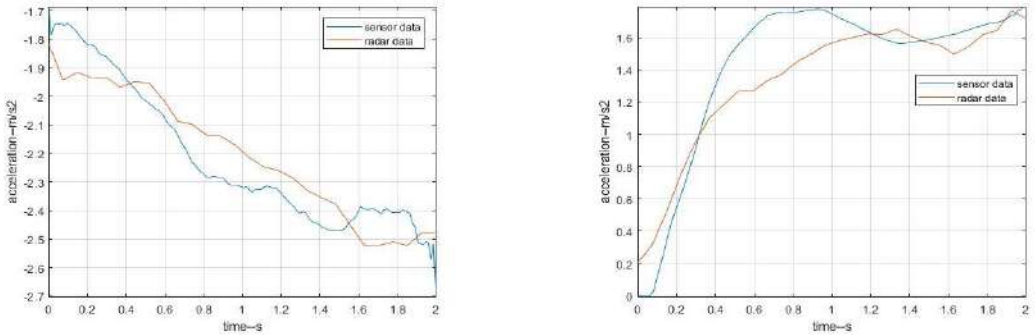
Algorithm	Step Size	Estimated Chirp Rate (Hz/s)	Estimated Error (Hz/s)	Running Time (s)
Conventional FrFT	0.01	2038.6	38.6	0.82
Conventional FrFT	0.001	1991.3	8.7	10.68
Conventional FrFT	0.0001	1998.7	1.3	138.52
Improved FrFT	-	2001.5	1.5	0.04

5. EXPERIMENTAL RESULTS

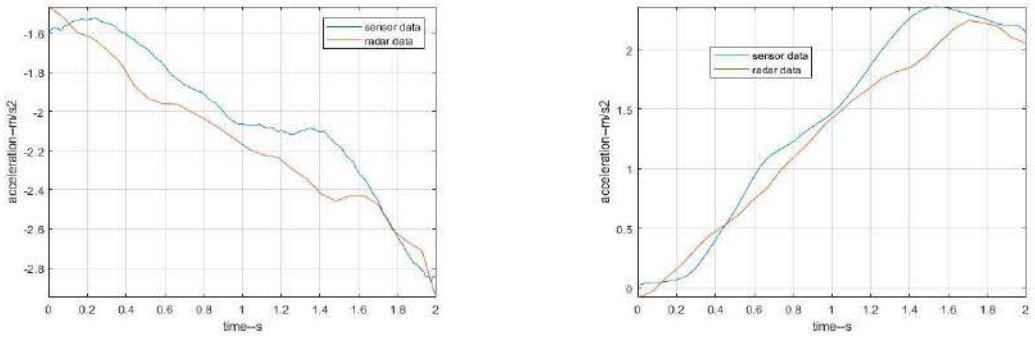
In order to demonstrate the effectiveness of the proposed algorithm in real scenarios. We implemented the algorithm in the TI 2243p cascade radar system and tested it in a reality situation. The test environment is shown in Figure 8.

Figure 8. Acceleration measurement experiment scenario

In the experiment, a car was located in front of the radar. After the experiment started, driving the car made acceleration and deceleration motions while turning on the radar to collect the data. In order to verify the accuracy of the radar data, we placed an acceleration sensor on the car for data comparison while the car was driving. After saving the radar and sensor data, the acceleration change curve of both is drawn by Matlab. The result is shown in Figure 9.



(a) The first set of experimental measurements



(b) The second set of experimental measurements

Figure 9. Actual scenario car acceleration measurement results

The experimental results prove that this paper's algorithm can effectively measure a moving target's acceleration. The estimation accuracy can reach $0.2m/s^2$. In addition, the improved FrFT can greatly reduce the data processing time. The processing time of one frame is about 50us, which meets the real-time requirement of the radar system.

6. CONCLUSIONS

This paper proposes a method to measure the acceleration of moving targets for FMCW radar. This method first addresses the problem of range migration in FMCW radar system. Then a more concise and efficient improved FrFT is performed to obtain the chirp rate of the signal and accordingly get the acceleration estimate. Simulation with Matlab and actual scenario tests are presented to validate the proposed method.

ACKNOWLEDGEMENTS

This work was supported in part by Special Innovation Project for National Defense 19-163-11-ZT-002-002-02.

REFERENCES

[1] Li, X & Wang, X & Yang, Q & Fu, S, (2021) "Signal processing for TDM MIMO FMCW millimeter-wave radar sensors", IEEE Access, Vol. 9, pp167959-167971.
[2] Hassanien, A & Vorobyov, S. A & Gershman, A. B, (2012) "Moving target parameters estimation in noncoherent MIMO radar systems", IEEE Transactions on Signal Processing, Vol. 60, No. 5, pp2354-2361.
[3] Portnoff, M. (1980) "Time-frequency representation of digital signals and systems based on short-time Fourier analysis", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 28, No. 1, pp55-69.
[4] Claasen, T. & Mecklenbraeuker, W. (1980) "The wigner distribution - a tool for time-frequency signal analysis. i: continuous-time signals", Philips Journal of Research, Vol. 35, No. 14.
[5] Gonalves, P. & Baraniuk, R. G. (1998) "Pseudo affine wigner distributions : definition and kernel formulation", IEEE Transactions on Signal Processing, Vol. 46, No. 6, pp1505-1516.
[6] Stankovic, L. (1994) "A method for time-frequency analysis", Signal Processing IEEE Transactions on, Vol. 42, No. 1, pp225-229.
[7] Wood, J. C. & Barry, D. T. (1992) "Radon transformation of time-frequency distributions for analysis of multicomponent signals" Signal Processing IEEE Transactions on, Vol. 42, No. 11, pp3166-3177.
[8] Barbarossa, S. (1995) "Analysis of multicomponent lfm signals by a combined wigner-hough transform", IEEE Transactions on Signal Processing, Vol. 43, No. 6, pp1511-1515.

[9] Lv, X & Bi, G & Wan, C & Xing, M. (2011) "Lv's distribution: principle, implementation, properties, and performance", IEEE Transactions on Signal Processing, Vol. 59, No. 8, pp3576-3591.
[10] Almeida, L. B. (1994) "The fractional Fourier transform and time-frequency representations", IEEE Transactions on signal processing, Vol. 42, No. 11, pp3084-3091.
[11] Ozaktas, H. M. & Arikan, O. & Kutay, M. A. & Bozdagt, G. (1996) "Digital computation of the fractional Fourier transform". IEEE Transactions on signal processing, Vol. 44, No. 9, pp2141-2150.
[12] Pei, S. C. & Ding, J. J. (2000) "Closed-form discrete fractional and affine Fourier transforms", IEEE transactions on signal processing, Vol. 48, No. 5, pp1338-1353.
[13] Richards, M. A. (2014) "The keystone transformation for correcting range migration in range-doppler processing", pulse, 1000, 1.
[14] Guo, Y & Yang, L (2019) "Method for parameter estimation of LFM signal and its application", IET Signal Processing, Vol. 13, No. 5, pp538-543.
[15] Xu, H. F & Liu, F. (2010) "Spectrum characteristic analysis of linear frequency-modulated signal in the fractional Fourier domain", Signal Process, Vol. 26, No. 12, pp1896-1901.

AUTHORS

Qingbo Wang was born in Jiangxi, China, in 1997. He is currently pursuing the Master's degree in Nanjing University of Aeronautics and Astronautics, Nanjing, China. Currently, the major research focus is on signal processing for mmWave radar.

AN INTELLIGENT AND DATA-BASED SKATE ANALYZER TO ASSIST IN ANALYZING MOVEMENTS OF SKATE ON ICE

Yirina Wang[1], Yu Sun[2]

[1]Santa Margarita Catholic High School, 22062 Antonio Pkwy,
Rancho Santa Margarita, CA 92688
[2]Computer Science Department, California State Polytechnic University,
Pomona, CA 91768

ABSTRACT

Ever since the start of Figure Skating, there has been an emphasis on skating technique, especially in the step sequences of a skater's choreography [1]. But Figure Skaters often are not able to detect the motion, edge, or placement of their blade on the ice without watching themselves skate. The solution to this problem would be to have a skate analyzer. A skate analyzer would record the movements of a skate on ice and one would be able to playback the recorded data and view their skate motion precisely [2]. Three main components that my project links together are the QTPY-ESP 32 microcontroller, the sensor that combines the accelerometer, gyroscope, and magnetometer, and the SD card reader. The QTPY-ESP32 is a microcontroller that acts as a main computer controlling the whole board. The QTPY is then connected to a sensor board through an I2c protocol. Then, through an SPI protocol, the QTPY is connected to an SD card reader. After the skater is finished recording, they can insert the SD card in a computer, upload the data into the app, and play it back. There is also a slider on the top of the screen that the skater can slide back and forth to view the skate at specific times in the file. This would be a great technology to use for skaters as they can playback their movements on ice and improve their technique [3].

KEYWORDS

Figure skate, Ice, Analyzer

1. INTRODUCTION

Ever since the start of Figure Skating, there has been an emphasis on skating technique, especially in the step sequences of a skater's choreography. Because of this emphasis, skaters are more attentive to the movements of their skate on ice. But Figure Skaters often are not able to detect the motion, edge, or placement of their blade on the ice without watching themselves skate. Especially when they are practicing by themselves. They would need to record themselves or have someone supervise them while they are practicing in order to productively get better in their technique.

The solution to this problem would be to have a skate analyzer. A skate analyzer would record the movements of a skate on ice and one would be able to playback the recorded data and view their skate motion precisely. There would be hardware with multiple sensors to record data based on the movements of the skate and then the data would be processed. After that, a skater can upload this data onto an app that would have a 3D model of a skate on ice, which would move

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 191-199, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130516

based on the data [4]. Then, the skater would be able to see whether the skate moved the wrong way, fell on the wrong edge, or did other wrong techniques.

The first blindspot that I wanted to test out was how to position the board on the shoe, whether it affected the accuracy of the data, and if the material used to secure the board on the shoe mattered. The second blind spot that I wanted to test out was how the gravity would affect the board in different directions. For both of theseblindspots, we set up experiments with different combinations of placements of the board, materials used to secure the board, and the different directions of gravity. Then, we would record data using each situation. Then, using the wide range of data collected, we would upload them and see which provided the best results.

The three methodology comparisons were all trying to test out their wearable device. The first experiment developed a prototype jump monitor for figure skaters [5]. They wanted to measure jump count, jump height, and rotation speed by using an IMU attached to the lower back [6]. These jump monitors could help skaters and coaches balance injury and performance concerns. The second experiment sought to improve on a previously developed wearable device named the IceSence device attached to the lower back. This study wanted to determine the force data for off-ice jump landings. This study demonstrates that their technology may be able to offer information that is helpful in upcoming figure skating studies. The third study put forth a new method for automatic identification of ice hockey skating stires as well as a technique to detect ice contact and swing phases of individual stries. After analyzing the data using a 3D accelerometer attached to the hockey skate, they proved that this technology is precise, user friendly, and efficient.

2. CHALLENGES

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

2.1. How to Place and Connect all the Components on a Board

One problem that we can experience is how to place and connect all the components on a board small enough to fit on the skater's shoe [14]. The components might not fit, or the soldering might cause problems, and not connect correctly. To solve this problem, we could make sure that all the components are bought small enough by measuring how big of a board can fit on the skater’s shoe. We could also possibly consider what material we would like the board to be, as the flexibility of the board would also contribute to solving the problem. For a secure connection, we could plan out how we will solder, and if soldering does not work in particular times, we could use cables to connect them.

2.2. Making Sure the Data is Correctly Processed and Presented

Another problem that we could experience is making sure the data is correctly processed, presented, and that the skate will be moving accordingly. We can program the data to be stored in charts and categorize them in a way that is easy for us to use. Additionally, we can code specific functions to calculate the velocity, position, and acceleration of the skate based on the data, and present it on the screen accurately. Then, we can also add specific variables in Unity, such as the position, velocity, and acceleration, so that the user can track these data instantaneously while watching the playback of their skate movements [15].

2.3. Deciding how the Board Should be Placed

Lastly, a problem that we can experience is deciding how the board should be placed on the skater's shoe so that it fits and is influenced by gravity properly. A way we can solve this is first find a way to lock the hardware board on the shoe. Either we can tuck it in the sides of the shoe, or use a velcro and tie it to the laces of the shoe. After that, we can experiment with the direction of the placement on the shoe, either horizontally or vertically, and see the resulting data. With the different outcomes, we can code to see how gravity effectively acts on the shoe so the data can come out correctly.

3. SOLUTION

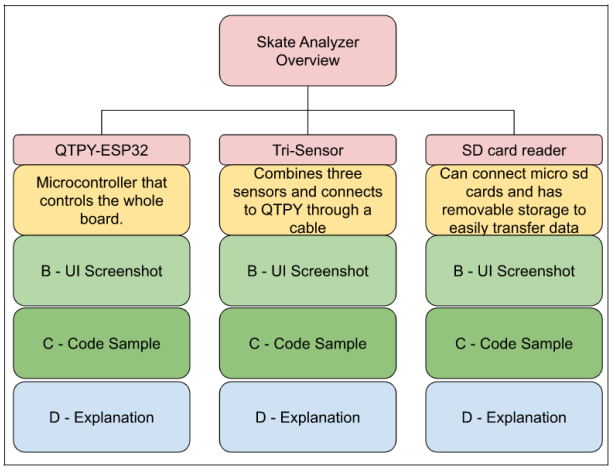


Figure 1. Overview of the solution

Three main components that my project links together are the QTPY-ESP 32 microcontroller, the sensor that combines the accelerometer, gyroscope, and magnetometer, and the SD card reader [7]. The QTPY-ESP32 is a microcontroller that acts as a main computer controlling the whole board. The QTPY is then connected to a sensor board through an I2c protocol. Then, through an SPI protocol, the QTPY is connected to an SD card reader [8]. When the skater is about to record data, he or she can position the board on their skate and turn the microcontroller on and the whole board will start to record data. The whole board will be powered through a battery that is connected with a battery controller. Then, as the skater moves, the sensors will detect the movements and all that motion data will be stored in the SD card inserted into the reader. After the skater is finished recording, they can insert the SD card in a computer, upload the data into the app, and play it back. There is also a slider on the top of the screen that the skater can slide back and forth to view the skate at specific times in the file. At the bottom of the screen, there will be the position, velocity, and acceleration data of the skate while it is moving. Then, the skater can also control the view, by pressing W, A, S, D, to represent up, left, down and right, respectively. This allows the skater to follow the skate as it moves in different directions.

One of the components that we used is the QTPY-ESP 32 microcontroller. This acts as a computer that controls the whole board. Turning this on allows the whole board to work. This component could work with WiFi, but in our project we chose not to incorporate this feature. It has a USB type c charger that can be used to charge the battery.

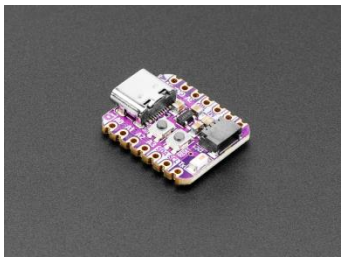


Figure 2. The microcontroller

```
def getSensorDataAsCSV(startime):
    deltaTime = (time.monotonic_ns() - startime) / 1000000000
    acc = acc_gyro_sensor.acceleration
    gyro = acc_gyro_sensor.gyro
    mag = magnetometer.magnetic

    line = str(deltaTime) + ','
    line +=  str(acc[0]) + ',' +  str(acc[1]) + ',' +  str(acc[2]) + ','
    line +=  str(gyro[0]) + ',' +  str(gyro[1]) + ',' +  str(gyro[2]) + ','
    line +=  str(mag[0]) + ',' +  str(mag[1]) + ',' +  str(mag[2])

    return line, [deltaTime, acc[0],  acc[1],  acc[2], gyro[0],  gyro[1],  gyro[2], mag[0],  mag[1],  mag[2]]

def getSensorDataAsJSON():
    return {
        'acc': acc_gyro_sensor.acceleration,
        'gyro': acc_gyro_sensor.gyro,
        'mag': magnetometer.magnetic
    }
```

Figure 3. Screenshot of code 1

```
def printAccel(data):
    print(data[1],data[2],data[3], sep='\t')


with open(filepath, "w+") as f:
    print('creating', filepath)
    f.write('second,acc_x,acc_y,acc_z,gyro_x,gyro_y,gy
    f.flush()
    startTime = time.monotonic_ns()
    acc_gyro_sensor.accelerometer_range = 1
    while True:
        #data = str(getSensorWithTime(startTime))
        csv, data = getSensorDataAsCSV(startTime)
        printAccel(data)
        print(acc_gyro_sensor.accelerometer_range)
        f.write(csv +'\n')
        f.flush()
        time.sleep(0.1)
```

Figure 4. Screenshot of code 2

The first code sample gets the timestamp from the microcontroller. Then, we save the accelerometer data into the variable acc, gyroscope data into the variable gyro, and magnetometer into the variable mag [9]. After this, we get the data from the SD card. We turn them into a single string to use later. Then, for the second sample, we create a file to store the data into. We add titles for the column, such as seconds, acceleration, gyroscope, and magnetometer for x, y and z axis. Then we have a while loop and for every 0.1 second we add a new column with new data.

Another component that I used is the sensor. The sensor conveniently combines the accelerometer, gyroscope, and magnetometer into one board. The accelerometer measures the acceleration forces (including gravity), the gyroscope senses the direction and speed (in degrees

per second) of rotation, and the magnetometer points to the strongest magnet force (usually the magnetic North Pole), respectively. It is connected with the QTPY through an I2c protocol.

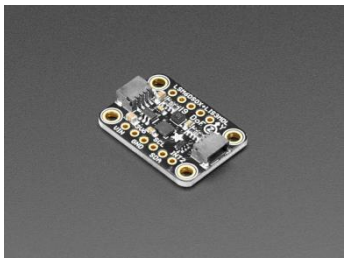


Figure 5. The sensor

The third component is an SD card reader [10]. This SD card allows us to insert or connect micro sd cards so we can save the data on the device based on the movements of the skate. This SD card reader also has removable storage so we can easily transfer the data to a computer.

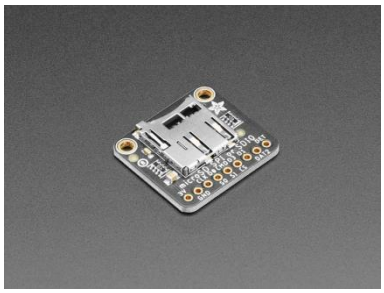


Figure 6. The SD card reader

4. EXPERIMENT

4.1. Experiment 1

One possible blind spot that I want to test out is the accuracy in the detection of the motion by the sensors. For example, sometimes the sensors would not the placed correctly, and therefore would not be giving us clear data.

We could set up an experiment where we would try different placements of the board and see the outcomes of the data. For example, the board can be put horizontally, vertically, or diagonally, based on the shoe. There could also be angles, such as slanted and put on the tongue of the shoe, or upright tucked into the edge. Different angles and positions might result in different data outcomes. Additionally, there can be different ways to secure the board onto the shoe. This can also affect the data. Therefore, we need to experiment with these two factors and find the best combination for accuracy of the data.

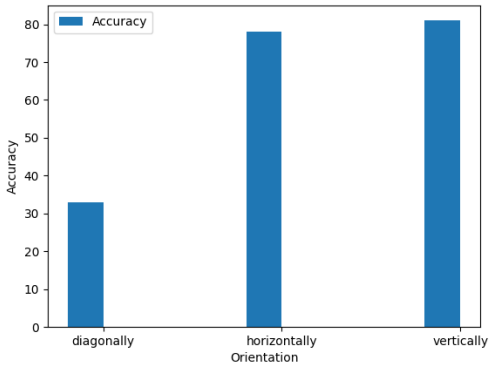


Figure 7. Graph of experiment 1

Here we can see that the orientation of the board doesn't have much difference when placed horizontally or vertically but there is a big drop in accuracy when placed diagonally. I believe this is because placing the board diagonally consistently at the exact same angle is very difficult. Place the board on board diagonally. Additionally, the orientation of the board may also affect the stability of the board, which can impact accuracy. When the board is placed horizontally or vertically, it is more likely to remain stable and not wobble or shift during use. However, when the board is placed diagonally, it may be more prone to wobbling or shifting. This is due to the limited ways I can attach the board to the skate.

4.2. Experiment 2

Another potential blindspot is the way gravity acts on the board and affects the data. Although there are ways listed before that could be used to solve this problem, sometimes just a bit of angle difference can result in the gravity not acting on the board correctly.
First, we need to solve the first blind spot of how to place the board and what to use to secure it. After deciding that, we need to experiment with gravity. There could be different directions in which the gravity acts on the board. However, we can also play with different combinations of placements of the board and directions of gravity acting on the board. With different placements come different programming of the direction of the gravity. Based on the outcome of that experiment, we can find the best combination that would give the most precise data so the skate can move accurately.

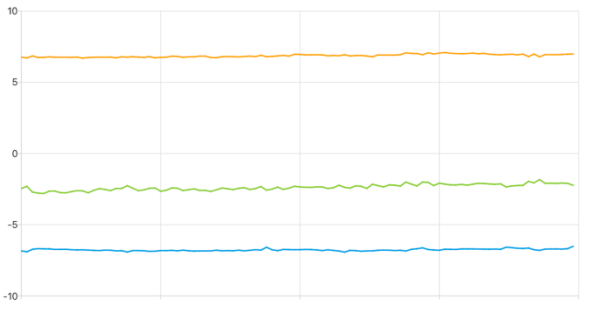


Figure 8. Diagonal Diagram

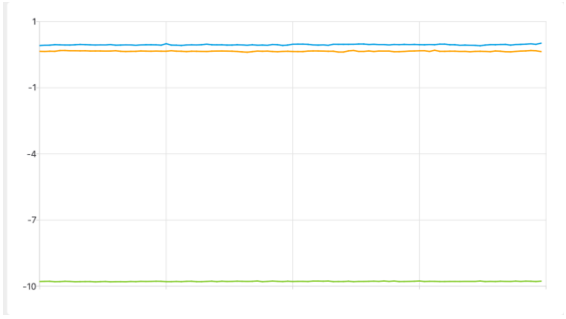


Figure 9. Vertical Diagram

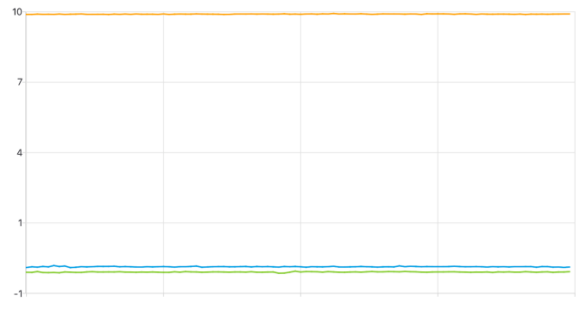


Figure 10. Horizontal Diagram

Orange: Z-Axis
Blue: X-Axis
Green: Y-Axis

The data shows that mounting the board diagonally causes the most noise. This is because of how unstable the board is mounted on the skate. Having the three axes be non-zero on the diagonal test makes canceling out the effects of gravity difficult. This also makes determining the orientation of the skate harder as well. The axis closest to zero between the horizontal and vertical orientation is the X-axis in the vertical. The Y-Axis on the vertical orientation is also fairly consistent at -9.8. This makes canceling the effects of gravity much easier as we only have to worry about gravity on one axis. This will also make finding the orientation of the board easier as well. Since the vertical mounting of the board give the most stable and simplest data to manipulate while easy to mount, so we designed our mount and code to correspond the a vertically mounted board.

5. RELATED WORK

Dustin A Bruening's study developed a prototype jump monitor for figure skating [11]. He wanted to determine whether he could use an inertial measurement unit, or IMU, to measure jump count, jump height, and rotation speed. Skater's wore an IMU attached to the lower back and were filmed with a camera for validation in the jump height. Overall, his study's findings suggest that a single waist-mounted IMU can accurately identify multi-revolution jumps and measure rotation speeds. The algorithm's leap height estimation accuracy should be improved. These features in a fully integrated jump monitor could help skaters and coaches balance injury and performance concerns.

By carefully measuring the power of on-ice landings, Sarah Ridge's research aims to further knowledge of injuries in figure skating and have an impact on training methods, injury preventive measures, and rehabilitation procedures [12]. The study makes use of a wearable IceSense device attached to a skater's shoe. This study proposes that the most accurate force data for off-ice jump landings can be obtained by calibrating the strain sensors against a load cell situated above the skate blade using a controlled drop method. Data from a single jump landing on the ice demonstrate that this technology may be able to offer information that is helpful in upcoming figure skating studies.

Stetter's research demonstrates a new method for automatic identification of ice hockey skating strides as well as a technique to detect ice contact and swing phases of individual strides by quantifying vibrations in 3D acceleration data during the blade-ice interaction [13]. The data was then analyzed using a 3D accelerometer attached to a hockey skate. To test the new method's precision on a variety of forward stride patterns for temporal skating competitions, the predicted contact times and stride times for a series of five consecutive strides were validated. These findings demonstrate the validity of the novel method for determining strides, ice contact, and swing phases during ice hockey skating. This technology, which allows for in-field ice hockey testing, is precise, user-friendly, and efficient.

6. CONCLUSIONS

Some limitations to my project might be the fact that the ice size displayed on the screen where the skate can move is a bit small. Sometimes if the skater is skating in a larger ice rink while having the device on and tracking their movements, after they upload their data onto the app, the skate might fall out of the ice displayed on the screen. So if I had time, in the future I would make this change. Additionally, I would like to possibly add another skate so then both shoes can be tracked at the same time. For example, sometimes one foot would constantly be off the ice while practicing, and if the device is attached to the shoe in the air, then the results would not be helpful. But if both of the shoes appear on the screen, then the skater can visualize their movements of both their shoes better.

This project aims to assist skaters to better visualize their movements on ice by recording the motion of their shoe on ice using a device board that incorporates a microcontroller, sensors, sdcard reader, and a battery controller. They can upload their data onto the app and closely view their movements and make changes to their techniques on ice.

REFERENCES

[1] V. V. Gudkov, A. V. Solovyev, and A. V. Knyazkov. (2019). Development of Skate Analyzer for Figure Skaters. In 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) (pp. 1980-1985). IEEE. https://doi.org/10.1109/EIConRus.2019.8656719

[2] W. E. Garrett and D. A. Viano. (2007). Biomechanics of figure skating: A review. Journal of Sports Science and Medicine, 6, 1-10. https://www.jssm.org/hf.php?id=jssm-06-01.xml

[3] A. E. Carlson, R. L. Frederick, and S. J. Vanderburgh. (2014). Enhancing Figure Skating Performance: Theoretical Frameworks and Empirical Findings. Journal of Dance Medicine & Science, 18(1), 30-39. https://doi.org/10.12678/1089-313X.18.1.30

[4] Rusinkiewicz, Szymon, Olaf Hall-Holt, and Marc Levoy. "Real-time 3D model acquisition." ACM Transactions on Graphics (TOG) 21.3 (2002): 438-446.

[5] Salles, A. S., da Costa, J. P., de Oliveira, D. F., & Costa, F. B. (2021). A wearable prototype for monitoring figure skaters' jumps. Journal of Instrumentation, 16(10), C10004. https://doi.org/10.1088/1748-0221/16/10/C10004

[6] Vanderburgh, P. M., Lee, S. M., & Deshpande, S. (2018). Measurement of ground reaction forces during off-ice jumps in figure skaters using a wearable device. Sports Engineering, 21(4), 377-388. https://doi.org/10.1007/s12283-018-0277-4

[7] Bolanakis, Dimosthenis E. "A survey of research in microcontroller education." IEEE RevistaIberoamericana de Tecnologias del Aprendizaje 14.2 (2019): 50-57.

[8] Leens, Frédéric. "An introduction to I 2 C and SPI protocols." IEEE Instrumentation & Measurement Magazine 12.1 (2009): 8-13.

[9] Gallego, Juan A., et al. "Real-time estimation of pathological tremor parameters from gyroscope data." Sensors 10.3 (2010): 2129-2149.

[10] Yang, Yansi, et al. "Hardware system design of SD card reader and image processor on FPGA." 2011 IEEE International Conference on Information and Automation. IEEE, 2011.

[11] Bruening, D. A., Reynolds, R. E., Adair, C. W., Zapalo, P., & Ridge, S. T. (2018). A sport-specific wearable jump monitor for figure skating. PLoS ONE, 13(11), e0206162. https://doi.org/10.1371/journal.pone.0206162

[12] Ridge, S., Bruening, D., Charles, S., Stahl, C., Smith, D., Reynolds, R., Adamo, B., Harper, B., Adair, C., & Manwaring, P. (2020, December 10). IceSense Proof Of Concept: Calibrating An Instrumented Figure Skating Blade To Measure On-Ice Forces. MDPI. https://www.mdpi.com/1424-8220/20/24/7082

[13] Vats, Kanav, et al. "Player tracking and identification in ice hockey." Expert Systems with Applications 213 (2023): 119250.

[14] Nadar, N., Tinku, R., Pandey, S. K., & Singh, N. (2019). Design and development of a wearable device for monitoring and analyzing motion data of figure skaters. In 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 328-333). IEEE. https://doi.org/10.1109/SPIN.2019.8711677

[15] Liang, Z., Lu, Y., Li, H., Li, C., Zhang, Y., & Li, Y. (2020). A study of smart sports equipment based on Unity3D and Internet of Things technology. In 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 96-100). IEEE. https://doi.org/10.1109/ITNEC49270.2020.9197405

A FULLY AUTOMATED MUSIC EQUALIZER BASED ON MUSIC GENRE DETECTION USING DEEP LEARNING AND NEURAL NETWORK

Kevin Hu1, Yu Sun2, Yujia Zhang3

1Sage Hill School, 20402 Newport Coast Dr, Newport Beach, CA 92657
2California State Polytechnic University, Pomona, CA, 91768,
Irvine, CA 92620
3University of California Irvine, Irvine, CA 92697

ABSTRACT

Recent years have witnessed the dramatic popularity of online music streaming and the use of headphones like AirPods, which millions of people use daily [1]. Melodic EQ was inspired by these users to create the best audio listening experience for listeners with various preferences [2]. Melodic EQ is a project that creates custom EQs to the user's custom music tastes and filters the audio to fit their favorite settings. To achieve this goal, the process starts with a song file taken from an existing file, for example, Spotify downloads or mp3s. This file is then uploaded to the app. The software sorts the song in a genre detecting Algorithm and assigns a genre label to that song. Inside the app, the user will create or select EQs for that genre and apply it to their music. The interface is easy to use and the app aims to make everyone's preferences achievable and on the fly. That's why there are presets for each category for users who are unfamiliar with equalizers, and custom settings for advanced users to create their perfect sound for each genre.

KEYWORDS

AI auto genre detection, Automatic genre switching, EQ, Convolution music equalizer network

1. INTRODUCTION

Audio has been an integral part of human society for millennia [3]. Ranging from conversations to music, it is integral to humans as group animals to communicate and express their feelings through music [4]. Now more than ever, people have access to all types of audio material. Recently there has been a meteoric rise of personal audio devices, headphone use, and entertainment viewing, phones, headphones, and other audio broadcasting devices are in every corner of society. This trend is also increasingly encouraged by corporations through YouTube, streaming services, and music streaming [5]. With so many different individuals around the world, everyone has their favorite listening experience in their corner of the audible frequency spectrum. EQs are the perfect solution to that problem, Melodic EQ steps this solution up by adding modern technology in neural networks and personalized user inputs to give every individual their own personalized best audio listening experience. I made Melodic EQ under the

mentorship of Jonathan Thamrun, Product Support Associate at Nodus Technologies, and Yu Sun, Associate Professor of Computer Science at Cal Poly Pomona.

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 201-209, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130517

Some techniques and systems that allow users to control EQs intuitively include built-in hardware settings for headphones, background software, and manual adjustment of EQ inside individual apps. However, these proposals do not possess genre recognition or switching to individual EQs for each setting [6]. Their implementations are also limited in scale, for example, limiting to only pairing with specific apps, scenarios, or static settings. An example of hardware settings is the Sony app. They are limited to only a static EQ pairing with the headphones the user has, and cannot be transferred to another pair. The method is simply a standard EQ and doesn't have any AI inputs. Then we move on to more advanced software like Nahmic 3, the method/algorithm used can already identify between scenarios like video calls and listening to music, however, the results often cannot satisfy the need for switching EQs fitting multiple scenarios in a short period like a YouTube video or movie scene that has music and dialogue crossed over constantly. Melodic EQ is more focused on identifying genre and then switching up the EQ. Some EQs for universal apps are also very specific to target quality speakers and headphones which might lead to buzzing. With Melodic EQ, we allow users to customize settings for each device they play their music on, which makes life a lot more simple as to not overdrive cheap quality speakers that cannot fit into the one-size fits all product.

In this paper, I followed the same line of research by Automatic Music Genre Classification Based on CRNN [7]. My goal is to integrate an AI algorithm working with equalizer to sort music, and I was inspired by their research to use certain features of their research. First, they used the GTZAN dataset with a CRNN in Python to make a neural network to predict the music genre. I was able to find the same exact dataset on kaggle to also use this dataset to create my own algorithm. Secondly, they made the data readable as an image using spectrograms, or images of the soundwaves. I too needed to use spectrograms and had to use the PyDub library with the Librosa library in python to edit the audio, get channel lengths, and spectrograms. Lastly, they used a CRNN as their main priority but we used a Feedforward Neural Network or FNN, which they compared to. They were able to get a similar accuracy using a STFN or Short-time Fourier transform, a similar feedforward network.

To test the working of the app along with the backend, we used a combination of techniques that increases the accuracy of the model. First, to prove the results of my neural network, I visualized the train-test split, where 80 percent of the GTZAN dataset was used to train the feedforward neural network and 20 percent of the dataset was used to test the accuracy of the algorithm. I visualized the dataset by using pandas to graph the accuracy of the neural network. Next I also used the keras built in model.summary() to check the accuracy of the algorithm which gives a summary of all the components and the accuracy of both the training and the testing of the dataset. Finally, in the actual app, I was able to import different genres of songs from hip-hop, country, and pop to test the working of the app working with the algorithm.

The rest of the paper is organized as follows: section 2 gives the details on the challenges that we met during the experiment and how those challenges influenced the designing of the app. Section 3 focuses on the details of the methodology of the app, including the backend and frontends corresponding to the challenges and goals that we mentioned in the previous sections. Section 4 presents the relevant details about the experiment we did, the various methods of testing and evaluation, and graphs and figures of the data we collected. Section 5 shows related works that were an inspiration and parallel studies based of the project. 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. License to Access Music

My biggest challenge with this app was the ability to access music because we do not have the license to access music through streaming services who have dedicated servers to store millions of audio files [8]. This is the biggest limitation to the app because it makes it so much harder to work with all forms of audio being processed by the user's phone. Unfortunately, this means the app cannot select any song off of an online platform and just plug Melodic EQ's algorithm onto it. The user can only play audio from a file picker for downloaded audio files which severely limited the usability of the app. However, if a user paid for a streaming service and downloaded mp3s, the user could go to the folder of the app and stream music directly from their downloads and we could only rely on this method being an independent 3rd party option. There is no queue for the app in beta either so it is just a gimmick without better implementation for more audio modifications and playing.

2.2. Using A New Coding Language in Android Studio

My second biggest challenge was using a new coding language in Android Studio by using Flutter [9]. Dart is the official Flutter language and I have never used it before in any scenario. Dart is very similar to Java but I had little to no experience in it so I had to start everything from the ground up. Fortunately, my mentor Jonothan was very adept in it and was able to help me code. The hardest integral part of the coding was inheritance, which involved multiple classes with object oriented code spanning the entire project. All of the variables, and instances got very messy very quickly and I had to spend hours fixing the code. What made this process even more challenging was the audio processing using Flutter, which involved multiple independent packages made by other programs to develop the logic and the backend. These are not able to be explained by the normal methods of coding in Dart, and rely on documentation and custom objects and functions. I had to follow my mentor Jonathan very closely so as to not get lost, and successfully played audio with equalizer filters when coding the app.

2.3. Never-Ending Grind to Perfect the Algorithm

My last challenge was the never-ending grind to perfect the algorithm and sort the correct genre. The music genre detecting algorithm is the most vital part of the audio sorting process, so the genre detection needs to acquire the highest test accuracy possible. To make this happen I had to adjust and learn multiple ways to sort the preprocessing functions, and the parameters of the neural network to get the best results. This meant expanding the libraries by many folds and ultimately finding the best ones through repeated testing. However, there was also the problem of overfitting or underfitting data [10]. Underfitting means that the model makes accurate, but initially incorrect predictions where both train error and val/test error is large. Overfitting means that the model makes false predictions because train error is very small but val/test error is large. After adjusting the preprocessing, dropout rate, and epoch number I was able to get an optimal fitting number.

3. SOLUTION

Melodic EQ is a custom equalizer that uses an AI algorithm and user input to filter all types of audio. The process starts with an independent algorithm that sorts the music into genres without any inputs, the genre detected is then sent to the algorithm. The algorithm is a feedforward neural network which is a fully connected deep learning network with multiple units. A unit in layer n receives input from all units in layer n-1, and sends output to all units in layer n+1, so there are no loops in the hidden layers (fig 2). Finally, the app checks if there is a pre-existing custom equalizer for that genre and applies a custom or preset equalizer onto the music when it is played (fig 1).

For the neural network, the experiment used Python to create and train a neural network on Google Collab. For the training of the neural network, the dataset was taken from Kaggle to find open-source prelabeled data for music genre classification. The "gtzan-dataset-music-genre-classification," was chosen because of its labeled data from 10 different genres and fit the criteria for most genres of music. The experiment used this dataset with a feedforward neural network by using the python library Pydub, Librosa, and tensorflow [15]. Using the PyDub with the Librosa library python can get certain information about the file. Librosa helps with this method by extracting the log cepstrum or Mel-Frequency Cepstrum Coefficients (MFCC) as input. Log cepstrum is the logarithm operation after the Fourier transform of the signal, and then perform the inverse Fourier transform to obtain the spectrogram. The feedforward neural network was fed the MFCC data along with hyperparameters to get the best results. Feedforward Neural Networks are fully connected, and use dense layers, which are a classic fully connected neural network layer where each input node is connected to each output node. These layers also have a dropout attribute attached to them so that when the layer is used, the activations are set to zero for some random nodes and prevent overfitting (fig 6). This tensorflow model was then run for 200 epochs with a batch size of 32 and was able to have a test accuracy of 94.5% and a valid accuracy of 66% (fig 3). Very similar to the research paper we were inspired by and also outperformed the CRNN in that paper. This wraps up the backend calculations and moves along to the front end. For the frontend, the app was developed on Android Studio using Flutter with Dart as the programming language. I created a GUI for the user to interact with the app which needed to choose the song to equalize, adjust the presets, and adjust settings of the actual app, which makes 3 different pages: Home, Equalizer, and Settings (fig 4). Everything depends on the filters in the Equalizer tab, the equalizer tab has 10 presets for the different types of genres the algorithm sorts into and is a one-size fits all solution for non advanced users who just want a better listening experience. Advanced users are also not left out and can use this page to create their own presets for each of the 10 genres if they find something that suits them more. These equalizer settings work with the home page where the users input their music through a file picking process of the downloaded music on their device. We integrated the neural network into a web service scheme with a Python Flask framework which listens to the web request from the app frontend and sends the music file back to the users. We use an AWS server to host the whole service to make the api call stable.

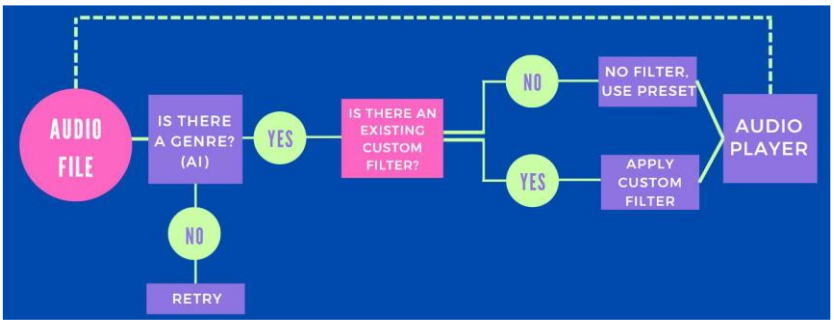


Figure 1. Overview of the solution

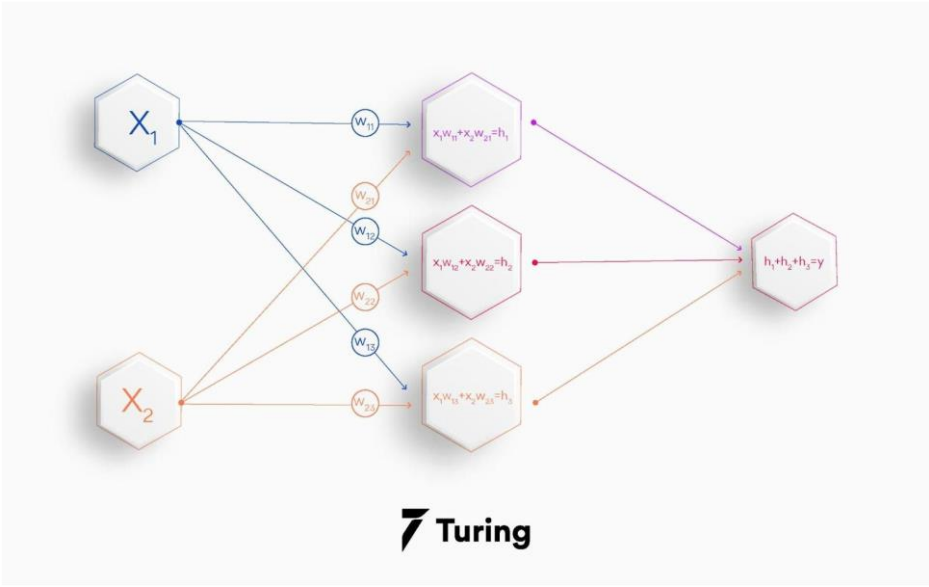


Figure 2. Hidden layers

```
from tensorflow.keras.callbacks import ModelCheckpoint
from datetime import datetime

num_epochs = 200
num_batch_size = 32

checkpointer = ModelCheckpoint(filepath=f'saved_models/audio_classification_{current_time}.h5f5',
                               verbose=1, save_best_only=True)

start = datetime.now()

history = model.fit(X_train, y_train, batch_size=num_batch_size, epochs=num_epochs, validation_data=(X_test, y_test), callbacks=[checkpointer], verbose=1)

duration = datetime.now() - start
print('Training complete in time: ', duration)

25/25 [==============================] - 0s 9ms/step - loss: 0.2196 - accuracy: 0.9387 - val_loss: 2.8662 - val_accuracy: 0.6700
Epoch 187/200
22/25 [=======================>....] - ETA: 0s - loss: 0.1871 - accuracy: 0.9503
Epoch 187: val_loss did not improve from 1.34838
25/25 [==============================] - 0s 9ms/step - loss: 0.1839 - accuracy: 0.9499 - val_loss: 2.8161 - val_accuracy: 0.6350
Epoch 188/200
22/25 [=======================>....] - ETA: 0s - loss: 0.1695 - accuracy: 0.9474
Epoch 188: val_loss did not improve from 1.34838
25/25 [==============================] - 0s 9ms/step - loss: 0.1820 - accuracy: 0.9424 - val_loss: 3.1807 - val_accuracy: 0.6700
Epoch 189/200
24/25 [===========================>..] - ETA: 0s - loss: 0.1869 - accuracy: 0.9453
Epoch 189: val_loss did not improve from 1.34838
25/25 [==============================] - 0s 9ms/step - loss: 0.1848 - accuracy: 0.9462 - val_loss: 3.1572 - val_accuracy: 0.6550
```

Figure 3. Screenshot of code

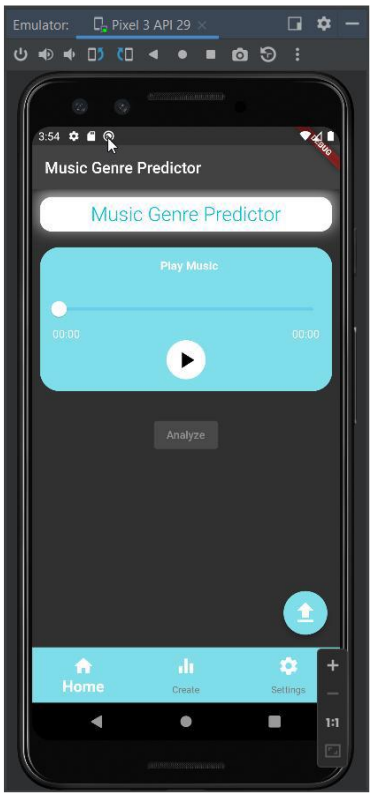


Figure 4. Screenshot of main page

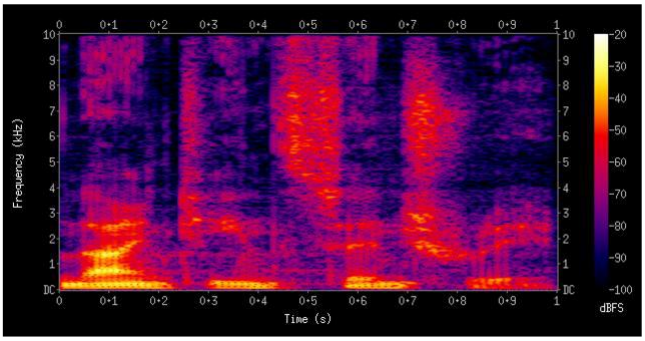


Figure 5. Frequency vs time

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 1024)              41984

 dropout (Dropout)           (None, 1024)              0

 dense_1 (Dense)             (None, 512)               524800

 dropout_1 (Dropout)         (None, 512)               0

 dense_2 (Dense)             (None, 256)               131328

 dropout_2 (Dropout)         (None, 256)               0

 dense_3 (Dense)             (None, 128)               32896

 dropout_3 (Dropout)         (None, 128)               0

 dense_4 (Dense)             (None, 64)                8256

 dropout_4 (Dropout)         (None, 64)                0

 dense_5 (Dense)             (None, 32)                2080

 dropout_5 (Dropout)         (None, 32)                0

 dense_6 (Dense)             (None, 10)                330

=================================================================
Total params: 741,674
Trainable params: 741,674
Non-trainable params: 0
_________________________________________________________________
```

Figure 6. Sequential model screenshot

4. EXPERIMENT

4.1. Experiment 1

The first experiment is built into tensorflow, where the values of the loss, accuracy, val_loss, and val_accuracy are given in a dataset, which can then be plotted as a graph. The sample size of the train test split is 80 percent training, and 20 percent testing, so with this ratio we can see any overfitting or underfitting of the dataset or overly high or low training accuracies that do not translate to better val_accuracy [14].

This is a graph of the experiment we performed (fig 7). we can see that there is overfitting of the graph after about 50 epochs as the loss went down but the val_loss increased. This means we needed to make adjustments in the epochs as it started overfitting after a 50 epochs.

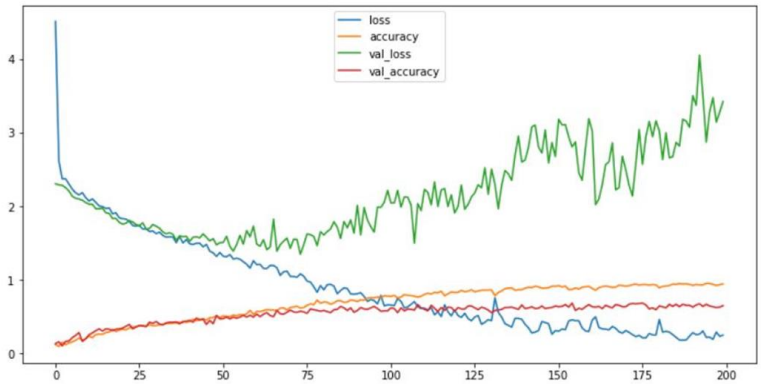


Figure 5. Voice vs mass 4.2. Experiment 2

The next experiment is inside of the android studio test, where we input two songs from two different genres like “Blank Space” by Taylor Swift, and m.A.A.d city by Kendrick Lamar. They test the algorithm and the app by sending the song to the server and run the algorithm, as well as testing the app’s ability to play and equalize the audio.

By seeing the output we can see that it was sorted and the process of uploading the audio, using the neural network, and getting the genre classification back works. This also shows that the app was able to receive the genre, and apply a filter. And most importantly we can hear that an equalizer was able to be applied.

The experiment addresses the problems above by figuring out if the feedforward neural network was actually having good accuracy for sorting the genre. Ultimately, it was able to prove some overfitting of the data, but also show little error for obvious music choices.

5. RELATED WORK

The 3D-DCDAE: Unsupervised Music Latent Representations Learning Method Based on a Deep 3D Convolutional Denoising Autoencoder for Music Genre Classification by Lvyang Qiu, Shuyu Li and Yunsick Sung was used to inspire the use of the GTZAN dataset on kaggle as well as combining some of the seven features they used such as MFCC, spectral roll-off, zero-crossing rate, chroma frequency, and rhythm histogram [11].

Automatic Music Genre Classification Based on CRNN by Yu-Huei Cheng, Member, IAENG, Pang-Ching Chang, Duc-Man Nguyen, and Che-Nan Kuo helped the experiment out by testing out the limits of CRNNs and Short-time Fourier transform to compare the accuracies [12]. By basing off their research, we were able to choose feedforward neural networks as more accurate model for classifying the music.

Comparing the Accuracy of Deep Neural Networks (DNN) and Convolutional Neural Network (CNN) in Music Genre Recognition (MGR): Experiments on Kurdish Music Aza Zuhair and Hossein Hassani helped us with the feature extraction and possible future implementations of data collection [13]. The feature extraction introduced the librosa library to extract things such as MFCC, as well as the shape of the spectral envelope into duration and segments for the neural network to include.

6. CONCLUSIONS

In the future, we hope that we can input streaming and cross app streaming into Melodic EQ so that more apps and audios can be processed in the background during all uses of audio. This can be achieved by asking for permission to constantly run in the background. The most important step would be detecting if any audio is playing and then taking the audio from the other app and then incorporating it into the Melodic EQ app and sending it to the server and back. Lag would need to be cut down and other problems such as compatibility with audio would also need to be addressed. However, if this works in the future, the app would be a non-interfering user experience that would only enrich the user’s experience with audio.

Current limitations include the accuracy of the dataset, as even more advanced neural networks are not achieving higher accuracy than simple feed forward loops. Until then there are limitations in software to get better accuracy. Another important step is the integration of streaming into the app, which will likely never happen. This severely limits the amount of songs that can be played

and the usability of the app in a simple manner. Lastly, is the user's trust if we implement background running, many users are protective of their privacy and we need to build that trust to allow the app to work in the background.

In the future, I hope to increase the accuracy as software improves and better optimization and parameters help with the accuracy. Finding better labeled music data might also help as music evolves as time moves on, ultimately the end goal is to improve the accuracy of the model with no lag.

REFERENCES

[1] Datta, Hannes, George Knox, and Bart J. Bronnenberg. "Changing their tune: How consumers'adoption of online streaming affects music consumption and discovery." Marketing Science 37.1 (2018): 5-21.

[2] VÂRLAN, Petre Marcel. "THE EMOTIONAL QUOTIENT – A FACTOR OF MUSICAL KNOWLEDGE." Bulletin of the Transilvania University of Brasov, Series VIII: Art & Sport 5 (2012).

[3] Zhu, Hao, et al. "Deep audio-visual learning: A survey." International Journal of Automation and Computing 18.3 (2021): 351-376.

[4] Roy, William G., and Timothy J. Dowd. "What is sociological about music?." Annual Review of Sociology 36 (2010): 183-203.

[5] Prey, Robert. "Nothing personal: algorithmic individuation on music streaming platforms." Media, Culture & Society 40.7 (2018): 1086-1100.

[6] Bar-On, Reuven. "Emotional and social intelligence: Insights from the Emotional Quotient Inventory." (2000).

[7] Silla, Carlos N., Alessandro L. Koerich, and Celso AA Kaestner. "A machine learning approach to automatic music genre classification." Journal of the Brazilian Computer Society 14.3 (2008): 7-18.

[8] Wlömert, Nils, and Dominik Papies. "On-demand streaming services and music industry revenues — Insights from Spotify's market entry." International Journal of Research in Marketing 33.2 (2016): 314-327.

[9] Esmaeel, Hana R. "Apply android studio (SDK) tools." International Journal of Advanced Research in Computer Science and Software Engineering 5.5 (2015).

[10] Van der Aalst, Wil MP, et al. "Process mining: a two-step approach to balance between underfitting and overfitting." Software & Systems Modeling 9.1 (2010): 87-111.

[11] Qiu, Lvyang, Shuyu Li, and Yunsick Sung. "3D-DCDAE: Unsupervised music latent representations learning method based on a deep 3d convolutional denoising autoencoder for music genre classification." Mathematics 9.18 (2021): 2274.

[12] Cheng, Yu-Huei, et al. "Automatic Music Genre Classification Based on CRNN." Engineering Letters 29.1 (2020).

[13] Zuhair, Aza, and Hossein Hassani. "Comparing the accuracy of deep neural networks (DNN) and convolutional neural network (CNN) in music genre recognition (MGR): experiments on Kurdish music." arXiv preprint arXiv:2111.11063 (2021).

[14] Koehrsen, Will. "Overfitting vs. underfitting: A complete example." Towards Data Science (2018).

[15] Abadi, Martín. "TensorFlow: learning functions at scale." Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming. 2016.

MACHINE LEARNING AND CHATTING SERVICES IN COLLEGE SEARCH: A STUDY ON STUDENTS' PREFERENCES AND SATISFACTION

Jiuyu Zhang[1], Sanskar Pokharel[2]

[1]La Jolla Country Day School, 9490 Genesee Ave, La Jolla, CA 92037
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

How does Machine Learning fit into the aspect of a highschooler searching for their dream college? It is widely known that being in high school is a stressful for most high school student, and creating something that can help them find the best college is important. Knowing that high school students like to participate in social activity, creating a chatting service that uses machine learning serves the purpose of helping them find their most suitable college without increasing the stress of their already stressful situation. To prove that this method is suitable for the situation, a survey asking 1000 high school students whether they would choose the chat option or traditional fill-out a form method, majority of them preferred the chat way. Based on the results supporting my thesis, I developed an application that will help high school students find their best college that consists of a chatting service that helps students achieve their goal with the help of Machine Learning.

KEYWORDS

College, Machine Learning, AI, High School

1. INTRODUCTION

Communicating with friends via a chat app is now incorporated into the daily life of a high school student, and finding the best colleges could be a challenge that is in the daily life of a high school student as well [1]. Creating an app that uses a feature that simulates a chatting environment can in fact help the student fight their challenge and can in fact make the college application easier. Using the popularity of using chat apps, it would also be easier for high school students to adapt to the new way of finding colleges. Since students already know how to use chatting software efficiently and easily, they would find it easy to ask the chatbot the questions that they want answers to [[2]. Using machine learning as the backbone of the chatbot is also beneficial to the user experience, as each message sent by the user will be analyzed by the bot and each time the user chats with the bot, the response will be more like a casual human being [3]. And based on government information, the bot automatically updates its college database with the up-to-date information, ensuring that the student gets the most up-to-date information about each college.

Some of the existing techniques and systems that have been proposed to help high school students find their dream college, which allows the user to fill out a long and extensive form to

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD- 2023
pp. 211-218, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130518

achieve the promised results. However, these proposals assume that students have the time to complete the form, which is rarely the case in practice. Other techniques, such as analyzing a quiz that they make students take. They failed in making this an accurate analysis because of the variation of the students' mood when they are taking the test, making their algorithm useless for accounting the error margins and often results in incorrect scoring. A second practical problem is that some users find it hard to understand the results returned by other software. Most college finder returns a wide array of colleges that they might be interested in, often needing the user to click into each to see details. This unnecessary step caused many to step away from the website and search for other easier services to use.

In college core, the chat feature stands out as the main tool to help the user achieve their goal. By using the power of Machine Learning, the chatbot understands the input of the user and respond with necessary questions to determine the information that the user wants. Our goal for the chat module in CollegeCore is to create a complete network of questions so that our users will have more information about what they need. This method is inspired by the automated support popup that some company (e.g. HP) have on their website, these pop ups are especially useful when we need help on something about the company [4]. Because of how helpful it is, having a similar option in the college counseling side could also be helpful to the user. Existing tools ask users to fill out a long form, and often having unnecessary results showing up after completing the form. CollegeCore's Chat feature, however, allows for a more fun and stress-free experience, and with the chat taking account of the user's datas, a more accurate and personalized result

By conducting a survey on US high schoolers of their opinion of using CollegeCore Chat versus normal form-based counseling, most survey takers claimed they prefer CollegeCore. This survey is conducted with a population size of the number of students from both a public and private high school, with the sample size of the students that are in 11th or 12th grade. Survey takers are asked to use college core to try to find the college he/she might be interested in, and after using college core, they are asked to use the normal form-based counseling method. After performing both tasks, they are asked on their preference, accuracy (whether both suits the student interest), and time (which method gives the result faster). The survey result is then divided into 11th grade pool and a 12th grade pool, with 11th graders in the 11th grade pool and 12th graders in the 12th grade pool. Both pools have similar result. Both showed preferences towards using CollegeCore Chat, and 82% of 11th graders have a better accuracy with CollegeCore, while 79% of 12th graders have a better accuracy with CollegeCore. All participants reported that CollegeCore chat returned the results faster. This survey reflected the advantage of usingCollegeCore. It should also be noted, that for both CollegeCore and form-style counseling services, the participants are all using fresh accounts to ensure fairness in the survey.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we faced during the experiment and during the design phase of the project; 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 papers that is relevant to CollegeCore in Section 5. Finally, Section 6 gives the conclusion remarks and points out the project's future work.

2. CHALLENGES

To build the project, a few challenges have been identified as follows.

2.1. Appropriate and Accurate Data About the Colleges

Making a chatbot about college is useless without appropriate and accurate data about the colleges [5]. Without accurate college data, users will be given incorrect information about their dream college, and it would be harder for the user to apply and get accepted to it.To find most accurate data, I searched the government and education websites around the world, in attempt to find the most accurate dataset for the bot to use while it is chatting with the users. The final data used are combined data from multiple credible sources (sources from government websites or trustable research websites). These data were then converted to a JSON file and were loaded into our backend server, which will analyze the data given by the user and find the best suitable information in the dataset to return to the user [6].

2.2. Keeping the Data Up to Date

Keeping the data up to date is also important for the accuracy of datas. Datas of the thousands of colleges around the world change all the time, and it is important for them to be the most recent. But it is not good enough to update it with different patches with the application. It must be updated regularly. The backend of the applications, which handles all the machine learning and AI chatting operations, are used to instantly update the data. The updated data, just needs to be uploaded to the backend server, and the data update will be complete. The datas are frequently updated, and they are usually updated 2-3 times per month to ensure that the data is accurate and up to date. Some datas that are not updated for years are marked and upon next update they will be treated as important updates so it receives a much more comprehensive search.

2.3. Data Together

Analyzing said data together is also important for a good user experience. Without good organization on how to analyze the data, it would also be hard to providean accurate data. With python's TensorFlow library, I was able to combine understanding the user's request and the analysis to retrieve the best result possible to return to the user. With the result from the algorithm, CollegeCore can return the suitable and crucial information to the user, allowing them to plan their high school life better. With on-demand backend pushing, the features available to the chatting service will be fine-tuned and updated frequently, allowing more features at a lower device storage usage rate. The communication between the frontend and backend are also monitored and will be used to train the machine learning furthermore to ensure a better and more accurate understanding of the user's intentions.

3. SOLUTION

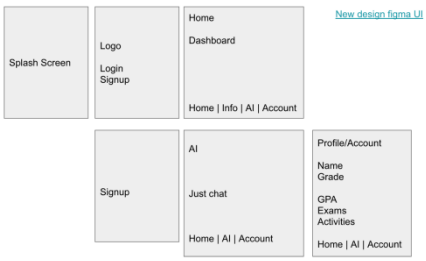


Figure 1. Overview of the solution

CollegeCore is a College Counseling app that aims to help the user, usually a middle or high schooler, to find their dream college and help them get through high school. As Figure 1 shows, users are first greeted with a Splash Screen which explains to them how the app works, then the user is led to login and sign-up page, where they can create their personalized CollegeCore account [7]. The user will then have access to our chatting system, where the user can then chat with our bot about anything, and the bot will try to answer as accurately as it can. The user could also edit their account information, such as their email, grade, and gpa, as the chat bot also consider the grade and gpa as a factor in producing the accurate answers. The signature feature, the Chat, consists of the following steps. The user first enters their message into the chat box, and press send. The message then gets embedded into a JSON object that contains other necessary information and sends it to the backend server. The backend then uses machine learning to convert the string of message to a specific category of answers, then using that category, the backend server analyzes it and then replies with the necessary information. If the backend needs extra information, it would ask the user about the question, and then process the returned information.

```
# Building a Model
# tensorflow.reset_default_graph()
tf.compat.v1.reset_default_graph()

net = tflearn.input_data(shape=[None, len(training[0])])
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, len(output[0]), activation="softmax")
net = tflearn.regression(net)

model = tflearn.DNN(net)


model.fit(training, output, n_epoch=2000, batch_size=8, show_metric=True)
model.save("chatbot/model.tflearn")
```

Figure 2. Building the Machine Learning Model

Figure 2 shows the necessary code to build a machine learning model with TensorFlow [8]. The model is trained 2000 times in this example, but the number will be changed based on how much data each training set contains. The more data the training set have, the less time it will be trained. However, if the trained data have more important and harder to train data, the times will be increased to make sure that the data is as accurate as possible.

```
Future<void> postMessage(String message) async {
  final url = Uri.parse(chatUrl);

  addNewMessages(message, "user");

  final header = {"Content-Type": "application/json"};
  final _data = {
    "msg": message,
    "context": currentContext,
    "botParams": currentBotParams
  };
  final res = await post(url, headers: header, body: json.encode(_data));

  if (res.statusCode == 200) {
    Map<String, dynamic> response = json.decode(res.body);
    if (response['context'] != null) {
      currentContext = response["context"];
    } else {
      currentContext = "";
    }
    if (response['botParams'] != null) {
      currentBotParams = response["botParams"];
    } else {
      currentBotParams = "";
    }
    print(currentBotParams);
    addNewMessages(response['msg'], "receiver");
  }
}
```

Figure 3. Communications to Backend

Figure 3 is a picture of how the chat messages gets sent to the backend for analysis. As the _data variable shows, each passage contains 3 different options. Msg is the message sent by the user, context is used to identify the category of the question/answer the user is sending, and bot params stores all the previous responses, if the category requires multiple responses. This structure allows for a better communication between the backend and the frontend, and it made sure that

the correct info is obtained and sent to the user. The communications are established using HTTP Rest methods, over https protocol, ensuring secured communications and enhanced communications between the user and the server would be secure. The ensuring the security of the communications, it can be ensured that the personal datas, such as the answers to questions, would be encrypted and secured, so no people would illegally obtain the user's datas.

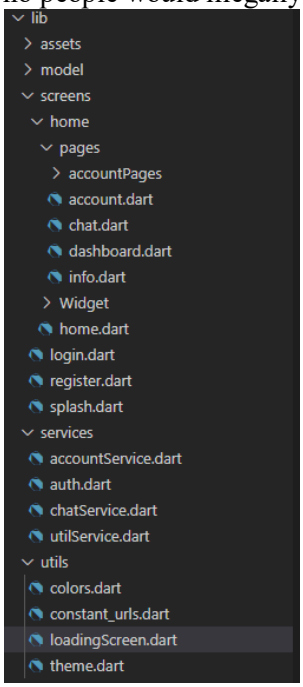


Figure 4. File Structure

Figure 4 shows the file structure used to create the frontend of the CollegeCore App. There are 4 pages in total at the time of writing (dashboard, chat, info, and account), and each time a page is opened, the current page they are on is updated to the new page instead of creating a new page, therefore ensuring a more optimized app. Also, repeated usage of dart/flutter codes are stored for easier usage and better optimization, making it easier for us to develop and update the app. We use Firebase as our database, as our effort to try and make accounts and user datas more secure [9]. User's data are very secure under firebase becauseno one, including the developers, except for the user, will be able to see the password.

An app cannot look good without proper colors, and that's why CollegeCore have defined all the necessary colors in a single file, ensuring that they could be changed swiftly.If our users reported that they do not like the theme, or for general updates. Our users can provide feedback to us for anything and if they are valid, they will be fixed as soon as possible.

4. EXPERIMENT

4.1. Experiment 1

To verify that people prefer chat-style college planning app instead of the traditional form-based college planning app, we have set a random college in the US, and attempted to use a traditional form-based College planning app and CollegeCore to test the time needed and the number of questions needed for a user to get the result to that college. Each college will have 3 trials, with each trial containing the times needed to get that college for both methods. There will be 5 colleges in total.

Table 1. Results of experiment 1

	Trial 1		Trial 2		Trial 3	
	CollegeCore(sec)	Traditional(sec)	CollegeCore(sec)	Traditional(sec)	CollegeCore(sec)	Traditional(sec)
UCSD	32	65	34	71	28	64
Hamilton College	45	70	48	68	48	65
Carnegie Mellon University	27	40	25	46	33	43
Boston College	25	51	31	49	29	45
Columbia University	31	45	30	49	32	42

The result of this experiment was not surprising, based on the premises of CollegeCore to simplify the process. The CollegeCore method is usually quicker than the Traditional method of trying to find a specific college, as seen in Table 1. For each trial and for each university, CollegeCore is usually around 20 seconds faster than the traditional methods. This, could be because of there are less repetitive questions being asked every time it is ran, because essential information, such as gpa, can be retrieved in the user profile. Eliminating a whole question because the information can be retrieved at any time, without having to ask. This removes at least 10 seconds from the total time, causing an easier and faster experience.

4.2. Experiment 2

Another way to make sure that CollegeCore is accurate is to see if the result match a person's dream college that they had before they used CollegeCore. This proves that collegecore is a better solution to college planning because it can returnan accurate result, even without all the necessary questions that traditional college planning apps have. There are 5 participants in this experiment, and each participant are asked to use college core 3 times without using the same answers for each trial. They are then asked on whether the app returned the correct college.

Table 2. Result of experiment 2

	Trial 1	Trial 2	Trial 3
Person 1	Yes	No	Yes
Person 2	No	No	Yes
Person 3	Yes	Yes	Yes
Person 4	Yes	Yes	no
Person 5	No	Yes	Yes

Of all the 15 responses of 5 randomly selected participant, ⅔ of the responses reported showed that they did indeed receive the right college. Although this ratio does not look like a bug percentage, considering the four thousand schools in the US, it would be hard to eliminate down to just 1 school for every trial. The negative options in the trials can also be explained because participants are asked to use different responses on each trial, showing how some of the different inputs can also led to the program returning a different school. Other factors that could result in a different school returned could be because of unrecognizable inputs the user might reply, or if the user have not fill out their profile data correctly.

These results showed how CollegeCore's chat feature could get its users results faster and more accurate than the normal services. In the first experiment, for 5 random schools from across the

us, ALL can be obtained faster than the traditional methods, proving that for most universities, CollegeCore can save the user time in looking up the college they wanted, saving more time for the student to work on their school work. From the second experiment, it showed the accuracy of CollegeCore. As ⅔ of the trials showed a positive result, it can be said that the result provided given the shorter time it took to return the college, is accurate. As a result of the experiments, it can be said that the dream college of a user could be accurately returned within a shorter amount of time than the traditional, formed-based methods.

5. RELATED WORK

Using several types of machine learnings could help students in achieving career placements [10]. The author of this source went into a dive into the relation between Machine Learning (ML) and Explainable AI(XAI) and talked about how a model should be trained. However, my paper talked about the method and the best solution to optimize and solve a problem that the existing industry has. For example, Guleria and Sood talked about the Pre-modeling, Explainable modelling, and post-modelling stage of the Machine Learning model, while I focused more on a broader scale, talking about the general solution and the results of the tested solution.

This article, by Yang and Talha, focuses on how Big Data technology and Artificial Intelligence are affecting college counseling firms [11]. Yang and Talha's paper shares similarities with this paper as they both studies the aspect of using machine learning in the field of college counseling, but Yang and Talha's paper also talked into the Big Data technology, stepping beyond what this paper studied and explained. The consideration of big data might even make the college counseling apps even moreaccurate based on how many datas can be taken from the world and fitting it to predict the right answer, making it even a step over how CollegeCore handles it.

This paper is like my project as it discusses the client-server communications [12]. It showed structures showing how a typical client-server model and how this model benefits websites and applications, while also discussing the possible issues that the model has. This is exactly the structure that CollegeCore uses for front-backend communications. CollegeCore uses a server from Digital Ocean, a virtual private server(vps) provider that is trustworthy and uses a secure and private URL, with authentication checks built-in to ensure secure data transfer. This paper showed a clear view on how front to backend communications should be handled and how to make it secure and have optimized performance.

6. CONCLUSIONS

Overall, CollegeCore is an app designed to help high school students to find their dream college through Machine Learning and AI chat bot. The chat bot, will understands the dialogs between the bot and the user, and the backend will produce the accurate result that the user has requested. The Chat service is effective, as the method is tested in experiments, where the result showed that the chatbot can respond accurately to the user's questions. The time-to-result experiment, which tested the time needed for college core to find the college and compare it to the times needed for traditional style college planner methods to find the college. The result is not surprising—CollegeCore were able to beat the traditional ones by 10-40 seconds. The second experiment conducted, the result-accuracy experiment, which asked participants to use collegecore and see if they receive the college the same as their dream college—two thirds of the trials turn out to be accurate. Considering there are as much as 4000 colleges in the US, achieving that rate is quite impressive. Analyzing the results of the two experiments, it can be concluded that if the user answers the questions the bot asks truthfully, the bot will return a close result to their dream college. With the chatting method, however, users are feeling less stress, as the method of the

chat is simulating the users chatting with a friend, coming as a less intense way of asking questions.

The possible limitations of CollegeCore AI Chat include understanding of spelling errors in messages. If the messages sent have considerable number of misspelled words, the model might not be able to categorize the request, and therefore having the mark the message as indecisive. This could only be improved by providing a larger data and by training the model multiple times. Other limitations might include the data structure. Although the data is updated frequently, there are occasionally errors in college data's JSON format.

The limitations could be solved in the future by introducing newer technologies as they come up. With the rise of Big Data technology, for example, the accuracy of the machine learning model could be improved, and most importantly, the data for colleges could also be improved. Another way is to make user editions available. If a user spots an inaccurate detail, they could edit it, where they could be approved by developers and be updated if the detail is verified. This ensures the maximum accuracy of college datas.

REFERENCES

[1] Botha, Johnny, W. C. Vant, and Louise Leenen. "A comparison of chat applications in terms of security and privacy." Proc. 18th Eur. Conf. Cyber Warfare Secur.. 2019.
[2] Adamopoulou, Eleni, and Lefteris Moussiades. "An overview of chatbot technology." Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II 16. Springer International Publishing, 2020.
[3] Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR). [Internet] 9 (2020): 381-386.
[4] Adair, Bill, et al. "Automated pop-up fact-checking: Challenges & progress." Proceedings of the Computation+ Journalism Symposium. 2019.
[5] Wilcox, Bruce, and Sue Wilcox. "Making it real: Loebner-winning chatbot design." ARBOR Ciencia, Pensamiento y Cultura 189.764 (2013): 1-13.
[6] Bangare, S. L., et al. "Using Node. Js to build high speed and scalable backend database server." International Journal of Research in Advent Technology 4 (2016): 19.
[7] Yilmaz, Yagiz, et al. "Investigating the impact of ransomware splash screens." Journal of Information Security and Applications 61 (2021): 102934.
[8] Abadi, Martín. "TensorFlow: learning functions at scale." Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming. 2016.
[9] Khawas, Chunnu, and Pritam Shah. "Application of firebase in android app development-a study." International Journal of Computer Applications 179.46 (2018): 49-53.
[10] Guleria, Pratiyush, and Manu Sood. "Explainable AI and machine learning: performance evaluation and explainability of classifiers on educational data mining inspired career counseling." Education and Information Technologies (2022): 1-36.
[11] Yang, Zhen, and Muhammad Talha. "A coordinated and optimized mechanism of artificial intelligence for student management by college counselors based on big data." Computational and Mathematical Methods in Medicine 2021 (2021).
[12] Oluwatosin, Haroon Shakirat. "Client-server model." IOSR Journal of Computer Engineering 16.1 (2014): 67-71.

EMOTIONAL MUSIC GENERATION: AN ANALYSIS OF EFFECTIVENESS AND USER SATISFACTION BY USING PYTHON AND DART

Crystal Chong[1], Ang Li[2]

[1]Chandler School, 1005 Armada Dr, Pasadena, CA 91103
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

An issue that is prevalent in today's society is the need for new music to be generated. More people are uploading videos and other forms of content to the internet through social media, and videos can often be enhanced by adding music to them [6]. However, creating music can be a time-consuming and expensive process. Therefore, an application was created that can generate music using emotions as inputs for the music generation model. To test how well the method of music generation through sentimental analysis works, an experiment was conducted that tests how accurately a sample of participants believe that the generated music was on a scale from one to ten [7]. According to the results of the experiment, the application appears to do fairly well at generating music that accurately represents the sentiment that was intended in the inputted message. A survey was also conducted to test user satisfaction when working with the application to generate music. The feedback from the participants indicated that they were generally satisfied with how well the generated music matched their intent in the inputted message, and they also seemed to be very satisfied with how convenient the application was to use and how intuitive the user interface was [8]. However, as the ratings for convenience were much higher than the ones regarding the effectiveness of the music generation itself, this may indicate that the application still has room for improvement when it comes to recognizing the sentiment of the inputted message.

KEYWORDS

Music generation, Sentimental analysis, Machine learning

1. INTRODUCTION

Music has been a vital part of human history, and music can influence how someone feels and provide a path for people to express their thoughts and emotions. For this reason, music has been utilized in movies, plays, and other forms of entertainment. Furthermore, music has been demonstrated to provide benefits such as lowered stress levels, boosted cognitive performance, and improved mood [1]. There are many various genres of music, from country to pop to death metal; this means that no matter what sort of preferences a person may have, there is likely a type of music that exists to suit any person's tastes. While music can have the consequence of hearing loss due to being played too loudly, this can be avoided fairly easily if the music is played responsibly at a reasonable volume, or concerts are not overly visited.

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 219-228, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130519

An explanation of why music is so important is that it is a form of expression and bonding. Musicians are provided with a channel to express themselves, and listeners can feel the emotions that the musicians intended. Music is often played during parties and other gatherings, as other people who have heard the songs before may be able to dance or sing to music together as a bonding activity. Music is often used as a part of culture and sharing this cultural music with others is a way to spread and combine cultures to make a more diverse and inclusive society.

Currently, something that has been done to meet the demands of new and fresh music is AI-generated music [9]. With the ability to constantly create new music without the requirement of manpower, AI-generated music is a very cheap and efficient solution that is available for the general public to use. However, the applications regarding AI-generated music that are currently available are not as intuitive to use as people may want them to be. In an ideal scenario, users could tell the AI exactly what they want to be included in their music using words, and the AI would construct the music exactly as the users instruct them to. However, the applications that we currently have access to are not that advanced. Instead, AI-generated music is performed by inputting music that already exists and building a similar song, or the music is generated based on factors such as whether the song should be in major or minor. With the first-generation method, a similar song that exists is required, and that may not be possible if the user wants to generate a unique kind of song and is not experienced in music creation [10]. The second-generation method has its flaws as well. Although users can take control of several factors, the options are generally quite limited. Rather than creating the song exactly to their liking, users will have to instead opt for the next best choice of generating a song that is only vaguely reminiscent of the song concept that they envisioned. With how restricted current AI technology involving music generation appears to be, there is still much room for improvement in the field. Something that is still a relatively untouched part of AI-generated music is the use of sentences and language to directly influence the outcome of generated music.

The tool that was created to help people with creating new music based on their needs is a mobile application that involves AI-generated music. Rather than commissioning people to create music, which would be both expensive and time-consuming, the artificial intelligence will simply take in a message that the user inputs, determine what sentiment the message gives off, then generates music based on what the sentiment of the user-inputted message was determined to be. This application may sound similar to other applications or programs involving AI-generated music. However, what separates this application from others is its intuitiveness. While other applications ask their users to check certain boxes and toggle certain options to tell the artificial intelligence how the music should be made, this application only requires the input of a message. Many members of the general public wish for applications to be easily used, and they may completely ignore complicated-seeming applications because they believe that taking the time to learn how those applications work is not worth the effort. On the other hand, providing these users with something incredibly easy and convenient to operate may make them give the application a chance and even potentially become long-time users of the application.

The effectiveness of the application was tested by combining an experiment with a survey. First, twelve participants were gathered to download the application and input ten different messages into the application. With each inputted message, the participant would listen to the music that the application outputted and determine whether the music accurately reflected the sentiment inside the message. The participant would record the number of times that the application successfully outputted music that represented the inputted message's sentiment, and this data is to be recorded in a table. The survey would be conducted immediately after the experiment is over, which ensures that the experience of using the application is still fresh in the participants' minds. The survey would ask whether the application was able to effectively create music based on their experience with the application in the experiment and whether the application was convenient to

use and intuitive. The participants are offered a scale from one to ten to rate how they feel about each aspect of the application. The survey also provides a free-response section at the bottom of the survey to allow for more customized feedback. While both the experiment and the survey are being conducted, each participant is encouraged to not collaborate with any other participants when completing these, so that the data represents the participants' uninfluenced opinions of whether the sentiment is accurately reflected or not. The experiment's purpose is to determine whether the application is successful in terms of whether the application can provide users with the music that they envision. The survey's purpose is to determine whether the general user interface and music generation process is smooth, intuitive, and performs up to standards from the user's perspective.

My paper will be organized so that the remainder is split into five separate sections. Starting with Section 2, the challenges that were encountered when planning the project and developing the application will be described in detail. The next section, Section 3, will cover how the application was created, starting with a general explanation before diving into a deeper view of how each feature and section was implemented. Section 4 will go over the experiments that were performed to test the effectiveness of the application. Section 5 compares and contrasts this work with related works. Lastly, Section 6 will give concluding remarks as well as a reflection on how the application could be improved in the future.

2. CHALLENGES

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

2.1. A Method to Generate New Music

The first challenge that was faced when starting the project was coming up with a method to generate new music. Human-made music was not an option, as this was specifically what was trying to be avoided. The most reasonable option seemed to be using an AI model to create the music. With an AI model, no manpower will be required in the creation of new music, and besides possible small server costs, this option was relatively cheap. Another choice to make was what device the application should be on. Considering that many people in today's society spend much of their time on a smartphone or mobile device and carry it wherever they go, making the application runnable on those devices would likely be best. While it would be helpful to have the application available on computers as well, a mobile application was decided upon as the top priority.

2.2. What Features Should be Included in the Application

The next challenge that was encountered was what features should be included in the application. The application's main purpose is to generate music, but it may be difficult to gauge how much should be included within the application. Ideally, the application should be simple to use and intuitive, yet have enough functionality for people to use this application as a reliable tool. Furthermore, the application should be lightweight, so that it takes very little storage and is quick to run. Keeping this balance in mind, the application only includes the necessary features of generating a song based on the emotion it was provided with as input. With an application that only has a single goal and a clean, distinguishable user interface, new users of the application will likely have no trouble using it. Although offering users with many different ways to generate music was considered, the idea seemed as if it would clutter the application too much, and this idea was ultimately scrapped.

2.3. The Implementation of the Functionality Itself

A third challenge was the implementation of the functionality itself. Because no easily accessible and reproducible version of artificial intelligence code could take in emotions as input to create music, some steps were taken to achieve this result. First, the problem was broken down into two main parts; the first part was assigning an inputted message to an emotion such as happiness or sadness, and the second part was selecting a song based on the emotion that the AI model recognized. Both parts were done within the same Python file, which allowed all the code to be run in one place and helped with implementing the functionality of the code more conveniently. The two parts of the code individually were relatively simple to find examples of on the internet. Although the functionality of the code seemed extremely difficult at first, diving the problems into more understandable and approachable sub-problems allowed the implementation to be a success.

3. SOLUTION

The application was created using Python and Dart as its programming languages. Flutter is an open-source framework backed by Google that is used as the front end of the application, as Flutter is proficient at creating application interfaces. Flutter is capable of creating new screens and all the necessary features that would be required in the application screens, such as buttons, text boxes, images, and layouts. An example of where the use of Flutter's functionality can be seen in the application is on the home screen, as a button is programmed to move to a different screen in the application when pressed [11]. Python is primarily used for the back end of the application since Python has many different libraries and frameworks to choose from. Therefore, any sort of machine learning or artificial intelligence feature that may be needed for a project will most likely be found in one of the numerous available libraries. In the case of this application, the open-source software library Keras was used due to its ability to create and utilize a Long Short-Term Memory neural network. The Long Short-Term Memory network is known for being able to remember information and predict sequences much more effectively than ordinary recurrent neural networks [5]. The music generation is created with the help of MIDI files. As MIDI files only store note data such as the length of the note, the pitch of the note, and the volume of the note, it is much easier for the neural network to use as data.

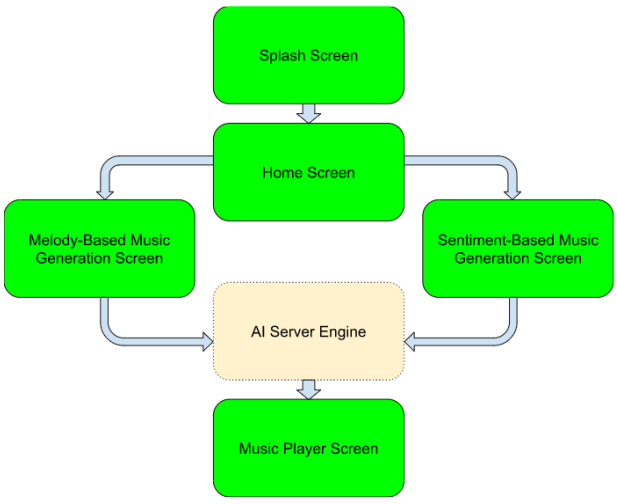


Figure 1. Overview of the solution

In the Flutter code, five primary screens are used to make up the interface of the application. The first screen is the splash screen, which only appears on the screen for a few seconds when first opening up the application and shows the logo of the application. The goal of this screen is to provide a smooth transition into the application for the user, and it was implemented primarily for aesthetic purposes. The splash screen was created with a Future that utilized a delay for a predetermined number of seconds before making a call to transition into the next screen, which is the home screen. With the home screen, there are two available options to choose from: melody-based music generation and sentiment-based music generation, and there are two buttons that lead to each page. In the melody-based generation screen, the user will be allowed to input an audio file by pressing a button, which will open up the mobile device's storage and show files with audio file extensions to choose from, such as MP3 and WAV [15]. When the file is chosen, a request is sent to the neural network code, then brings the application to the music player screen once the newly generated music has been completed. The other choice, which is sentiment-based generation, is done by prompting the user to type a message into a text box object added in Flutter. The Natural Language ToolKit in Python is used to extract the sentiment from the message and output a suitable piece of music based on the determined sentiment. The final primary screen is the music player screen, in which the player could listen to the generated music. A simple interface is created that allows the user to press a button to play it, and the user can see the progress of the song that is being played with a slider that automatically moves as the song plays and monitors the song's progress.

The Python code was responsible for generating the music [14]. Since Flutter and Python are two separate programming languages, a Flask server is used to connect the two languages through HTTP requests. Once the application requires that music be generated, the audio or message that was inputted by the user (depending on which method is used to generate the music) is sent through an HTTP request to the Python code, which will create and run a Long Short-Term Memory neural network using the data from the HTTP request. To create the neural network, a pre-trained model is fetched that was only trained using the piano; this is because using only one instrument to train seems to produce more accurate and desirable results. In the code to run the neural network, a certain number of notes are defined to be generated. Finally, the MIDI file is converted into a WAV file, and the WAV file is sent back to the application for the user to listen to from their mobile device.

4. EXPERIMENT

4.1. Experiment 1

An experiment to test the effectiveness of the application at generating music based on the sentiment that is provided. Twelve participants were gathered for the experiment, which should be a large enough sample size to account for any variability. Each participant was instructed to input ten different messages with sentiment, listen to the outputted audio, and record how many times they believed that the outputted music invokes feelings that are similar to the inputted message or not. As some people may have different interpretations of what is considered "accurate" in terms of sentiment, having multiple participants test this may help to counter any bias.

Participant Number	Number of Successful Runs
1	5
2	8
3	8
4	9
5	7
6	10
7	8
8	7
9	8
10	6
11	10
12	9
Average	7.917

Figure 2. Table of experiment 1

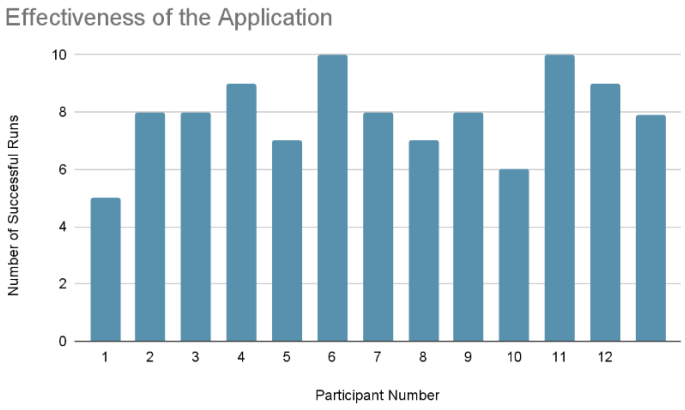


Figure 3. Effectiveness of the Application

From the results, it seems that the application had a decent success rate when it came to generating a piece of music that matched the sentiment in an inputted message. The highest score, which was 10, was achieved twice. On the other hand, the lowest score was 5. Overall, the average score was approximately 7.9. Out of every 10 pieces of music to be generated from the application, a little over 2 of them do not accurately represent the inputted sentiment. The majority of the participants who left feedback stated that they were impressed by how well the application worked. While the data seemed to fluctuate among the participants, this may be because their interpretation of what was considered "accurate" to the inputted sentiment may be more lenient or stricter depending on who it was that recorded the data. By gathering the average of all the participants' data, a balance could be found.

4.2. Experiment 2

A survey was conducted over Google Forms, and the twelve participants from the previous experiment took the survey. The first question asked whether the application is effective at creating new music based on the sentiment that was inputted. The second question asked how convenient and easy the application was to use. For both of the questions, the participants were provided a scale from one to ten to use as their answers. At the end of the Google Forms survey, a free-response survey is provided; if the participants have any feedback that they wish to express that cannot be expressed with the previous questions, they can do so.

Participant Number	Music Generation Effectiveness	Convenience/Ease of Use
1	6	7
2	8	8
3	9	9
4	8	9
5	7	10
6	10	10
7	8	8
8	7	7
9	7	8
10	6	8
11	9	9
12	8	9
Average	7.75	8.5

Figure 4. Table of experiment 2

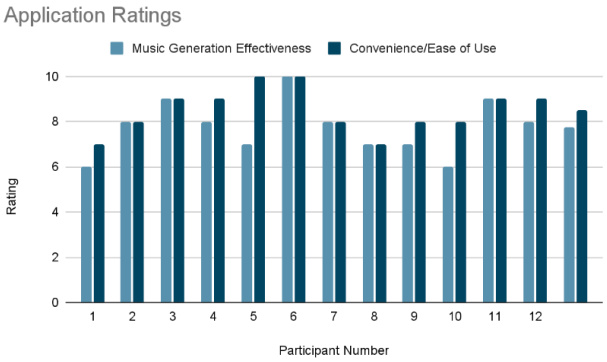


Figure 5. Application ratings

The table and chart indicate that the application is both competent at producing music to the users' preferences and intuitive to work with. The average rating of the music generation effectiveness was 7.75, while the average rating of the convenience of using the application was 8.5. While both of these ratings are high, the average rating for the application's ease of use is much higher. This may indicate that the participants generally felt that the effectiveness of using the application was weaker than the convenience and could use much more improvement. This idea was also reinforced in the optional feedback that was provided at the end of the survey, in which the participants generally felt like the ability of the application to generate music based on the user's messages in particular was the weakest point of the application. The data from the previous statement can further strengthen this notion, as it appears that participants who had fewer music pieces generated that successfully reflected the message sentiment tended to rate the effectiveness of the music generation lower.

According to the results of the experiment, the application does fairly well at covering the sentiment of the users' inputted messages in the outputted music, which is to be expected. The survey indicated that the intuitiveness and the effectiveness of the music generation were done well, but the intuitiveness had a significantly higher rating than the music generation's ability to accurately reflect the message's sentiment. This was an expected result, as the user interface had plenty of effort poured into it to ensure that new users of the application would have no trouble operating the application without any overly detailed instructions or external help. However, with the current AI techniques used in the application, it was difficult to get a very

accurate reflection of the message, and participants who were stricter on what was considered "accurate" may not have been as satisfied with the outputted music.

5. RELATED WORK

Louie et al. created a study regarding how music composers would be able to work with artificial intelligence tools in an efficient and organized manner by developing tools that can steer artificial intelligence to do as instructed by the user. For instance, some tools could focus a voice in the music to a specified range of notes. When tested, the tools appeared to have a strong positive effect on composers, as they provided them with a stronger sense of trust, control, and productivity [2]. The work from Louie et al. is similar to this work in that a tool is being developed to help with music creation, but the related work studies examples of how this has been done while this work focuses more on implementing the tool itself and experimenting on it.
In another related work written by Mantaras and Arcos from the Spanish National Research Council, computer music systems that involved artificial intelligence were analyzed based on how the composition and improvisation were handled, as well as how well the music could be performed. The results concluded that a useful technique that could be implemented in future computer music systems is case-based reasoning, as it can directly use information gained from audio samples performed by humans [3]. Both works deal with the topics of artificial intelligence, but Mantaras and Arcos emphasize the implementation of an AI music-generation concept as a whole while this work focuses on the effectiveness of a specific application.

Yang and Chen state that with more digital music available than ever, the organization of music is essential for easily accessing a target song. By organizing the emotions that music conveys, such as happiness or sadness, this problem may be remedied. Therefore, research has been done to tackle the issue of model training and visualization of emotion recognition [4]. Yang and Chen focus more on the various possible methods to go about implementing emotion recognition in music organization. Meanwhile, this work aims to create an application involving sentimental analysis for music generation and test its effectiveness to gauge how well it would perform if it was released to the public.

6. CONCLUSIONS

The current world situation involving a heightened need for quick and new music can be slightly improved through the creation of an application that is designed to generate new music based on whatever sentiment is inputted into the model. By typing a message into the application's text box, the application can determine whether the message's sentiment is positive, negative, or neutral. Then, the application will use this sentiment to output a piece of music for the user to listen to. To test whether this application would be applicable in the real world, an experiment was conducted in which twelve participants were gathered to type in a message into the application, listen to the music that was provided, and the participants also filled out a Google Forms survey that asked how well the music generation was as a whole and how convenient the application was to use. According to the results, the application seems to be fairly effective when it comes to generating music that accurately reflected the emotions of the inputted message [12]. The ratings of both the music generation and the intuitiveness of the application interface were somewhat high. However, because of how limited the current music generation method is in terms of determining only three types of sentiment, the participants took notice that there was still room for improvement with the music generation and gave it a significantly lower rating overall than the convenience of using the application [13]. More effort can be directed towards a music generation system that makes use of more parameters in the future.

One major limitation in the current version of the application is that there is no way for the user to directly download the music after generating it. While the application functions as intended and the user can hear the music from within the application, the user is unable to access that music outside the application easily without the use of external tools. As a result, users cannot share the music on social media or add them to videos without the use of a mobile screen/audio recorder. To improve the convenience of current and future users of the application, one or multiple features could be created to handle such an issue in future updates of the application.

One way to solve such an issue in the future is to create a download button on the same page where the generated music would be. Ideally, it would be located in an easy-to-notice spot with either an icon of the download symbol or the word “Download” on it. Once the user clicks the button, the user can store the audio file on their mobile device, which allows them to use it in other forms of social media easily.

REFERENCES

[1] Schneck, Daniel J., and Dorita S. Berger. The music effect: Music physiology and clinical applications. Jessica Kingsley Publishers, 2005.

[2] Louie, Ryan, et al. "Novice-AI music co-creation via AI-steering tools for deep generative models." Proceedings of the 2020 CHI conference on human factors in computing systems. 2020.

[3] De Mantaras, Ramon Lopez, and Josep Lluis Arcos. "AI and music: From composition to expressive performance." AI magazine 23.3 (2002): 43-43.

[4] Yang, Yi-Hsuan, and Homer H. Chen. "Machine recognition of music emotion: A review." ACM Transactions on Intelligent Systems and Technology (TIST) 3.3 (2012): 1-30.

[5] Nayebi, Aran, and Matt Vitelli. "Gruv: Algorithmic music generation using recurrent neural networks." Course CS224D: Deep Learning for Natural Language Processing (Stanford) (2015): 52.

[6] Wang, S., Li, J., Wang, H., & Zhu, Z. (2021). Emotional Music Generation with Attention-Based Bidirectional Gated Recurrent Unit. IEEE Access, 9, 12978-12986. doi: 10.1109/ACCESS.2021.3052969

[7] Moro, S., Sánchez-Monedero, J., Moreno-Sandoval, A., & Ricarte-Sabater, A. (2020). Music Emotion Recognition for User-Centered Systems: A Review of Challenges and Opportunities. Sensors, 20(6), 1687. doi: 10.3390/s20061687

[8] Liu, Z., Zeng, Q., & Zhang, B. (2020). Music Emotion Recognition with Machine Learning Methods: A Systematic Review. Journal of Intelligent & Fuzzy Systems, 38(3), 3091-3105. doi: 10.3233/JIFS-201483

[9] Liu, C., Lu, Z., Guo, Y., Hu, R., & Du, X. (2020). A Review of AI Music Generation. In 2020 International Conference on Computer Science and Artificial Intelligence (CSAI) (pp. 133-137). IEEE. doi: 10.1109/CSAI49736.2020.9162209

[10] Koops van 't Jagt, R., & Gomez-Marin, A. (2020). Generative models of music: Recent progress and future directions. Current Opinion in Behavioral Sciences, 32, 135-140. doi: 10.1016/j.cobeha.2019.12.002

[11] Zhang, W., & Wu, Y. (2020). Design and Implementation of a Mobile Social Platform Based on Flutter. In 2020 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA) (pp. 599-603). IEEE. doi: 10.1109/ICMTMA49291.2020.00149

[12] Hossain, S., & Amin, M. R. (2020). Sentiment Analysis Based Music Generation: A Review. In 2020 23rd International Conference on Computer and Information Technology (ICCIT) (pp. 1-6). IEEE. https://doi.org/10.1109/ICCIT51738.2020.9342782

[13] Wang, Y., & Su, Y. (2021). Design and Implementation of an Application for Music Generation Based on Sentiment Analysis. In Proceedings of the 2021 3rd International Conference on Artificial Intelligence and Big Data (ICAIBD 2021) (pp. 159-163). Atlantis Press. https://doi.org/10.2991/assehr.k.210611.021

[14] Zhang, W. (2021). Sentiment-based Music Generation using LSTM. GitHub Repository. https://github.com/willzhang05/Music-Generation-using-LSTM

[15] Liu, Qingzhong, Andrew H. Sung, and Mengyu Qiao. "Spectrum steganalysis of WAV audio streams." Machine Learning and Data Mining in Pattern Recognition: 6th International Conference, MLDM 2009, Leipzig, Germany, July 23-25, 2009. Proceedings 6. Springer Berlin Heidelberg, 2009.

AIRBNB RESEARCH: AN ANALYSIS IN NEXUS BETWEEN VISUAL DESCRIPTION AND PRODUCT RATING

Chun Kit Fu[1], Yu Sun[2]

[1]7 Lakes high school, 9251 S Fry Rd, Katy, TX 77494
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Hosts are often desperate to find ways to rent their house, However, most of them do not have possess the knowledge of knowing what type of image cover would grasp the attention of their customer. Gilded by these needs, I have designed an application that uses machine learning to find the relationship between the images and their rating [1]. I first used JSON to convert the HTML file resource to a format where we can use in python for web scraping [2]. This paper designs an application tool to find all the object or characters inside images by web scraping and changes it into a model for machine learning [3]. Applied our application to predict the rating and conducted a qualitative evaluation of the approach. In order to prove our result, I imported an image from Airbnb and found its rating. It turns out that the predicted rating is extremely close to the real rating, Proving The system's usability.

KEYWORDS

Web scraping, Machine learning, Airbnb

1. INTRODUCTION

Rental service, or homestay service, is a common sight these days. Its history are so old that the origin is probably lost in prehistory. In medieval times, most land was owned either by the King or by lords, and almost all farmers were tenant farmers who paid a rent - usually a percentage or portion of crops grown - in return for living on and farming the land with. Nowadays , rental service is a popular form of hospitality and lodging, whereby visitors share a residence with a local of the area to which they are traveling.

Airbnb is a vacation rental company that operates an online marketplace focused on short-term homestays and experiences [4]. Its features compared with its competitor include Reservation screening,$3M damage protection, Pet damage, 24-hour safety line and much more. On the app, the host could offer to rent their housing by uploading information of their house in terms of images, price, location, etc. However, there are still many hosts who are still having a hard time determining what they need to grasp the attention of their customers. According to RMA, Rental Market Analysis, images play a major role in what customers think of the housing [5]. In the hopes of helping hosts in Airbnb find the best image, we have created a program called hbnb to find the relationship between the attributes inside the images and its rating by using web scraping

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 229-238, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130520

and machine learning [6]. This program was designed to help many real estate agents and hosts to find the images that will produce the highest rating, so it would save many host's time.

Some techniques and systems proposed by previous methods such as regression lines, hedonic models, or time-series methods all allow users to predict rich attributes in real estate images by using their own unique way [7]. However, they all have their own limitations in certain field and are inconvenient in some ways. Regression lines assume that the price is a weighted sum of property characteristics, and are unable to address non-linearity or detect outliers. Hedonic models, or hedonic regression, is a revealed preference method for estimating demand or value [8]. However, it's rather restrictive, imposing uniformity of coefficients across both space and time. They both a lack of datasets for predicting rich attributes such as landscaping, restroom, ceilings, hardwood floors, fireplaces, etc., Time series method on the other hand is a really handy tool for forecasting purposes since it's high in accuracy and simplicity. However, it also requires more skill than regression analysis since model needs to be adapted according to the historical database. It has been proven that it's only efficient when inaccessible, meaning that if a model has been built on historical data, it cannot be used to predict future values or trends because no one can guarantee that the historical data will remain the same as time passes. However, in our program, we used a more variety of what's inside the picture to find the connection between image and rating, but not as heavy(overfit) as time-series, so it would take less time. The JSON value we extracted for web scraping includes the column labels that finds the furniture, and then we save the module into a file named 'aribnb_model.sav' by defining a function called save_module [9]. This process that save the model for future use save much more time than most method and are less likely to overfit. In addition, the image can change adjust base on the need of the host or real estate, so it won't be so reliant on historical data like time-series method in order to construct the models.

In this paper, we follow the same line of research by the use of machine learning and web scraping. Our goal is to successfully predict the outcome and identify the similarity in the images of high rating. Therefore, the feature included in this project are web scrapping and prediction of rating. In the Web scrapping feature, the user just needs to provide a URL to the software, and it will automatically extract the JSON value that contains the image attribute. With those resources, the program will be able to download the corresponding image from the internet. In this case, the user does not need to load the image into the software manually, so it would reserve much time for the user. The rating prediction feature on the other side can be used to find the rating of a house given the provided image URL and ID. Our methods are inspired by the IRIS flower method, so we used lots of computational methods and class methods to operate our project. There are many good methods and feature that build up the project. For the Method, we First used a method like Univariate Selection by using Scikit-learn [10]. In addition, we used the pandas method to build up the data frame.

In two application scenarios, we demonstrate how the above combination of techniques increases the accuracy of rating by inputting another ID along with its image to compare the output with its rating. First, we show the usefulness of our approach by a comprehensive case study on the data and rating of airbnb. Second, We analyze the relationship between the attribute inside the image and the rating of the airbnb. Then we applied machine learning to produce a table containing an x variable from the input data frame to count the picture attribute as the input data and a y variable as the rating for the output variable. Lastly, to prove that our project worked successfully, We used photos from Airbnb to forecast each listing's rating, and then compared those predictions to the listings' actual ratings to demonstrate our accuracy. We repeat this process a total of 5 times for analysis purpose and proved that the percent error between the predicted rating from our model and the actual rating from airbnb are extremely close. This result shed light on the conclusion that the project has worked well and proved our result to be accurate.

The paper is organized as the followed, we put the abstract the in front of all sections. Section 1 gives the introduction to what we are doing (background, benefits, etc.), what previous method exist, what feature is within our application and how we proved our results. Section 2 gives the details on the challenges that we met during the experiment and designing the sample along with how we conquered them; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2 and what is the component of our application; 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. Libraries

A Python library is a reusable chunk of code that you may want to include in your programs in order to save time. Unlike other coding languages like C++, Python libraries do not pertain to any specific context in Python. A 'library' is just a loose description of a collection of core modules. In this project we used many libraries like BeautifulSoup, JSON, Image from PIL, torch, glob, pandas, requests, OS, wget, and pickle to save time and perform the function we needed. To find all the Libraries, it took quite a bit of time to find all the features of the libraries and what we needed each library to perform. In addition, when importing is not enough, we took a while to understand that we needed to install some libraries before importing it like wget, which was not usable until we installed it.

2.2. Web Scraping

JSON is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays. In order to use machine learning to achieve our goal, we had to first use web scraping to extract the right JSON value that contains the information of the items in the webpage to make into readable text [11]. It was a difficult task since we had quite a bit of trouble understanding what sections in the JSON value are the sections that need to be extracted and what's the best pre-build model (python library) to use. JSON value contains a huge data set and are The two primary, made up of keys and values. Together they make a key/value pair, and finding the right information is quite difficult. In addition, converting the value into a table is also quite troublesome as it involves the use of libraries and changing it into a readable data file and organization. in the end, we decided to used BeautifulSoup to get the content (the whole HTML file) in airbnb_url in the form of parser(html.parser) to locate the image resource in the JSON.

2.3. Machine Learn

Machine Learning is the instrumental tool that we use to achieve our goal, it can be used to analyze large datasets and conduct model selection in the context of causal inference. After using the beautiful soup method to extract the JSON value to complete web scraping and creating an input data using, we tried many ways to fit the input and output. For machine learning, our input data is the data frame created for the count item function, while the output data is the rating of the images. The job of machine learning is to connect the variety of items inside the image that we downloaded from the web scraping with its rating and predict what is the rating if we input an image inside. It was a difficult task since it requires meticulous attention to optimize an

algorithm and debugging machine learning algorithms is difficult because the code includes multiple dimensions where information can be incorrect.

3. SOLUTION

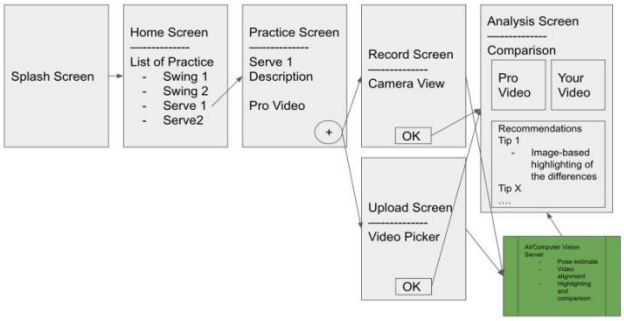


Figure 1. Overview of the solution

For this project, its application can be separated the application into 5 components, the raw data, web scraping, counting items, creating input dataset and machine learning

The raw data is the most fundamental building block of our program. We got these data from a rental company name airbnb directly. We used airbnb's housing information to create a big list of all places for future machine learning by adding the ID, URL and rating of the image. The idea was to use machine learning to connect the attribute inside the URL of the picture along with it's rating.

```
PLACES = [{'id': "45054521", 'url':"https://www.airbnb.com/rooms/45054521?adults=1&category_tag=Tag%3A8099&children=0&infants=0&search_mode=flex_destinations_search&check_in=2022-08-1
,{'id': "48511845",'url': "https://www.airbnb.com/rooms/48511845?adults=1&category_tag=Tag%3A8099&children=0&infants=0&search_mode=flex_destinations_search&check_in=2023-02-
{'id': "53060518",'url':"https://www Follow link (ctrl + click) )60518?adults=1&category_tag=Tag%3A8225&children=0&infants=0&search_mode=flex_destinations_search&check_in=2022-08-0
 {'id': "569739029479086311",'url':"https://www.airbnb.com/rooms/569739029479086311?adults=1&category_tag=Tag%3A8099&children=0&infants=0&search_mode=flex_destinations_searc
{'id': "569739029479086311",'url':"https://www.airbnb.com/rooms/569739029479086311?adults=1&category_tag=Tag%3A8099&children=0&infants=0&search_mode=flex_destinations_search
{'id':"3301885",'url':"https://www.airbnb.com/rooms/3301885?adults=1&category_tag=Tag%3A8099&children=0&infants=0&search_mode=flex_destinations_search&check_in=2022-08-13&ch
```

Figure 2. URL

For the web scraping portion, it is also very important and well connect to the raw data section. This part of the code is used to find the attribute inside a URL of the raw data set. To do web scraping, we fetched the URL from the webpages inside the raw dataset (PLACES) by using a python library called BeautifulSoup to get the content (the whole HTML file) in airbnb_url in the form of parser(html.parser) to locate the image resource in the JSON value inside the page source to extract that specific section of JSON to return the URL of the images. Next, we download the image from the URL that we got from the previous step. Each place (web page) will have its own folder for images, and the folder name will be the 'id' of these places. Lastly, we will fetch all Places by repeating the previous 2 steps in a loop for every element in our raw dataset to ensure that we have the folder for each of the webpage.

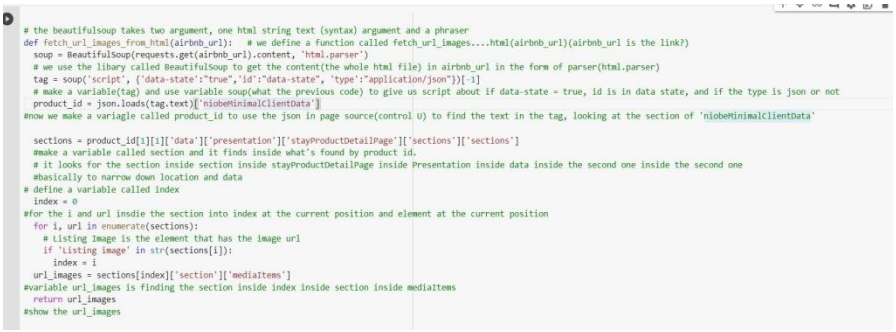


Figure 3. Webpage 1

The count item section is rather simple compared with the rest, we imported some pre-build model (python libraries) like Yolov5 and glob, to complete our objective. We used yolov5 to count each different items in an image by dividing images into a grid system to count the input dataset and glob to return all file paths that match a specific pattern.

Figure 4. Webpage 2

Figure 5. Webpage 3

However, before we start counting, we would have to create an input dataset in order for the program to know what to look for. We created a function name get_training_dataset specifically for this by using a python libraries name pandas (short for pd).The Pandas library is an open source Python package that is most widely used for data science/data analysis and machine learning tasks, in this scenario, it made a perfect model to get the items and quantities we need into a data frame to let the application know the items it needs to count

Lastly, after completing all the previous steps, the final and most important part is to connect them using machine learning. We created a training dataset by using the input x value (data frame from the previous step) and the output y value (the rating value from the raw dataset (Places) to build a model to save from future use and use the model to predict rating from new PLACES that were not included in the raw dataset.

```
[ ] # output
    Y = df['rating']  #assign the rating into the y, y is equal to the rating portion of df
    # Y(output) is the data frame with the column of rating
    # input
    X = df.drop(columns=['rating'], axis = 1)  #we are assigning data to x, x should not contain the y(rating) so we are deleting the column
    print(X)
    #axis can only be 0 or 1
    # x right here is the drop the column called rating(we need to separate the rating so we used rop)
    #axis indicate the area we are deleting, if it is 1, it should the whole column, if it is 0, it delete the whole row)

[ ] print(X.shape)  # x is data frame, y is also data frame, but x is data frame without the column of rating  (6,19), 6 is the column, 19 is the row

[ ] print(Y.shape)

Train model

+ Code  + Text

[ ] from sklearn.model_selection import train_test_split
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.linear_model import LinearRegression
    #Linear Regression is the way we link our x and y (think of math linear function)
    # create a RF classifier
    airbnb_model = LinearRegression()
    X_train, X_text, y_train, y_test = train_test_split(X, Y, test_size = 0.2, random_state = 42)
    # first two - always representing training input and test input (the name does NOT matter )
    # test size is how much portion we use(percentage) as the whole data
    # random state is how the split work, 42 is guaretten to be the same grouping everytime
```

Figure 6. Webpage 4

4. EXPERIMENT

4.1. Experiment 1

In this project, we utilized machine learning to create a table with a y variable as the rating for the output variable and an x variable from the input data frame to count the picture attribute as the input data. We used photos from Airbnb to forecast each listing's rating, and then compared those predictions to the listings' actual ratings to demonstrate our accuracy. We forecast the rating of the existing airbnb using photographs to support our claim. Then, we repeat the process four more times with different datasets (image). For analysis, we take the predicted values and their corresponding actual values from Airbnb. To compare them and determine the correctness of our mode, we determined the percent inaccuracy. Some Accuracy problem that sill may occur in these predictions because the data sample for machine learning is rather small.

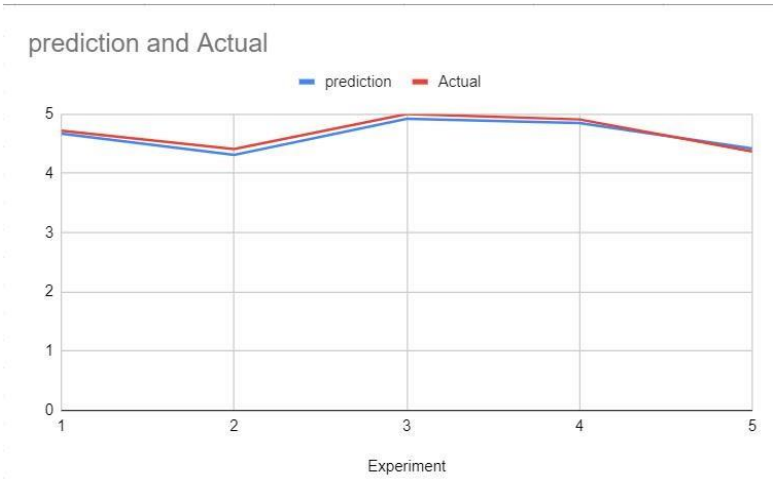


Figure 7. Prediction and Actual of experiment 1

	person	bed	chair	toilet	couch	...	sink	oven	microwave	bird	bus
0	1.0	3.0	11	1.0	2.0	...	NaN	NaN	NaN	NaN	NaN
1	NaN	5.0	5	1.0	NaN	...	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	5	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
3	NaN	1.0	1	1.0	NaN	...	1.0	NaN	NaN	NaN	NaN
4	NaN	1.0	1	1.0	NaN	...	1.0	NaN	NaN	NaN	NaN

5 rows × 20 columns

Figure 8. Table of amounts

```
{'person': 1, 'bed': 3, 'chair': 11, 'toilet': 1, 'couch': 2, 'bowl': 1, 'bench': 1, 'tv': 1}
{'potted plant': 5, 'chair': 5, 'bed': 5, 'refrigerator': 1, 'bench': 3, 'bottle': 2, 'toilet': 1}
{'chair': 5, 'bowl': 2}
{'potted plant': 11, 'bowl': 3, 'tv': 1, 'chair': 1, 'train': 2, 'dining table': 1, 'bench': 1, 'bed': 1, 'sink': 1, 'bottle': 2, 'toilet': 1
{'potted plant': 11, 'bowl': 3, 'tv': 1, 'chair': 1, 'train': 2, 'dining table': 1, 'bench': 1, 'bed': 1, 'sink': 1, 'bottle': 2, 'toilet': 1
{'refrigerator': 1, 'oven': 1, 'chair': 2, 'microwave': 1, 'bird': 1, 'bench': 1, 'bed': 2, 'couch': 4, 'bus': 1, 'person': 3}
```

Figure 9. Code of amounts

Our first hypothesis is that the percentage error of the predicted value from our model and actual value from the Airbnb page will within 5 percent. To prove our hypothesis, we use images from the existing airbnb and predict its rating. Next, We perform the same procedure 4 more times using different datasets (image). We take the prediction values, and it's corresponding actual value in airbnb for analysis. We calculated the percent error to contrast them to see the accuracy of our model. This process is completed by using 5 PLACES (experiment image set) from airbnb using its ID and URL. The average percent error turns out to be 1.456 percent, which indicate that the project has worked well.

4.2. Experiment 2

For this experiment, we used machine learning to creating a table with x variable from the input data frame to count the image attribute as the input data and y variable as rating for output variable. To prove our accuracy, we used images from airbnb to predict it's rating to see if it's similar to its actual rating in airbnb, However Accuracy maybe a major issue since the dataset is so small and the result proves it quite well.

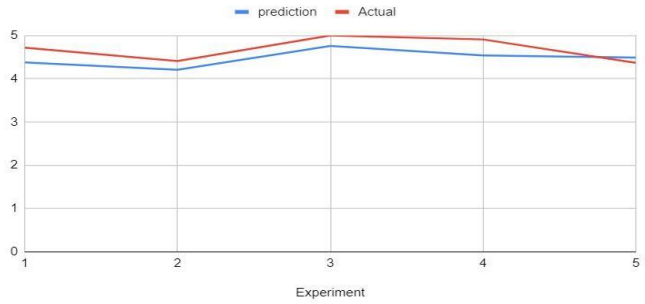


Figure 10. Prediction and Actual of experiment 2

My prediction is that there will be a sizable percentage difference between the anticipated number from our algorithm and the real figure from the Airbnb page. We forecast the rating of

the existing airbnb using photographs to support our claim. Then, we repeat the process four more times with different datasets (image). For analysis, we take the predicted values and their corresponding actual values from Airbnb. To compare them and assess the quality of our model, we determined the percent error. The 5 PLACES (experiment image collection) from airbnb is used to finish this process using its ID and URL. The project did not work too well because the average percent error is above 15%.

The experiment proved that all the project's components, including the challenges like python libraries, web scraping, and machine learning, worked successfully, therefore it answered all the issues. The percent error between the projected rating and the actual rating from Airbnb in the examination of 50 sample machine learning data states that the percent error between the predicted rating using the model and the actual rating from airbnb are less than 1.5 percent. The initial objective, or assumption, was a 5% error rate between these two. These experiment findings thus demonstrate that all problems have been resolved and go above and beyond my expectations.

5. RELATED WORK

David Koch, Miroslav Despotovic, Sascha Leiber, Muntaha Sakeena, Mario Döller and Matthias Zeppelzauer of the University of applied science presents Real Estate Image Analysis by the use of real estate applications [12]. Compare to our work, these people used different method and contains feature like classification and others. Although we share some similar feature like object detection, they have a bigger data set and many more feature like 3D reconstruction, classification, and image registration. However, since they have so many features and other consideration, their model run rather slowly compare to ours.

Nick Desmond presents the Predicting Airbnb Review Scores, in his work, he tries to ascertain an adequate customer satisfaction metric and understand what Airbnb hosts can do to improve the customer experience [13]. He used methods like numpy to make function convert Airbnb listing data to integers and Running OLS regression on a 70/30 train/test split of the data yields a very impressive R2 score of 0.95. Regression analysis like these performs exceptionally well for linearly separable data and Easier to implement, interpret and efficient to train. However, these types of methods are usually prone to overfitting and have problems with assumption of linearity between dependent and independent variables. Our project on the other hand are much more defined to find the linearity between dependent and independent variables.

Shunyuan Zhang,Dokyun Lee, Param Vir Singh, And Kannan Srinivasan presents the work to find What Makes a good Image [14]? They studied how Airbnb property demand changed after the acquisition of verified images (taken by Airbnb's photographers) and explore what makes a good image for an Airbnb property. They used method like deep learning and difference-indifference analyses on an Airbnb panel for their work and has a dataset spanning 7423 properties. Furthermore, they have a total of 12 human-interpretable image attributes that pertain to three artistic aspects—composition, color, and the figure-ground relationship as they find systematic differences between the verified and unverified images. Their result are applicable to any photographers who wish to optimize their image. Compare to our work, they have many more dataset and aspect to put into consider for optimization but are comparatively slow to give the result, which also indicate they have a presumably higher accuracy compare to us.

6. CONCLUSIONS

What we have done is proposing an application to predict the rating of a house by the attribute inside it's image by first collecting the raw data set from a house rental company name airbnb that contains the ID, URL, and the rating of the house. Then we Fetch the URL from the webpage by locating the image resource in the JSON and extracting that section of the JSON value out to return the URL of the image. Afterward, we downloaded the image from the URL that we extracted from the previous step to created a folder so that each place(web page) will have its own folder for image. To ensure that we got all the places, we keep repeating the last 2 steps in a loop for every element in our raw dataset. Next up, we imported the pre-build model Yolov5 to count different items of the image in the imputed data set that we have made into a data frame [15]. Lastly, use machine learning to train data of the input value x(data frame) and the output value y which is the "rating" value from the raw data set. We build a model for future use and apply the application to experiment by predicting rating from a new "place" other than the raw data set. The experiment results indicate its effectiveness and solve challenges because it successfully found all the input value and the rating was extremely close to the actual rating in the airbnb.

There are still quite a few limitations in our application, first of all, the data is considerably small, so the accuracy might not be enough to cover all, secondly all data used for machine learning are from airbnb, although Airbnb is a considerably large website, it might still affect the accuracy, in addition, the current version of our project does not consider location and outdoor view, this might also affect practicality of some user. In the future optimization, we plan on adding more variables like how many customers selected the price rate, how far away is the house, or how many people rated and percent of how many people like it.

In the future, I plan on adding a lot more database to improve the accuracy of the project, another limitation I will solve is the renter's location, I plan in the future to extract the renter's location JSON out of the page source so that it could also be placed in the machine learning.

REFERENCES

[1] Jordan, Michael I., and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." Science 349.6245 (2015): 255-260.
[2] Pezoa, Felipe, et al. "Foundations of JSON schema." Proceedings of the 25th international conference on World Wide Web. 2016.
[3] Glez-Peña, Daniel, et al. "Web scraping technologies in an API world." Briefings in bioinformatics 15.5 (2014): 788-797.
[4] Guttentag, Daniel. "Progress on Airbnb: a literature review." Journal of Hospitality and Tourism Technology 10.4 (2019): 814-844.
[5] Ayouba, Kassoum, et al. "Does Airbnb disrupt the private rental market? An empirical analysis for French cities." International Regional Science Review 43.1-2 (2020): 76-104.
[6] Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9 (2020): 381-386.
[7] Ekeland, Ivar, James J. Heckman, and Lars Nesheim. "Identifying hedonic models." American Economic Review 92.2 (2002): 304-309.
[8] Ekeland, Ivar, James J. Heckman, and Lars Nesheim. "Identification and estimation of hedonic models." Journal of political economy 112.S1 (2004): S60-S109.
[9] Nurseitov, Nurzhan, et al. "Comparison of JSON and XML data interchange formats: a case study." Caine 9 (2009): 157-162.
[10] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.

[11] Stevenson, Julie S., Gordon C. Bruner, and Anand Kumar. "Webpage background and viewer attitudes." Journal of advertising research 40.1-2 (2000): 29-34.
[12] Koch, David, et al. "Real estate image analysis: a literature review." Journal of Real Estate Literature 27.2 (2019): 269-300.
[13] Luo, Yuanhang, Xuanyu Zhou, and Yulian Zhou. "Predicting airbnb listing price across different cities." (2019).
[14] Zhang, Shunyuan, et al. "What makes a good image? Airbnb demand analytics leveraging interpretable image features." Management Science 68.8 (2022): 5644-5666.
[15] Zhu, Xingkui, et al. "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

NON-FUNGIBLE TOKEN BUBBLE PREDICTION USING EXTENDED LOG- PERIODIC POWER LAW MODEL

Ikkou Okubo[1],Kensuke Ito[2], Kyohei Shibano[2] and Gento Mogi[2]

[1]Department of Technology Management for Innovation, The University of Tokyo, Tokyo, Japan
[2]Endowed Chair for Blockchain Innovation, The University of Tokyo, Tokyo, Japan

ABSTRACT

Non-fungible token (NFT) bubbles are a problematic issue, and this study aims to predict NFT bubbles using an extended log-periodic power law singularity (LPPLS) model. The classic LPPLS model targets the endogenous nature of bubbles caused by the mimetic behavior of investors without external influences; however, the extended model attempts to incorporate exogenous influences. First, we compare the performance of the two models for NFT price prediction. The exogeneous variable in the extended model is cryptocurrency volatility. Then, we calculate the bubble confidence using both models. The results show that the explanatory power and forecasting accuracy of the extended model are superior in all projects. We also find that the bubble confidence indicator reinforces the results of bubble prediction.

KEYWORDS

Bubble, Cryptocurrency, Log-periodic power law, NFT

1. INTRODUCTION

The prediction of non-fungible token (NFT) bubbles is an industry-wide problem. Despite the strong growth of the NFT market, bubbles—faster-than-exponential price increases—have been causing crashes(see Section 3.1.3). All stakeholders (e.g., buyers, creators, and platformers) have been thrown into disarray. For buyers, prices are highly volatile, and the risk of substantial loss in a short time is high. As noted by Kong and Lin [1], the geometric mean of monthly returns for NFTs is 13.92% so far, whereas for stocks, bonds, and gold, they are 1.01, −0.61, and 0.63%, respectively. Meanwhile, the standard deviation of returns for NFTs is the highest (65.57%): approximately 14 times greater than that of stock returns. For NFT creators, bubbles are a disincentive to improving NFT quality. Hence, the mass production of low-quality NFTs for speculative purposes dominates creator motivations. On OpenSea, the world's largest NFT marketplace, more than 80% of new NFTs generated using the free mint function were found to be fraudulent (i.e., plagiarized, spam, or fake), including many copyright violations [2]. Platformers also experience significant negative impacts from price fluctuations. According to DappRadar [3], OpenSea saw a 99% decline in trading volume in just four months in 2022. Obviously, there is a domain-wide demand for bubble prediction. This study responds to this societal demand.

Notably, there are no current studies on NFT bubble prediction that consider both endogeneity and exogeneity. In the stock market, one line of research focuses on the endogenous and

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 239-255, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130521

exogenous nature of bubbles [4], [5], [6], endogeneity refers to a dramatic change in prices in one direction caused by mimetic investor behaviors without external shocks [7]. On the other hand, exogeneity refers to price changes caused by external shocks, such as the Nazi invasion of Western Europe in 1940 and the COVID-19 pandemic [7]. For NFTs, a few studies [8], [9] have focused on endogenous bubbles. However, none have considered both endogeneity and exogeneity [5], [10]. To answer this academic requirement is the aim of this study and its novelty.

To answer the call for a combined model, this study answers two hypotheses:

(1) Extending an existing model that focuses on endogenous bubbles to incorporate external variables improves the model's explanatory power and forecasting accuracy.
(2) The extended model reinforces the results of bubble prediction.

To support these hypotheses, we use an extended log-periodic power law singularity (LPPLS) model that assumes rational expectations and investors who behave in irrational and mimetic manners. The mechanism by which the price of an asset rises rapidly and crashes due to the collective mimetic behaviors of investors is thus examined. Although the original LPPLS model targets the endogenous nature of bubbles, the extended model includes exogenous influences. The extended approach consists of three steps:

(1) Using the concept of drawdown [10], we select NFT projects for LPPLS analysis.
(2) We compare the performance of traditional vs. extended LPPLS models on NFT bubble prediction. The exogeneous variable is *cryptocurrency volatility*.
(3) We calculate the LPPLS bubble confidence indicator [11] to appraise the potential for nearterm price decreases or increases based on price data from which external influences are eliminated. We then compare the indicators of both models.

The results show that the explanatory power and forecasting accuracy of the extended model increase when external variables are considered. However, the explanatory power of the original model varies from project to project. We also find that a type-2 error in predicting the bubble risk may have occurred. That is, the exogenous factors may have offset the endogenous changes, causing the fluctuations to lose their bubble characteristics.
The rest of the paper is organized as follows. Section 2 discusses previous studies on the LPPLS model, Section 3 explains the drawdown construct, the LPPLS model, its extended version, and bubble confidence indicators, and Section 4 describes the data used for the analysis. Section 5 then discusses the implications of the results, and Section 6 concludes this study.

2. RELATED WORKS

Sornette [12] was the first to apply LPPLS to financial markets, and its success has been tracked by many scholars in different markets. Gonçalves *et al*. [13] used LPPLS to analyze the Portuguese stock market. Johansen [14] did so for the US stock markets, and Indiran *et al*. [15] did so for Malaysia. Separately, Takagi [16] detected numerous bubbles forming in memetic stocks in the US, but he had difficulty predicting social media-induced exogenous rallies, as LPPLS is meant to detect only endogenous bubbles.

In recent years, LPPLS has been applied to cryptocurrency and NFT markets. For cryptocurrency, Shu and Zhu [17] proposed an adaptive multilevel time-series LPPLS detection method to analyze a finer-than-daily timescale for Bitcoin (BTC) price data. Geuder *et al*. [18] revealed the existence of some frequent bubble periods in BTC prices. Regarding NFTs, Ito *et al*. [8] applied LPPLS to the time-series price data of major NFT projects and detected bubbles with

a discernable bubble confidence indicator. Wang *et al*. [9] used LPPLS for robustness testing in the detection of NFT bubbles using supremum augmented Dickey–Fuller (SADF) and generalized SADF tests.
As can be inferred from the extension of the model in conventional markets, both endogeneity and exogeneity should be grasped in NFT markets. However, no studies have thus far made this attempt.

3. METHODOLOGY

In this section, we describe the drawdown concept[10], which is used to select NFT projects to which we apply the LPPLS model. Next, we explain the LPPLS model and how it is extended to incorporate external variables. Finally, we explain how to calculate the bubble confidence indicator.

3.1. Drawdown for Capturing Downward Price Trends

In this study, we apply the drawdown concept when selecting the NFT projects to be analyzed with LPPLS. Drawdown is needed because the target data for the LPPLS model must be carefully selected to ensure that the time range does not include a single crash. Otherwise, the LPPLS model will not fit the data after the crash. Furthermore, a key LPPLS parameter is the *date of highest crash probability*. Hence, if there are multiple crashes in the range, the data will fail to set a unique parameter. Therefore, it is necessary to remove data containing extant crashes.

In this section, we describe the original and coarse-grained types of drawdowns. We then explain how to identify crashes from ordinary drawdowns. The variables in this section are listed in Table 1.

Table 1. Drawdown variables.

Variable	Explanation
$P_{\min}$	Local minimum price
$P_{\max}$	Local maximum price
ε	Threshold of drawdown being smaller than coarse-grained
σ_d	Volatility used to set ε
N_d	Number of data
r	Log price return
$E[*]$	Expected value of *
$p(t)$	Price at a given period t
R(Drawdown)	rank number of Drawdown

3.1.1. Original Drawdown for the Detail Trend

Drawdown and drawup concepts were introduced by Johansen and Sornette [10] to capture trends in price fluctuations, as shown in Figure 1. Drawdown is the percentage of change from one local maximum to the next local minimum price. Thus, as long as prices continue to fall, it is considered a drawdown. A drawup, in contrast, is the percentage change in price from one local minimum to the next local maximum. Both are given, respectively, as follows:

$$\text{Drawdown} = \frac{P_{\min} - P_{\max}}{P_{max}}, \quad (1)$$

$$\text{Drawup} = \frac{P_{\max} - P_{\min}}{P_{\min}}. \quad (2)$$

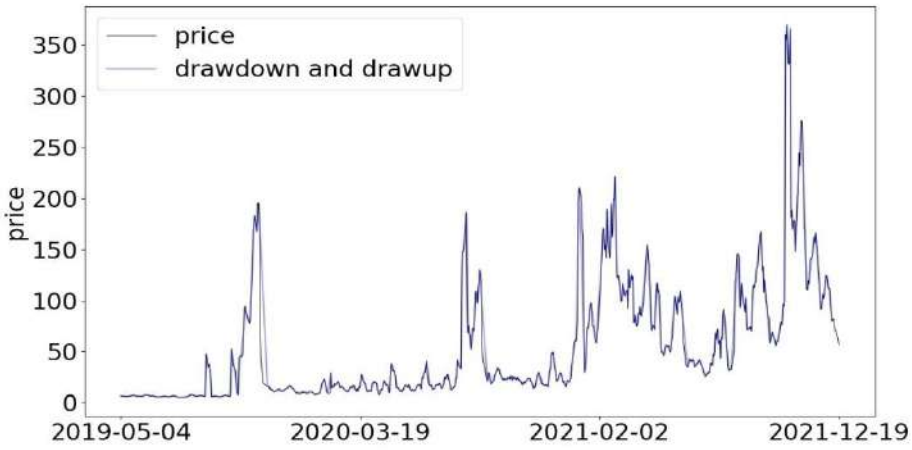


Figure 1. Drawdown using Ethereum Name Service data series

3.1.2. Coarse-Grained Drawdown for Macroscopic Trends

A coarse-grained drawdown is used to capture downward price trends based on the "big picture." A sequence of price declines for which a drawdown is calculated can be terminated by any small price increase. Generally, price fluctuations contain a great deal of noise. Therefore, the original drawdown concept may not adequately capture downward price trends.

Therefore, we introduce a threshold epsilon such that the coarse-grained drawdown reflects the rate of change from the local maximum to the local minimum, where the minimum is the point at which the first drawup larger than ε occurs after the local maximum. The local maximum is the point at which the first drawdown smaller than -ε occurs after the local minimum. ε is obtained by using the following σ_d:

$$\sigma_d = \sqrt{\frac{\sum_{i=1}^{N_d}(r_{i+1} - E[r])^2}{N_d}}, \quad (3)$$

where

$$r_{i+1} = \log p(t_{i+1}) - \log p(t_i). \quad (4)$$

In this study, we set $\sigma_d/4$, $\sigma_d/2$, and σ_d as the values for ε based on previous studies [19].

3.1.3.Crash as an Outlier

Drawdowns large enough to be called "crashes" must be distinguished statistically from other relatively small or ordinary drawdowns. Johansen and Sornette [10] analyzed past markets and

showed that nearly all drawdowns were well-fitted based on a stretched exponential function. However, some drawdowns that are not well-fitted are considered outliers. The stretched exponential function is given as

$$\mathrm{R(Drawdown)} = \mathrm{A_d exp(-b|Drawdown|^z)}, \quad (5)$$

where A_d is a constant and $b = \lambda^{-z}$.

Per Johansen and Sornette [10], by taking the logarithm of (5) for convenient and efficient fitting, the following equation is obtained:

$$\log \mathrm{R(Drawdown)} = \log \mathrm{A_z} - \mathrm{b|Drawdown|^z}. \quad (6)$$

This model is calibrated for ordinary least squares (OLS) applications using the dogbox algorithm [20]. In this study, an outlier is a drawdown value that deviates from the fit of (6) in the 99.5% tail of the distribution.

3.2. LPPLS Model

The LPPLS model [12], [20], [21] is used to detect and predict bubbles (i.e., faster-than exponential increases in asset prices). We use the LPPLS extension for bubble prediction, which targets the average NFT prices of all projects. Although each NFT has a unique price, we treat them as homogeneous. The variables used for this work are shown in Table 2.

Table 2. LPPLS model variables.

Variable	Explanation
$p(t)$	Price at a given period, t
A	Price at t_c
B	Power law acceleration amplitude
C	Log-periodic oscillation amplitude
D'	Parameter of $\int_{t_0}^{t} r(\tau) + \sigma(\tau)\varphi(\tau)d\tau$
t_c	Most probable time at which the bubble ends
m	Superexponential acceleration degree
ϕ	Oscillation period
$r(t)$	Interest rate at a given period, t
$\sigma(t,n)$	Historical volatility at t for n days
n	Number of volatile days
$\varphi(t)$	Market price of the stochastic discount factor risk at t
ω	Oscillation frequency
$C1$	$C\ cos\ \phi$
$C2$	$C\ sin\ \phi$
D	Parameter of $v(t)$
p_i	Price on the day i of a given period
$p(t)$	Average price over n days

3.2.1. Original LPPLS Model

The LPPLS model [12], [20], [21] is written in a suitable form for fitting time-series data:

$$\ln[p(t)] \approx A + B|t_c - t|^m + C|t_c - t|^m \cos(\omega \ln|t_c - t| - \phi) + D' \int_{t_0}^{t} r(\tau) + \sigma(\tau, n)\varphi(\tau)d\tau, \quad (7)$$

where $A = \ln[p(t_c)]$, t_c denotes the critical time at which the bubble is likely to end, m is the degree of super exponential acceleration, ω is the frequency of the oscillation, and ϕ is its period.

Assuming that $r(\tau) = \varphi(\tau) = 0$, the original LPPLS model is obtained as

$$\ln[p(t)] \approx A + Bf(t) + Cg(t), \quad (8)$$

where

$$f(t) = |t_c - t|^m, \quad (9)$$

$$g(t) = |t_c - t|^m \cos(\omega \ln|t_c - t| - \phi). \quad (10)$$

Using the method of Filimonov and Sornette [21], the nonlinear parameter, ϕ, can be eliminated to obtain the following equation:

$$\ln[p(t)] \approx A + Bf(t) + C_1 g_1(t) + C_2 g_2(t), \quad (11)$$

Where

$$g_1(t) = |t_c - t|^m \cos(\omega \ln|t_c - t|), \quad (12)$$

$$g_2(t) = |t_c - t|^m \sin(\omega \ln|t_c - t|). \quad (13)$$

3.2.2. Extended LPPLS Model

To extend the original model so that external effects can be considered, we loosen the assumption that $r(\tau)$ and $\varphi(\tau)$ are equal to zero. Although both $r(\tau)$ and $\varphi(\tau)$ equal zero in the original model, they are not zero in reality. As performed by Zhou and Sornette [5], we assume that $\varphi(\tau)$ is a constant φ, and we employ the historical volatility of a specified asset as a proxy for the volatility factor, $\sigma(\tau)$. Note that we still assume that the interest rate is zero for simplification. However, please note that Hu and Li [6] used real values (i.e., the risk-free interest rate and deposit reserve rate in China).
Therefore, we obtain

$$\ln[p(t)] \approx A + Bf(t) + C_1 g_1(t) + C_2 g_2(t) + Dv(t), \quad (14)$$

where

$$D = D'\varphi, \quad (15)$$

$$v(t) = \int_{t_0}^{t} \sigma(\tau, n)d\tau, \quad (16)$$

$$\sigma(t, n) = \sqrt{\frac{\sum_{t=1}^{n}(p_t - \bar{p})}{n-1}}. \quad (17)$$

Following Zhou and Soenette [5], we use the trapezoid scheme to integrate $\sigma(\tau, n)$ as follows:

$$v(t) = \sum_{\tau=t_0+1}^{t} \frac{[\sigma(\tau-1,\mathrm{n}) + \sigma(\tau,n)]}{2}. \qquad (18)$$

As mentioned, the original LPPLS model cannot account for external impacts. Hence, we propose this extended model based on the cryptocurrency's volatility.

Although studies [22] and [23] showed that volatility transmission effects between cryptocurrencies and NFTs are limited, Ante [24] revealed that a BTC price shock caused an increase in NFT sales. Furthermore, between BTC and Ethereum (ETH), a bidirectional relationship between returns and long-term spillovers was found.

Therefore, it is advisable to use cryptocurrency volatilities as the proxy for the volatility factor, $\sigma(\tau,\ n)$. Specifically, we use the historical volatilities for 7, 30, and 90 days of each BTC and ETH blockchain and compare the models while incorporating each external variable (see Section 3.2.7 for model evaluation details).

3.2.3.Calibration

Both LPPLS models can be calibrated using OLS to minimize the sum of squared residuals using a modified Python module [25].

3.2.4.Evaluation

To compare the original and extended LPPLS models using each explanatory variable, we apply the Akaike information criterion (AIC)[26]and adopt the model with the minimum AIC value. As performed by Zhou and Sornette [5], AIC is calculated by fitting models (11) and (14) with each external factor (see Section 4). We also calculate the adjusted R^2 from (11) and (14) to compare the models' explanatory power. Finally, we test the significance of each external explanatory factor by fitting each model (14) and calculating the *p*-values of the external factor coefficients.

Although these tests require that residual errors be i.i.d. with a Gaussian distribution and that the errors remain somehow dependent, the tests are still helpful in comparing relative model performance [5], [6].

3.2.5. Bubble Confidence Indicator

We use the bubble confidence indicator [11] to detect and predict bubbles. This index shows how well the targeted data fit the price movements from empirical bubble evidence from previous studies [11], [27], [28]. The larger the bubble confidence indicator, the more reliable the pattern (i.e., a crash is more likely).

A bubble confidence indicator for a given t_2 is calculated using the following steps:

(1) Iterate calibration for each time window where the start time, t_1, moves toward the end time, t_2, with a specific step, dt. In this study, we set the initial time range as 120 days and dt as 5 days, following Ito *et al*. [8]. Thus, each t_2 has 24 time windows.

(2) Count the number of cases in which $B < 0$ for each 24-calibration outcome, denoted as $[B<0]_{\text{count}}$
(3) Count the cases in which the parameters satisfy the conditions listed in Table 4 and name them as $[B<0]^{*}_{\text{count}}$, as derived from previous studies [11], [27], [28]. (4)Obtain the bubble confidence indicator as

$$\text{bubbleindicator} = \frac{[0<0]^{*}_{\text{Count}}}{[0<0]_{\text{Count}}}, \tag{19}$$

A higher bubble confidence indicator means that the price is likely to experience faster-thanexponential growth [29].

The bubble confidence indicator assumes that if a bubble is endogenous, the parameters must satisfy certain conditions, and the conditions are obtained inductively from empirical evidence [11], [27], [28]. Therefore, there are two drawbacks. Even if no apparent endogenous bubble trend is detected, it may only be offset by exogenous influences, and the hidden endogenous bubble may continue to grow. Furthermore, an apparent endogenous fluctuation may actually be a false endogenous detection caused by exogenous influences.

To test for these errors, we first calculate the bubble confidence indicator by fitting the original LPPLS model to the price data for each NFT project. We then calculate the indicators by fitting the original LPPLS model to the processed price data from which external effects are eliminated using the D and $v(t)$ from (12). The model from which we adopt the D and $v(t)$ depends on the comparison. If both calculate higher values, then there is a strong possibility that an endogenous bubble is occurring.

Table 3. Filtering conditions for each item in calculating the bubble confidence indicator.

Item	Condition
$\frac{\omega}{2\pi} ln \frac{t_c-t_1}{t_c-t_2}$	$[2.5, +\infty)$
$\frac{m\|B\|}{\omega\|C\|}$	$[0.5, +\infty)$

4. DATA

To compare the results with those of previous studies [8], the data used for our analysis included the same projects and periods as [8]. Namely, we used data from four major NFT projects: Decentraland (from March 19, 2018, to December 20, 2021), CryptoPunks (from May 17, 2018, to December 12, 2021), Ethereum Name Service (from May 4, 2019 to December 20, 2021), and ArtBlocks (from November 27, 2020 to December 20, 2021). These price data are available at https://nonfungible.com/ [30].

We used the same processed data as Ito *et al.*, provided by the Non-Fungible Corporation [30]. These data include weekly moving averages and average daily values of all NFT projects [8]. Notably, NFTs are not necessarily traded frequently, and huge price differences can be found within the same project.

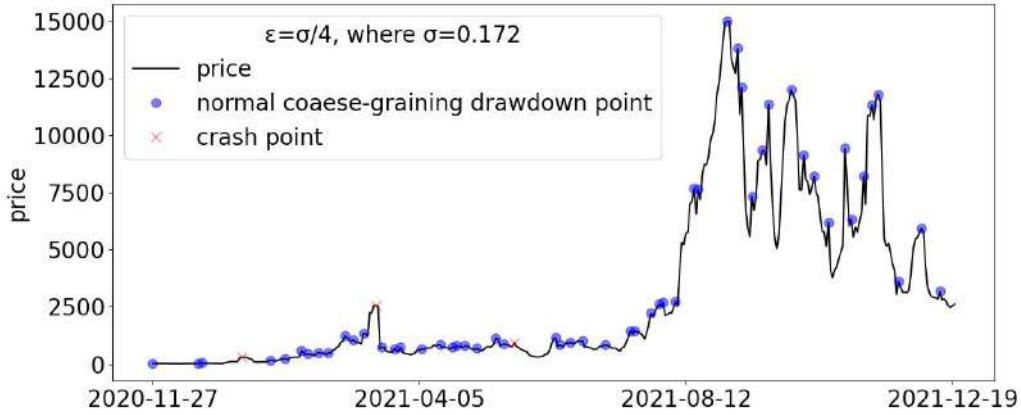
Regarding the volatility of cryptocurrency data, we applied the historical volatilities of 7, 30, and 90 days per BTC and ETH. These data are available at https://finance.yahoo.com/ [31].

5. RESULTS

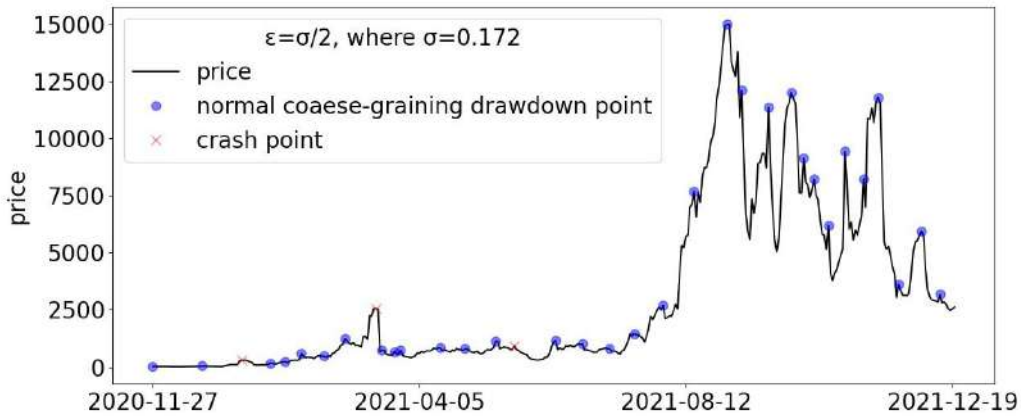
5.1. Selection of NFT Projects for the LPPLS Model

First, we obtained coarse-grained draw downs for the four NFT projects. Then, we extracted crash points by fitting a stretched exponential function with a 99.5% distribution point to indicate a crash. Figure 2 illustrates the coarse-grained drawdown points for ArtBlocks, some of which were considered crashes. Decentraland, CryptoPunks, and Ethereum Name Service did not indicate crashes, regardless of the epsilon value set.

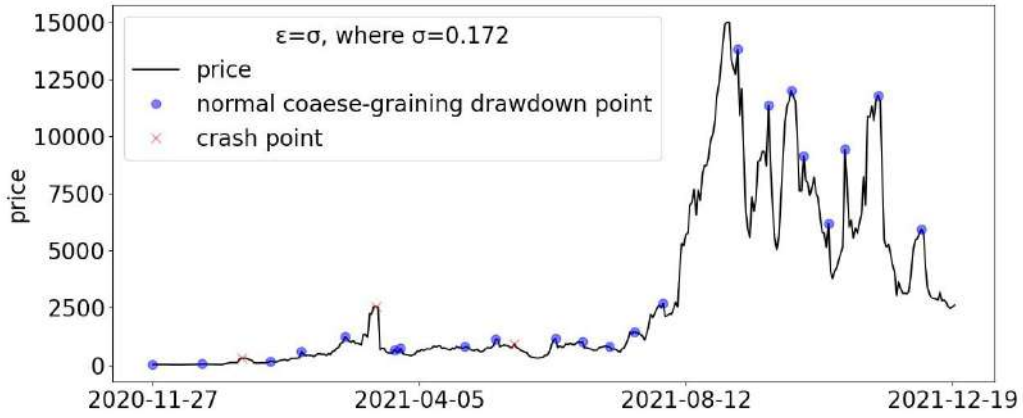
The following analytical results were obtained for the three crypto services. Although proper data extraction would improve LPPLS model fitting, even for these projects, extending the data extraction and their pretreatments was outside of the scope of this study. This may provide a future research opportunity in the near future.



(a)



(b)



(c)

Figure 2. Coarse-grained drawdowns in ArtBlocks: (a) $\varepsilon = \sigma/4$, (b) $\varepsilon = \sigma/2$, and (c) $\varepsilon = \sigma$.

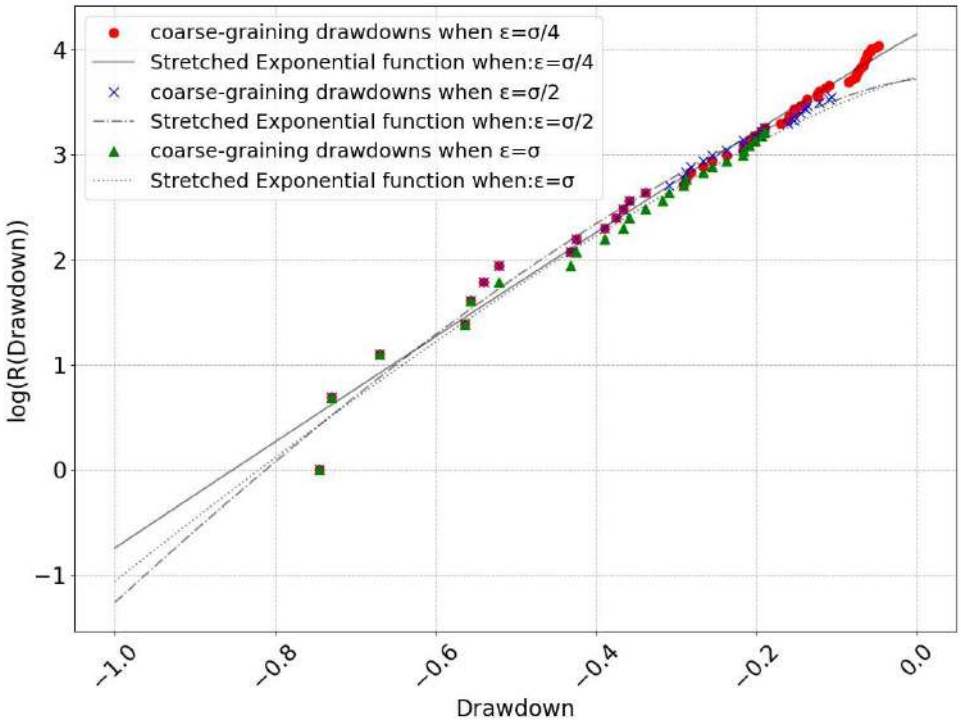


Figure 3. Stretched exponential function for drawdowns in ArtBlocks

Table 4. Logarithmic parameters of the stretched exponential function for ArtBlocks.

ε	A	b	z
σ/4	63.336	4.894	1.040
σ/2	41.271	4.986	1.403
σ	41.855	4.780	1.263

Table 5. Largest ArtBlockscrash.

Rank	Size	Start Date
1	−0.745	3/15/2021
2	−0.730	1/9/2021
3	−0.670	5/21/2021

5.2. Comparison of Original and Extended Models

When comparing the original LPPLS and extended models based on each explanatory variable, we calculated the AIC[26]and adjusted the R^2 values by fitting models (11) and (14) to each external factor. We also tested the levels of significance for each external explanatory factor by fitting model (14) and calculating the *p*-value of each external factor's coefficient.
The results revealed the following:

(1) As listed in Table 13, each adjusted R^2 in the extended model was higher than that of the original, and the AIC shrank in the extended models. This means that the explanatory power and forecasting accuracy, respectively, increased when the external variables were considered.
(2) For each project, the extended model was superior for detecting 90-day BTC volatility in Decentraland, 30-day ETH volatility in CryptoPunks, and seven-day ETH volatility in Ethereum Name Service.
(3) The explanatory power of the original model was much higher for CryptoPunks (Table 6-b) than for Decentraland (Table 6-a) and Ethereum Name Service (Table 6-c).
(4) In each best case, the external factor had a positive effect in Decentraland (Table 6-a), while it was negative in CryptoPunks (Table 6-b) and Ethereum Name Service (Table 6-c).
The implications inferred from these results are as follows:
(1) Assets are more sensitive to the cryptocurrency with which NFTs are traded.
(2) Owing to the fact that some projects had a good fit in the original model, the LPPLS model's assumption of mimetic irrational investors may be valid.
(3) Considering that there are collection-oriented NFTs and others for practical applications, a cross-sectional study based on these characteristics may be needed in the future (see Section 6).

Table 6. Adjusted R^2, AICs,and significance test results of the original and extended models.
(a) Decentraland

Model	**Original**	**Extended**					
External variable	-	7-day BTC volatility	30-day BTC volatility	90-day BTC volatility	7-day ETH volatility	30-day ETH volatility	90-day ETH volatility
Coefficient *t*-value	-	64.017 ***	−3.120 ***	64.453 ***	−17.506 ***	−18.617 ***	−19.517 ***
Adjusted R^2	0.579	0.763	0.584	0.770	0.656	0.665	0.669
AIC	3,403	2,612	3,3878	2,575	3,124	3,089	3,071
N	1,373	1,373	1,373	1,373	1,373	1,373	1,373

(b) CryptoPunks

Model	**Original**	**Extended**					
External variable	-	7-day BTC volatility	30-day BTC volatility	90-day BTC volatility	7-day ETH volatility	30-day ETH volatility	90-day ETH volatility
Coefficient *t*-value	-	−3.266 ***	−1.946	−7.784 ***	−17.852 ***	−19.529 ***	−19.363 ***
Adjusted R^2	0.969	0.954	0.954	0.970	0.963	0.974	0.973
AIC	2,169	2,688	2,694	2,138	2,412	1,932	1,987
N	1,314	1,314	1,314	1,314	1,314	1,314	1,314

(c) Ethereum Name Service

Model	**Original**	**Extended**					
External variable	-	7-day BTC volatility	30-day BTC volatility	90-day BTC volatility	7-day ETH volatility	30-day ETH volatility	90-day ETH volatility
Coefficient *t*-value	-	−8.931 ***	−8.690 ***	−8.341 ***	−9.788 ***	−11.369 ***	−11.361 ***
Adjusted R^2	0.622	0.664	0.664	0.663	0.707	0.669	0.667
AIC	1,953	1,839	1,840	1,841	1,707	1,825	1,830
N	962	962	962	962	962	962	962

Coefficient *t*-values are of the external variables. *** shows the significance level at the 1% confidence level.

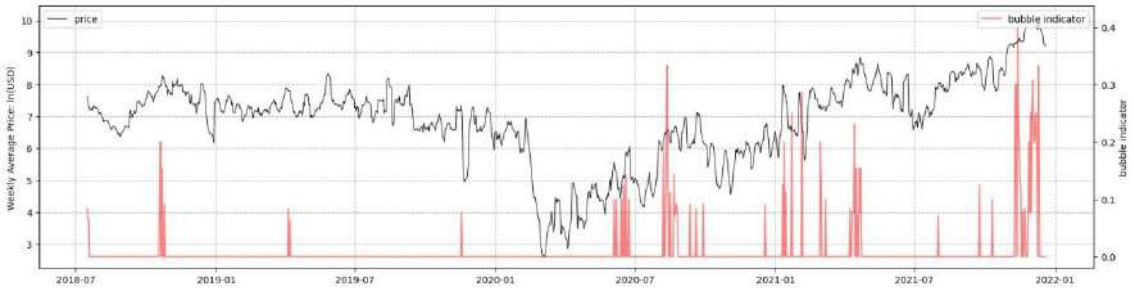
5.3. Comparison of Bubble Confidence Indicators

We calculated the bubble confidence indicators for the original data using the logarithm of the weekly moving average price of each NFT project (Fig. 4-a, b, c) and the processed price data from which external effects were eliminated using the D and $\upsilon(t)$ in (12) and shown in (Fig.

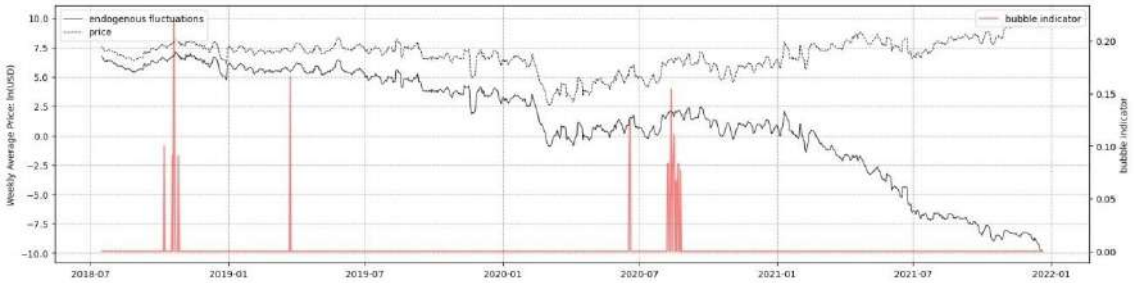
4a',b', and c'). The model D and $\upsilon(t)$ parameters adopted were determined by comparing the model evaluation results.

Overall, the original indicator correctly detected endogenous bubbles because the trends revealed by the indicators were roughly aligned. However, as in July 2022 with CryptoPunks, the original bubble confidence indicator did not detect a high endogenous risk; however, the indicator from the processed data did.

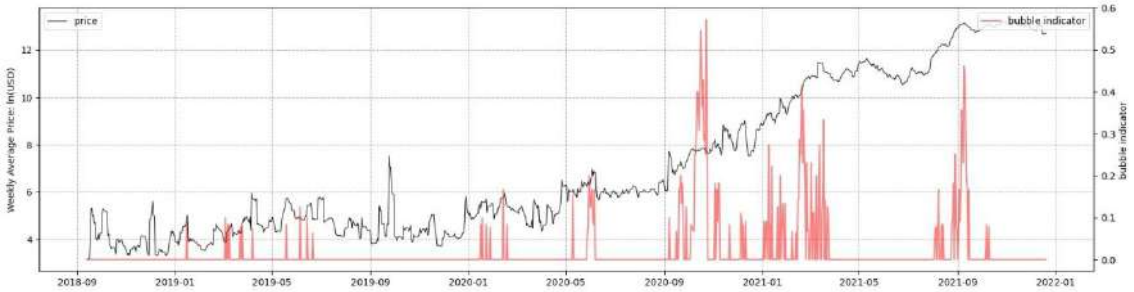
Therefore, a type-2 error may have occurred when predicting the bubble risk based on the confidence indicators calculated from the original data. However, a more realistic bubble confidence indicator might not be determined, as we did not examine the bubbles using multiple approaches (see Section 6).



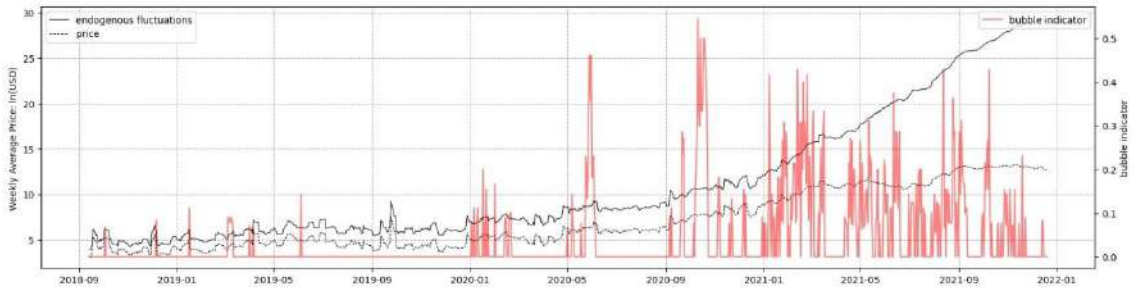
(a) *Decentraland* (the bubble confidence indicator calculated from the original price data)



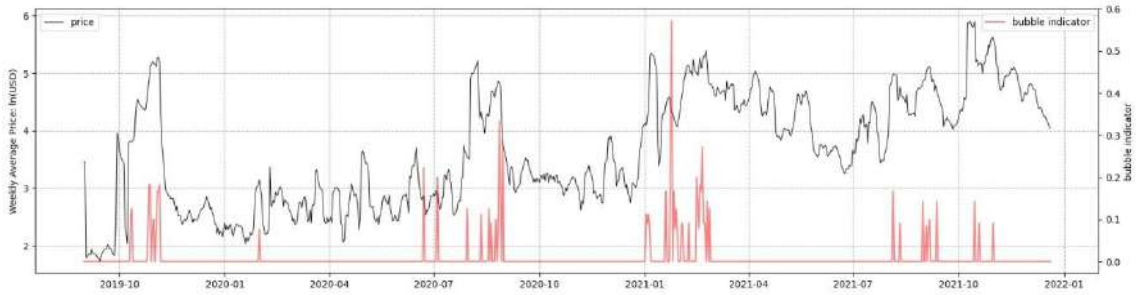
(a') *Decentraland* (the bubble confidence indicator calculated from the processed price data)



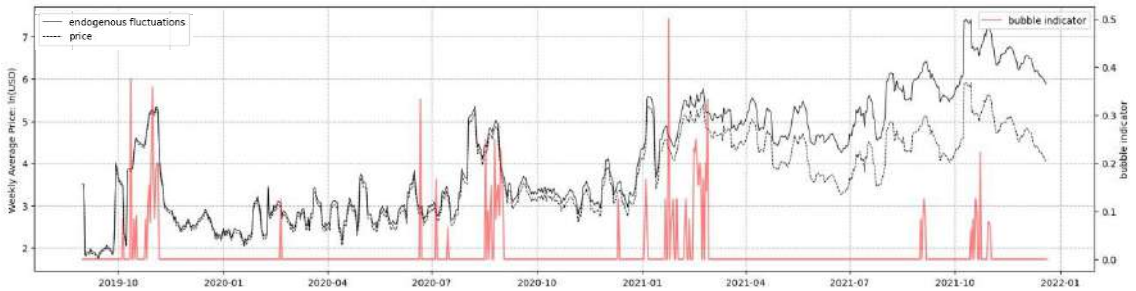
(b) *CryptoPunks* (the bubble confidence indicator calculated from the original price data)



(b') *CryptoPunks* (the bubble confidence indicator calculated from the processed price data)



(c) *Ethereum Name Service* (the bubble confidence indicator calculated from the original price data)



(c') *Ethereum Name Service*(the bubble confidence indicator calculated from the processed price data)

Figure 4. Results of the bubble confidence indicator of the original and processed data

6. CONCLUSION

Our conclusions are summarized as follows:
(1) Using the concept of drawdown, we successfully selected NFT projects to fit LPPLS analysis appropriately.
(2) We extended the original LPPLS model by incorporating external variables to improve its explanatory power and forecasting accuracy.
(3) By comparing the two kinds of confidence indicators, the results of endogenous bubble detection were confirmed.

Notably, the methods used to select appropriate NFT projects can be improved. For example, a crash might also be defined by considering its course-grained time horizon. It may also be possible to use the LPPLS model to analyze projects by cropping the period to be analyzed.

Moreover, there is room to consider additional explanatory variables. Cryptocurrency volatilities were employed in this study as explanatory variables, but as noted, their relationship with NFT prices remains unclear. Notably, an attention index that reflects public interest in a given project could be leveraged as an alternative variable.

We showed that bubble confidence indicators can appear differently depending on whether external variables are considered. However, this did not necessarily improve the predictive power of the model. We must continue to examine the bubbles using different approaches (e.g., Metcalfe's Law) for triangulation purposes.

Finally, because only a few projects were covered, it was difficult to make cross-sectional comparisons that considered the characteristics of each project. By broadening the scope in the future, it may be possible to identify project characteristics that influence their susceptibility to internal or external influences.

REFERENCES

[1] D.-R. Kong & T.-C. Lin, (2021) Alternative Investments in the Fintech Era: The Risk and Return of Non-fungible Token (NFT). SSRN Electronic Journal.

[2] @opensea, (2022) "However, we've recently seen misuse of this feature increase exponentially. Over 80% of the items created with this tool were plagiarized works, fake collections, and spam.," [Online]. Available: https://twitter.com/opensea/status/1486843204062236676?s=20&t=u12c2bYJgh4_Vxqg1DYjQ. [Accessed 11 2022].

[3] DappRadar, "OpenSea | DappRadar," [Online]. Available: https://dappradar.com/ethereum/marketplaces/opensea. [Accessed 17 11 2022].

[4] D. Sornette & W.-X. Zhou, (2006) "Predictability of large future changes in major financial indices," International Journal of Forecasting, Vol. 22, No. 1, pp153-168.

[5] W.-X. Zhou & D. Sornette, (2006) "Fundamental factors versus herding in the 2000-2005 US stock market and prediction," Physica Part A. Statistical Mechanics and its Applications, Vol. 360, No. 2, pp459-482.

[6] Z. Hu & C. Li, (2017) "New JLS-Factor Model versus the Standard JLS Model: A Case Study on Chinese Stock Bubbles," Discrete Dynamics in Nature and Society, Vol. 2017, pp1-15.

[7] R. Song, M. Shu & W. Zhu, (2022) "The 2020 global stock market crash: Endogenous or exogenous?," Physica A: Statistical Mechanics and its Applications, Vol. 585, p126425.

[8] K. Ito, K. Shibano & G. Mogi, (2022) "Bubble Prediction of Non-Fungible Tokens (NFTs): An Empirical Investigation," arXiv preprint arXiv:2203.12587.

[9] Y. Wang, F. Horky, L. J. Baals, B. M. Lucey & S. A. Vigne, (2022) "Bubbles all the way down? Detecting and date-stamping bubble behaviours in NFT and DeFi markets," Journal of Chinese Economic and Business Studies, Vol. 20, No. 4, pp1-22.

[10] A. Johansen & D. Sornette, (2002) "Endogenous versus exogenous crashes in financial markets," arXiv preprint cond-mat/0210509.

[11] Didier Sornette, Guilherme Demos, Qun Zhang, Peter Cauwels, Vladimir Filimonov & Qunzhi Zhang, (2015) "Real-time prediction and post-mortem analysis of the shanghai 2015 stock market bubble and crash," Swiss finance institute research paper, pp15-31.

[12] Didier Sornette, (2003) Why stock markets crash, Princeton University Press.

[13] Tiago Cruz Gonçalves, Jorge Victor Quiñones Borda, Pedro Rino Vieira & Pedro Verga Matos, (2022) "Log periodic power analysis of critical crashes: Evidence from the Portuguese stock market," Economies, Vol. 10, No. 1, p14.

[14] A. Johansen, (2004) "Origin of crashes in three US stock markets: Shocks and bubbles," Physica Part A. Statistical Mechanics and its Applications, Vol. 338, No. 1-2, pp135-142.

[15] D. Indiran, M. Ismail & Z. Isa, (2019) "Financial bubble theory and the log periodic power law application to Malaysian stock market," International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol. 8, No. 4S, pp2278-3075.

[16] Hideyuki Takagi, (2021) "Exploring the endogenous nature of meme stocks using the log-periodic power law model and confidence indicator," Vol. 12,No. 1, pp263-274.
[17] Min Shu & Wei Zhu, (2020) "Real-time prediction of Bitcoin bubble crashes," Physica Part A. Statistical Mechanics and its Applications, Vol. 548, No. 124477.
[18] J. Geuder, K. Harald & N. F. Wagner, (2019) "Cryptocurrencies as financial bubbles: The case of Bitcoin," Finance Research Letters, Vol. 31, No. C.
[19] Emilie Jacobsson, "How to predict crashes in financial markets with the Log-Periodic Power Law," Department of Mathematical Statistics, Stockholm University.
[20] C. Voglis & I. Lagaris, (2004) "A rectangular trust region dogleg approach for unconstrained and bound constrained nonlinear optimization.,"" WSEAS International Conference on Applied Mathematics, Vol. 7.
[21] Vladimir Filimonov, Didier Sornette, (2013) "A stable and robust calibration scheme of the logperiodic power law model," Physica Part A. Statistical Mechanics and its Applications, Vol. 392, No. 17, pp3698-3707.
[22] S. Osivand & H. Abolhasani, (2021) "Investigating the factors affecting the price of NonFungible Tokens (NFTS)," Journal of Economics and Finance, Vol. 12, No. 5, pp6-8.
[23] Michael Dowling, (2022) "Is non-fungible token pricing driven by cryptocurrencies?," Finance Research Letters, Vol. 44, p102097.
[24] Lennart Ante, (2022) "The non-fungible token (NFT) market and its relationship with bitcoin and Ethereum," Fintech, Vol. 1, No. 3, pp216-224.
[25] B. I. Technologies, (2022) "lppls," [Online]. Available: https://github.com/Boulder-InvestmentTechnologies/lppls.git. [Accessed 9 2022].
[26] H. Akaike, (1974) "A new look at the statistical model identification," IEEE Transactions on Automatic Control, Vol. 19, No. 6, pp716-723.
[27] D. Sornette & A. Johansen, (2001) "Significance of log-periodic precursors to financial crashes," Quantitative Finance, Vol. 1, No. 4, p452.
[28] Anders Johnsen & Didier Sornette, (2010) "Shocks, crashes and bubbles in financial markets.," Brussels Economic Review, Vol. 53, No. 2, pp201-253.
[29] T. Mathur, (2020) "The theory behind a bubble burst," SSRN Electronic Journal, Vol. 3629319.
[30] N. Corporation, "NonFungible.com | NFT market stats, sales tracker, rankings & news," [Online]. Available: https://nonfungible.com/. [Accessed 1 11 2022].
[31] Yahoo, "Yahoo Finance - Stock Market Live, Quotes, Business & Finance News," [Online]. Available: https://finance.yahoo.com/. [Accessed 1 11 2022].
[32] Usman W. Chohan, (2021) "Non-fungible tokens: Blockchains, scarcity, and value," Critical Blockchain Research Initiative (CBRI) Working Papers.
[33] A. Johansen & D. Sornette, (2010) "Shocks, crashes and bubbles in financial markets," Brussels Economic Review, Vol. 53, No. 2, pp201-253.
[34] L. Onsager, (1944) "Crystal statistics. I. A two-dimensional model with an order-disorder transition," Physical Review, Vol. 65, No. 3-4, pp117-149..
[35] Bernard Derrida, L. de Seze & C. Itzykson, (1983) "Fractal structure of zeros in hierarchical models," Journal of Statistical Physics, Vol. 33, No. 3, pp559-569.
[36] J. Nelder & R. Mead, (1965) "A simplex method for function minimization," Computer Journal, Vol. 7, No. 4, pp308-313.
[37] Decentraland, "Decentraland," [Online]. Available: https://play.decentraland.org/. [Accessed 15 1 2023].
[38] L. Labs, "CryptoPunks," [Online]. Available: https://www.larvalabs.com/cryptopunks. [Accessed 15 1 2023].
[39] A. Blocks, "Art Blocks," [Online]. Available: https://www.artblocks.io/. [Accessed 15 1 2023].
[40] ENS, "Ethereum Name Service," [Online]. Available: https://ens.domains/. [Accessed 15 1 2023].
[41] Y. Wang, (2022) "Volatility spillovers across NFTs news attention and financial markets," International Review of Financial Analysis, Vol. 83, p102313.
[42] A. Johansen, D. Sornette & O. Ledoit, (1999) "Predicting financial crashes using discrete scale invariance,," arXiv preprint cond-mat/9903321.
[43] A. Johansen, O. Ledoit & D. Sornette, (2000) "Crashes as critical points," International Journal of Theoretical and Applied Finance, Vol. 3, No. 02, pp219-255.
[44] Š. J. Širca & M. Omladic, (2017) "The JLS model with ARMA/GARCH errors," Ars Mathematica Contemporanea, Vol. 13, No. 1, pp63-79.

AUTHOR

Ikkou Okubo, attended the University of Tokyo as an undergraduate, and continued a higher education there and completed a master's degree in Technology Management for Innovation at the Graduate School of Engineering.

ADDRESSING CLASS VARIABLE IMBALANCE IN FEDERATED SEMI-SUPERVISED LEARNING

Zehui Dong, Wenjing Liu , Siyuan Liu and Xingzhi Chen

School of Data Science and Application, Inner Mongolia University of Technology, Hohhot, China

ABSTRACT

Federated Semi-supervised Learning (FSSL) combines techniques from both fields of federated and semi-supervised learning to improve the accuracy and performance of models in a distributed environment by using a small fraction of labeled data and a large amount of unlabeled data. Without the need to centralize all data in one place for training, it collect updates of model training after devices train models at local, and thus can protect the privacy of user data. However, during the federal training process, some of the devices fail to collect enough data for local training, while new devices will be included to the group training. This leads to an unbalanced global data distribution and thus affect the performance of the global model training. Most of the current research is focusing on class imbalance with a fixed number of classes, while little attention is paid to data imbalance with a variable number of classes. Therefore, in this paper, we propose Federated Semi-supervised Learning for Class Variable Imbalance (FCVI) to solve class variable imbalance. The class-variable learning algorithm is used to mitigate the data imbalance due to changes of the number of classes. Our scheme is proved to be significantly better than baseline methods, while maintaining client privacy.

KEYWORDS

Federal semi-supervised learning, Federated learning, Semi-supervised learning, Class variable imbalance

1. INTRODUCTION

In recent years, the rapid development of both Internet of Things (IoT) and Artificial Intelligence (AI) technologies has driven the boom of mobile smart devices, but also the massive amount of data generation that accompanies them. According to Cisco (Cisco), it is expected that by 2025, the data generated by IoT devices worldwide at the edge of the network will reach 79.4 ZB [1]. Faced with the data explosion brought about by the rapid development, the cloud computing model converges IoT and AI by using the data generated by IoT devices to train machine learning models and then using the models to analyze the data (e.g., image recognition, intelligent prediction, etc.), which in turn drives the implementation of AI and IoT in various application scenarios [2]. However, this model requires all data to be aggregated to a cloud data center and then processed centrally, which can lead to huge transmission delays and privacy leaks.

The combination of Edge Computing (EC) [3] and Federated Learning (FL) [4] offers the possibility to solve the above problems. First, EC runs deep learning model training tasks on edge nodes close to data sources by shifting from centralized to distributed edge networks based on cloud data centers. Second, FL unites many edge nodes at the edge of the network to jointly participate in distributed collaborative model training on the basis of not sharing data. In this

David C. Wyld et al. (Eds): CCSEA, AIFU, EMSA, NLCAI, NCOM, SIPRO, SEA, DKMP, BDML, BIoT, CLOUD - 2023
pp. 257-272, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130522

way, not only the latency of the data transmission process is reduced, but also the level of security and privacy protection of the whole training process is improved.

However, at the data level, the data collected by different devices are scattered and variable due to the different geographical locations of the devices [5], and the data used for FL model training is class unbalanced. These class-imbalanced data are reflected in the model classification task as class imbalance [6], and the models trained using such data are limited. In the case of a binary classification task, the ratio between positive and negative data may be as high as 999:1, and it is difficult for a model trained with such data to identify the data classes that account for a relatively small amount of data [7]. Meanwhile, FL enables multiple devices to collaboratively train a global model by aggregating the local model parameters of multiple devices, without the need to upload data from local to the cloud. However, in FL iterative training, after some devices stop participating in the training midway, it will result in the loss of some data, as well as new devices joining the training will add new data, which in turn will lead to an unbalanced global data distribution, thus affecting the performance of the global model [8]. Most of the current research is focusing on class imbalance with a fixed number of classes [6], while little attention has been paid to class imbalance with a changing number of classes.

In edge networks, we found that for most mobile smart devices, most of their collected data are unlabeled data [9], and only a small fraction of them are labeled data, which are unable to support model training. A potential solution is to use Federated Semi-supervised Learning [10] (FSSL) approach, which updates FL models using labeled data and unlabeled data with pseudo-labels by using semi-supervised learning to train these data and generating pseudo-labels for unlabeled data, this approach is very effective in the case of class imbalance and lack of labels. Since FL training is performed by exchanging gradients in an encrypted form, the training data is not fully observable to the edge server and aggregation server. Therefore, it is crucial to detect the category changes in FL and mitigate their impact.

In order to better address the class imbalance problem of class number changes, this paper proposes Federated Semi-supervised Learning for Class Variable Imbalance (FCVI) to address class variable imbalance, which enables the model to be trained on a large amount of decentralized data, while can effectively solve the class imbalance problem in the dataset. The main contributions of this paper are 3-fold.

(1) A federal gradient monitoring scheme is designed, which can infer the changes of each class in the FL training process through the gradient without transmitting other data, ensuring the privacy protection of the model training process.
(2) A class-variable learning algorithm is designed. After the monitoring scheme monitors the class changes, we selectively expand the data using unlabeled data and semi-supervised learning self-training methods to alleviate the class imbalance due to the changes in the number of classes.
(3) A real scenario with imbalanced image data is deployed to design distributed training experiments using an image classification model, which in turn improves the accuracy of model recognition.

2. RELATED WORK

2.1. Class Imbalance Machine Learning

Research on class imbalance machine learning has focused on supervised learning and semi-supervised learning.

In Supervised Learning (SL), models require labeled data to update their parameters, and their approaches to address class imbalance can be divided into data-level and algorithm-level. The data level utilizes data oversampling or undersampling methods. In the literature [11], the authors use an oversampling approach to replicate randomly selected samples from a small number, which reduces both inter-category and intra-category imbalances, but such an approach can lead to the occurrence of overfitting. In the literature [12], the authors use the undersampling method to remove randomly from the majority of classes until all classes have the same number of samples, but the significant disadvantage of this is that it discards a portion of the available data. Unlike the data level, the algorithm level is to modify the training algorithm or the network structure. For example, in the literature [13], the authors use a re-weighting approach to assign weights to different training data and address the problem of low accuracy in a few classes by equalizing the loss function and ignoring the gradient of most classes. All the above methods are tuned with labeled data, while their performance is unknown in the absence of labeled data.

In semi-supervised learning (SSL), label-free data can be used to improve model performance. In the literature [14], the authors found that SSL using unlabeled data can mitigate the impact of data imbalance on model training. In the literature [15], the authors propose a method to suppress the loss of consistency to suppress the loss of a few classes and thus mitigate the impact of data imbalance. In the literature [16], the authors mitigate the data imbalance in SSL scenarios by softening the pseudo-labels generated in the deviation model through convex optimization. In the literature [17], the authors eliminate the effect of unbalanced data by pseudo-labeling unlabeled data and then expanding these data into labeled data to improve the distribution of the data.

2.2. Class Imbalance Federal Learning

Due to the high computational effort of training machine learning models, researchers have been exploring the use of multiple devices to learn a general model. Federated learning has emerged as an effective solution to learn a global model while keeping all training data on the local device. In federated learning, its research has focused on reducing communication overhead and preserving privacy, and only a few studies have paid attention to the problem of class imbalance in training data leading to a decrease in model accuracy. In the literature [18], the authors use weighted optimization to alleviate the imbalance problem by assigning higher weights to the nodes that produce better training results. However, this method makes the network bandwidth increase, which in turn increases the network burden, and uploading local data to the server causes privacy issues. In the literature [19], the authors dealt with this by reassigning training nodes and grouping them for training based on their local and global similarity degree. But it does not solve the problem of unbalanced local data distribution of nodes. In the literature [20], the authors improved the distribution of unbalanced data using data augmentation methods such as random displacement, random rotation, random shearing, and random scaling. Because the augmented data are similar data generated on the original data, this makes the model training more prone to overfitting.

3. DESIGN OF FCVI

3.1. Problem Definition

In the conventional training scenario, the model training is focused on a single device and the category distribution is known, while the training data are labeled and sufficient. In the edge network FSSL, the model training is distributed over an aggregation server and an edge server and the class distribution is not known, while the training data consists of a large amount of unlabeled data and a small amount of labeled data. Specifically, there is one label set $\mathcal{D}_i^{(L)} = \{(x_j, y_j): j \in (1, \cdots, N)\}$ on each edge server i. The number of training samples for class l in label set $\mathcal{D}_i^{(L)}$ is denoted as N_i^l, i.e., $\sum_l^L N_l = N$. In addition to label set $\mathcal{D}_i^{(L)}$, there is an unlabeled set $\mathcal{D}_i^{(U)} = \{u_c \in \mathbb{R}^d: c \in (1, \cdots, C)\}$, where $C \gg N$, label set $\mathcal{D}_i^{(L)}$ and unlabeled set $\mathcal{D}_i^{(U)}$ have the same class distribution. We use the label fraction $\beta = N/(N + C)$ to represent the percentage of labeled data to the total data.

In FSSL, local model training using labeled data can be considered as regular centralized learning, so the local model training is on a multilayer feedforward neural network with inputs in the feature space X and label space $Y = \{1, \cdots, L\}$, and the L output size of the classifier is equal to the number of classes. In this paper, all the intermediate layers are combined into a single hidden layer HL, containing s neurons. We send the j-th sample of class l, denoted as $X_j^{(l)}$, to the classifier L, whose output corresponding to HL is denoted as $Y_j^{(l)} = \left[y_{j,(1)}^{(l)}, \cdots, y_{j,(s)}^{(l)}\right]$. the output of the last layer is $Z_j^{(l)} = \left[z_{j,(1)}^{(l)}, \cdots, z_{j,(L)}^{(l)}\right]$, and then the $softmax$ operation is performed to obtain the probability vector $S = \left[f_{j,(1)}^{(l)}, \cdots, f_{j,(L)}^{(l)}\right]$. the function $f: \{X \Rightarrow S\}$ will be mapped to the output probability simplex S. f parameterizes the hypothetical class $\mathbb{W}$, i.e., the overall parameters of the classifier. In addition, the connection weights from HL to the output layer are denoted as $W = \left[W_{(1)}, W_{(2)}, \cdots, W_{(Q)}\right]$ and $W \in \mathbb{W}$. In each training iteration, we apply the backpropagation method to calculate the gradient of the loss function $L(\mathbb{W})$ under $\mathbb{W}$. Denoting the weight of the t-th training iteration by $\mathbb{W}(t)$ and the learning rate by λ, we then have $\mathbb{W}(t + 1) = \mathbb{W}(t) - \lambda \nabla L(\mathbb{W}(t))$.

3.2. Federal Gradient Monitoring Method

In edge intelligence scenarios, the data collected by edge devices are differentiated. Because of the different locations of the edge devices, the data collected by them are also different from other edge devices in terms of categories. At the same time, edge devices are often mobile and unstable, therefore, in the iterative training of the FL model, some of the devices will no longer participate in the training, which will result in the loss of some categories of data, and new devices will add new categories of data after joining the training, which will lead to changes in the number of data categories in the iterative training of the FL model, thus affecting the performance of the global model. In order to detect and mitigate the performance degradation caused by class-variable imbalance, we design a global monitoring method based on FL gradient to estimate the class variation of FL model iterative training as follows.

In any real-valued neural network f, when its last layer is a linear layer with $softmax$ operations, then for any input samples $X_i^{(p)}$ and $X_j^{(p)}$ of the same class p, if the inputs Y_i and Y_j in the last layer are the same, then the gradient W of the connection weights caused by and between the layers before the last layer $X_i^{(p)}$ and $X_j^{(p)}$ is the same [21].

In small-batch (mini-batch) training, the gradients of the samples within a batch are accumulated to update the model parameters, i.e.

$$\Delta_{batch}W = -\frac{\lambda}{n^{batch}}\sum_{p=1}^{Q}\sum_{j=1}^{n^{(p)}}\nabla_{w_j^{(p)}}L(\mathbb{W}) \tag{1}$$

Data samples of the same class p will have similar $Y^{(p)}$,and therefore the corresponding gradients are very similar. If the mean value of the gradient is $\overline{\nabla_{w^{(p)}}L(W)}$, then equation (1) can be written as

$$\Delta_{batch}W = -\frac{\lambda}{n^{batch}}\sum_{p=1}^{Q}(\overline{\nabla_{\nabla_{w^{(p)}}L(W)}}\cdot n^{(p)}) \tag{2}$$

where $n^{(p)}$ is the number of samples of class p in the batch and n^{batch} is the size of the batch. For a round of local training in FL, the total number of iterations (Iteration) of local gradient updates is $\left[\left(\sum_{p=1}^{Q}\frac{N_p}{n^{batch}}\right)\cdot N_{ep}\right]$,where N_{ep} denotes the number of local Epochs. To illustrate the proportional relationship between the gradient size and the sample size, we assume that the parameters are relatively small and negligible within an Epoch. In this case, the different batches of $\overline{\nabla_{w^{(p)}}L(W)}$within an Epoch remain constant, and we can aggregate them to obtain an Epoch whose weights are updated as

$$\Delta_{epoch}W = -\frac{\lambda}{n^{batch}}\sum_{p=1}^{Q}(\overline{\nabla_{w^{(p)}}L(W)}\cdot N_p) \tag{3}$$

where N_p is the total number of p class samples.

In the standard FL distributed training setting, the global server generally aggregates the selected local gradients by the FedAvg algorithm as follows.

$$\nabla L(W)_{t+1}^{Avg} = \frac{1}{K}\sum_{j=1}^{K}\nabla L(W)_{t+1}^{j} \tag{4}$$

where K represents the number of clients.

Based on the above analysis, for any local training starting from the same current global model, the data samples of the same class p output very similar Y and similar $\overline{\nabla_{w^{(p)}}L(W)}$. In this case, the gradients obtained from class p in a global epoch are

$$\begin{aligned}\Delta_{global}W^{(p)} &= -\frac{\lambda}{n^{batch}\cdot K}\sum_{j=1}^{K}\left(\overline{\nabla_{w^{(p)}}^{J}L(W)}\cdot N_p^j\right)\\ &= -\frac{\lambda}{n^{batch}\cdot K}\overline{\nabla_{w^{(p)}}L(W)}(\sum_{j=1}^{K}N_p^j)\end{aligned} \tag{5}$$

Based on this relationship, we develop the following global monitoring method based on FL gradient. At round $t+1$, when the aggregation server monitors the change in the number K_{t+1} volume of the client, the global gradient by comparing the class p at round t is

$$\frac{\Delta_{global}W_t^{(p)}}{\Delta_{global}W_{t+1}^{(p)}} = \frac{-\frac{\lambda}{n^{batch}\cdot K_t}\overline{\nabla_{w^{(p)}}L(W)}(\sum_{j=1}^{K_t}N_p^j)}{-\frac{\lambda}{n^{batch}\cdot K_{t+1}}\overline{\nabla_{w^{(p)}}L(W)}(\sum_{j=1}^{K_{t+1}}N_p^j)} = \frac{K_{t+1}}{K_t}\cdot\frac{\sum_{j=1}^{K_t}N_p^j}{\sum_{j=1}^{K_{t+1}}N_p^j} \tag{6}$$

Because the aggregation server knows the number of edge servers K_{t+1} in round $t+1$ and the number of edge servers K_t in round t, the change ratio R_p of class p samples can be obtained from Equation (6) as :

$$R_p = \frac{K_{t+1}}{K_t}\cdot\frac{\Delta_{global}W_{t+1}^{(p)}}{\Delta_{global}W_t^{(p)}} = \frac{\sum_{j=1}^{K_{t+1}}N_p^j}{\sum_{j=1}^{K_t}N_p^j} \tag{7}$$

In the global monitoring method based on FL gradient, when the number of edge servers participating in FL training changes, the aggregation server uses Equation (7) after calculating the local FL gradient, we can obtain the change ratio vector $R = [R_1, \cdots, R_p, \cdots R_Q]$ for all classes in the current training round.

3.3. Class Variable Learning Algorithm

Once the aggregation server monitors a change in the number of edge servers, it indicates a change in the local and global data class distribution, which will negatively affect the performance of FL model training. Due to the different physical locations where the edge servers are located, some edge servers will have their class data samples that distinguish them from other edge servers. Therefore, among the class-variable imbalances caused by equipment movement, four categories can be classified according to the value of the change ratio R_p.

(1) When $R_p > 1$, then $\sum_{j=1}^{K_{t+1}}N_p^j > \sum_{j=1}^{K_{t+1}}N_p^j$, it is known that the global data class p samples are increasing in round $t+1$.
(2) When $0 < R_P < 1$, then $\sum_{j=1}^{K_{t+1}}N_p^j < \sum_{j=1}^{K_{t+1}}N_p^j$, it can be known that the $t+1$ round of global data p class samples are decreasing.
(3) When R_p is an outlier, then $\sum_{j=1}^{K_t}N_p^j = 0$, it can be known that the number of p class samples in the t round of global data is 0.
(4) When $R_p = 0$, then $\sum_{j=1}^{K_{t+1}}N_p^j = 0$, it is known that the number of p class samples in the $t+1$ round of global data is 0.

Single application of existing methods through local or global cannot effectively mitigate the impact of changes in data class distribution, so this paper designs a class-variable learning algorithm to mitigate the negative impact on FL model training by taking different approaches from local and global through the change ratio R_p obtained from FL gradient monitoring method, respectively, and Algorithm 1 shows the details of the class-variable learning algorithm.

Algorithm 1:class-variable learning algorithm

AggregationServer:

1: Initialize parametersθ_0;
2: **for** each global epoch$t+1$ in$1,\cdots,T$ **do**
3: **for each** client $i$in$1,\cdots,I_{t+1}$**do** //Collect edge server parameters
4: receive edge server $\theta_{i,t+1}$;
5: **end fo**r
6: $\theta_{t+1}=\sum_{i=1}^{|I_{t+1}|}\omega_i\theta_{i,t+1}$; //Aggregate Edge Server Parameters
7: **if** $I_{t+1}\neq I_t$**then** //Determine if the number of edge servers has changed
8: **for** each class $l$in $1,\cdots,L$**do**
9: $R^{(l)}\leftarrow$FLGradientMonitorMethod($\theta_{t+1}^{(l)},\theta_t^{(l)}$);
10: **if** $R^{(l)}>0$ **then**
11: $R_{min}=min(R_{min},R^{(l)})$;
12: **end if**
13: **end for**
14: **for** each $R^{(l)}$in $1,\cdots,R^{(L)}$**do**
15: **if** $R^{(l)}>0$ **then**
16:$\mu_{t+1}^{(l)}=\frac{R_{min}}{R^{(l)}}$;
17: **else if** $R^{(l)}=0$**then**
18: $\theta_{t+1}^{(l)}$=$\theta_t^{(l)}$;
19: $\mu_{t+1}^{(l)}=1$;
20: **else**
21: $\mu_{t+1}^{(l)}=1$;
22: **end if**
23: **end for**
24: μ_{t+1}=$\left[\mu_{t+1}^{(1)},\cdots\mu_{t+1}^{(l)},\cdots\mu_{t+1}^{(L)}\right]$;
25: $\theta_{t+1}=\left[\theta_{t+1}^{(1)},\cdots\theta_{t+1}^{(l)},\cdots\theta_{t+1}^{(L)}\right]$;
26: **end if**

Edge Server：

1: **for** each edge server i in $1,\cdots,I_{t+1}$ **do**
2: **for** each local epoch $t+1$ in$1,\cdots,T$ **do**
3: Class Variable Self-Training Method (μ_{t+1})
4: **end for**
5: **return** $\theta_{i,t+1}$to AggregationServer；
6: **end fo**r

First, when R_p is (1), the number of class$l$ samples in the global data of round t is 0, while the number of class l samples in round $t+1$ is not 0, indicating that the number of categories increases, and the aggregation server aggregates the class parameters normally in round t1 without performing other operations. Secondly, when R_p is (2), the number of class l samples in the global data in round t is not 0,while the number of class l samples in the $t+1$round is 0, indicating that the number of categories decreases,and the aggregation server uses round t to update the aggregation of the class parameters in round $t+1$ to avoid the problem of gradient disappearance due to the number of samples in the class being 0.Finally, when R_p is (3) or (4), selective data expansion is performed on the local edge server using unlabeled data through the SSL self-training method.

The self-training method [22] is a widely used iterative approach in semi-supervised learning. Traditionally, self-training involves training an initial model by using a small amount of labeled data, then using that model to predict unlabeled samples, and then selecting unlabeled data with high confidence to expand the label set. This process actually involves the classification model using its own prediction results to improve the model performance. The self-training approach requires only an initial model, a small amount of labeled data, and a large amount of unlabeled data to perform a complex semi-supervised learning task. Specifically, self-training works as follows.

(1) Step 1: An initial model is first trained using the labeled set $\mathcal{D}_i^{(L)} = \{(x_j, y_j): j \in (1, \cdots, N_i)\}$.
(2) Step 2: The trained classification model is used to predict the class labels of all the unlabeled datau_c. Among these predicted class labels, the one with the highest correct rate and exceeding the probability threshold is considered as the pseudo-label $\widehat{y_c}$, and the set of unlabeled data with such labels is the pseudo-label set $\widehat{\mathcal{D}}_i^{(U)} = \{(u_c, \widehat{y_c}): c \in (1, \cdots, C_i)\}$.
(3) Step 3: Include the pseudo-label set $\widehat{\mathcal{D}}_i^{(U)}$ in the label set $\mathcal{D}_i^{(L)}$, i.e., $\acute{\mathcal{D}}_i^{(L)} = \mathcal{D}_i^{(L)} \cup \widehat{\mathcal{D}}_i^{(U)}$. Retrain the model on the new label set.
(4) Iterate step 2 and step 3 repeatedly until the predicted class labels in step 2 are not satisfying the probability threshold or until no unlabeled data are retained.

To accommodate the class imbalance, we modify the self-training method. Instead of including all the data of pseudo-label set $\widehat{\mathcal{D}}_i^{(U)}$ in the label set $\mathcal{D}_i^{(L)}$, we select a subset $\hat{S}_i^{(U)} (\hat{S}_i^{(U)} \subset \widehat{\mathcal{D}}_i^{(U)})$ from it to expand the label set $\widehat{\mathcal{D}}_i^{(U)}$, i.e., $\acute{\mathcal{D}}_i^{(L)} = \mathcal{D}_i^{(L)} \cup \hat{S}_i^{(U)}$. The selection rule for this subset$\hat{S}_i^{(U)}$: the smaller the change ratio R_p of class p, the more unlabeled data are predicted as class p. Specifically: the amount of sample change R_p of class p is known, and the unlabeled data corresponding to being predicted as class p are contained in a pseudo-labeled subset ratio of

$$\mu_p = \frac{R_{min}}{R_p}$$

where R_{min} is the minimum value of the amount of sample change in class p. For example, when class p is the class with the smallest amount of change in each class ($R_{min} = R_p$) and the unlabeled data of class p is included in the pseudo-labeled subset ratio μ_p =1, then all the unlabeled data of class p is retained. When $R_{min} = \frac{1}{2}$, $R_p = 2$, the unlabeled data of class p is contained in the pseudo-labeled subset ratio $\mu_p = \frac{1}{4}$, then the unlabeled data of $\frac{1}{4}$ in class p is retained.

4. EXPERIMENTS AND RESULTS

In this paper, the Convolutional Architecture for Fast Feature Embedding (Caffe) [23] deep learning framework is used to implement the proposed multilevel class rebalancing distributed method and its accuracy is verified by comparison.

4.1. Experimental Configuration

4.1.1. Dataset and Model

Simulating the classification task in a real environment, the ImageNet dataset is chosen to construct the class-variable imbalance dataset in this paper. The ImageNet dataset is a large

image classification dataset established to promote the development of computer graphics recognition technology, with a large number and high resolution of thousands of classes of image data. We used the ILSVRC2012 dataset, which contains 1000 categories and 1.2 million images. We selected 15 categories out of the 1000 categories, each containing 1300 images, and then divided them into 15600 training images and 3900 test images according to the allocation ratio of 8:2 between the training and test sets. For the training images, labeled data accounts for 30% of all data, and the remaining 70% of data is unlabeled data, i.e., label score $\beta = 0.3$.

To observe the effect of class variable imbalance due to device movement under the edge network, in the first 50 rounds of training, the number of classes of the training data is 10, as well as the class distribution is balanced. Starting from the 50th round, a new class is added or reduced in the 50th, 100th, 150th, 200th, and 250th rounds, respectively, as well as the class distribution of their data also changes randomly. In the class increase imbalance, the number of categories changes from 10 to 11, 12, 13, 14, and 15 at rounds 50, 100, 150, 200, and 250; in the class decrease imbalance, the number of categories changes from 10 to 9, 8, 7, 6, and 5 at rounds 50, 100, 150, 200, and 250.

Due to the complexity of image data in the ImageNet dataset, we choose the AlexNet image classification model as the backbone model. AlexNet is the model that first introduced convolutional neural network into the field of computer vision and achieved a breakthrough, winning the ILSVRC 2012 championship. AlexNet is a convolutional network consisting of five convolutional layers and three fully connected layers, with a total of eight layers of convolutional neural network. We tuned the parameters of AlexNet according to the hardware configuration of the physical machine and the characteristics of the dataset.

4.1.2. Edge Network Setup

To evaluate the scheme proposed in this paper and compare it with other schemes in the literature, we built a real distributed training edge network and deployed it on a physical device, as shown in Figure 1. We use a Raspberry Pi 4B (Raspberry Pi 4B) as the edge device; a workstation with a GPU as the edge server, which is closer to the edge device; and a server superior to the workstation GPU as the aggregation server, which is further away from the edge device than the edge server. The specific hardware configuration is shown in Table 1, which is common in actual production environments and has some generality.

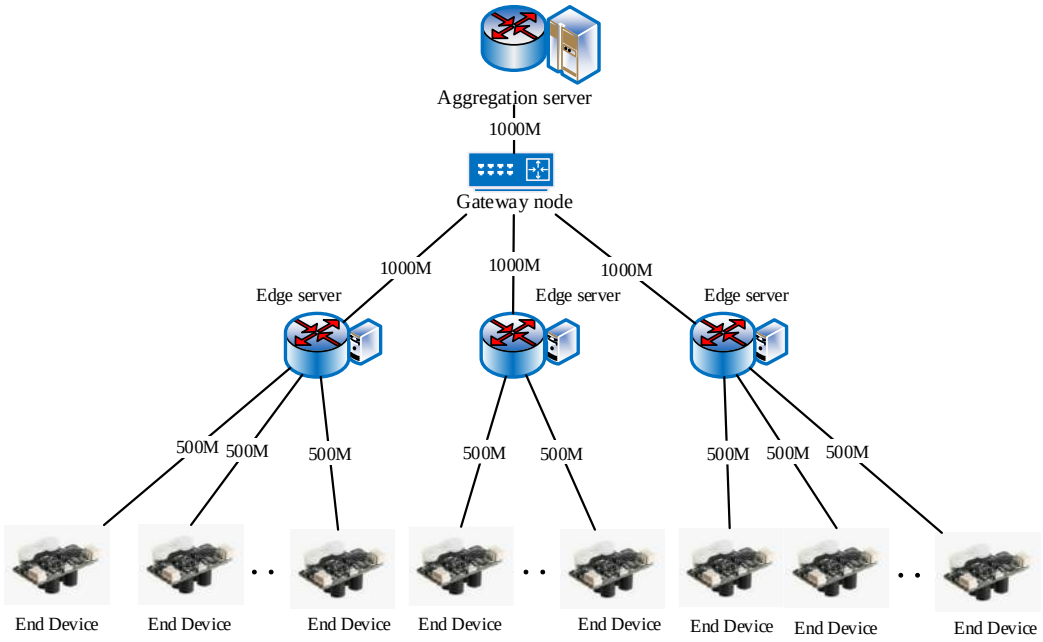


Figure 1 Distributed training edge network topology diagram

Table 1 Hardware configuration information of each edge device

Node Type	End Device		Edge Server		Aggregation Server	
OS	Raspberry Pi OS		Ubuntu 20.04 LTS		Ubuntu 16.04 LTS	
CPU	ARM CortexA-72 @ 1.5GHz		Intel(R) Core(TM) i5-10500 @3.10GHz		Intel(R) Xeon(R) Gold 6130 CPU @2.10Ghz	
Memory	2GB DDR4		16GB DDR4		512GB DDR4	
GPU	500 MHZ VideoCore VI		NVIDIA GeFroce RTX 2060 SUPER		NVIDIA Tesla P100 *4	
Wireless Network	802.11ac(2.4/5GHz) Bluetooth 5.0		802.11ac(2.4/5GHz) Bluetooth 5.0		802.11ac(2.4/5GHz) Bluetooth 5.0	
Wired Network	Gigabit Ethernet		Gigabit Ethernet		Gigabit Ethernet	
Power supply	Idle power	10W	Idle power	110W	Idle power	500W
	CPU full load power	15W	CPU full load power	65W	CPU full load power	125W
	GPU Full Load Power	12W	GPU Full Load Power	180W	GPU Full Load Power	2400W

4.1.3. Comparison Scheme

In order to compare the effectiveness of FCVI, we implemented the scenarios proposed in this paper and the most representative comparison scenarios. All these scenarios are implemented on our testbed using the same dataset and backbone model, and the details are described as follows.

(1) FCVI: FCVI's are set according to the algorithm above.
(2) FedRL[21]: FedRL can infer the composition of the training data for each round of FL through the aggregation server and mitigate the effect of category imbalance by a new loss function, Ratio Loss.

(3) K-SMOTE[20]: K-SMOTE updates its local model by directly interacting with the connected neighboring edge servers and generates synthetic data for a few classes based on linear interpolation to rebalance the local token set on the edge servers. In K-SMOTE, the aggregation server is not involved in the training process except for the initial model deployment.
(4) FedAvg [24]: In FedAvg, all edge servers use asynchronous stochastic gradient descent (SGD) in parallel to compute and update their weights, and then the server collects updates from clients and aggregates them using the FedAvg algorithm.

4.1.4. Evaluation Metrics

$Accuracy$ is the ratio of the number of all correct classifications of the classifier to the total number of.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Where, TP、FP、FN and $TN$are abbreviated forms of true positive, false positive, false negative and true negative, respectively.

Precision is the ratio of all "correctly classified numbers (TP)" to all "actually classified numbers ($TP + FP$)".

$$Precision = \frac{TP}{TP + FP}$$

Recall is the ratio of all "number of correct classifications (TP)" to all "actual classifications as correct ($TP + FN$)".

$$Recall = \frac{TP}{TP + FN}$$

F1 is the summed average of the precision rate and recall rate.

$$F1 = \frac{2Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN}$$

4.2. Analysis of Experimental Results

In this section, the effectiveness of this paper's scheme and the model performance is experimentally evaluated. Firstly, it focuses on comparing the trends of accuracy rates of various methods to verify the ability of this paper's scheme to monitor class variable imbalance. Secondly, the accuracy, recall, precision and F1 values of the compared solutions are compared under different rounds of class variable imbalance to verify that this solution can effectively mitigate the degradation of model performance due to class variable imbalance. The experimental results in this section are all the expected values of 10 independent experiments.

4.2.1. Monitoring Method Validity

In this section, FCVI is compared with FedRL, K-SMOTE, and FedAvg to assess the ability of each method to monitor class variable imbalance. Specifically, class variable imbalance can be divided into two cases, class increasing imbalance and class decreasing imbalance, and this section compares the accuracy variation curves of these strategies under these two classes of variable imbalance, respectively.

Figure 2 shows the comparison of the accuracy change curves under class-increasing imbalance. After starting from the 50th round, the number of classes of the classification model will increase by one class every 50 rounds. When the number of classes increases, the accuracy of K-SMOTE and FedAvg decreases, and the accuracy of both FCVI and FedRL increases. In contrast, the accuracy curve of FCVI has a significant increase and is significantly higher than the other three methods. Figure 3 shows the comparison of the accuracy change curves under the class reduction imbalance. After starting from the 50th round, the number of classes of the classification model will decrease by one class every 50 rounds. When the number of classes is reduced, the accuracy of FedRL, K-SMOTE and FedAvg obviously decreases, while the accuracy of FCVI only slightly decreases and remains basically stable overall. This is because K-SMOTE and FedAvg are unable to process the increased and decreased classes, which affects the overall accuracy change. FedRL is able to process the increased and decreased classes, but does not monitor the change in the number of classes in a timely manner, resulting in the collation accuracy being affected. On the contrary, FECI can accurately monitor the changes in the category data by passing the training gradient parameters and timely expand them using unlabeled data, which ultimately has little impact on the collation accuracy and ensures the accuracy of the model.

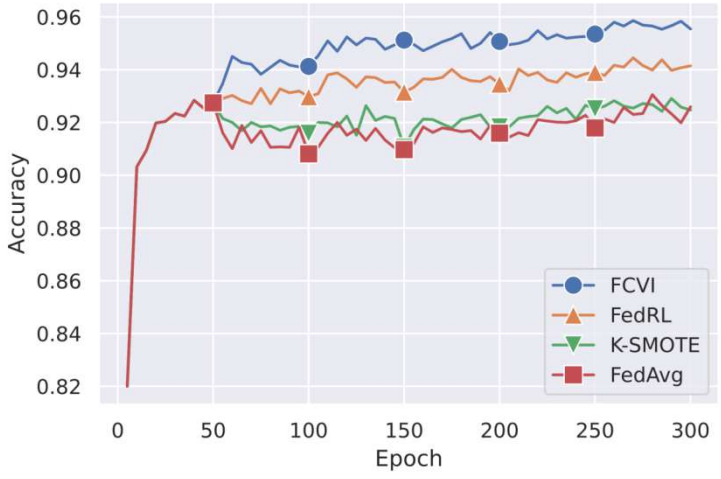


Figure 2 Comparison of accuracy change curves under class increasing imbalance.

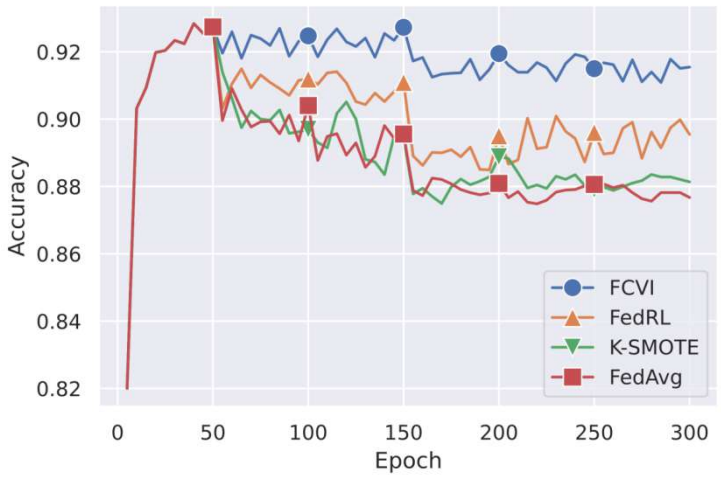


Figure 3 Comparison of accuracy change curves under class decreasing imbalance.

4.2.2. Model Performance Comparison

To evaluate the execution efficiency of FCVI, this section compares FCVI with FedRL, K-SMOTE, and FedAvg for different model training rounds under class-variable imbalance. In the first 50 rounds of training, the number of classes and class distribution of the training data does not change, and from the 50th round onward, the number of new or reduced classes and other classes are also randomly changed in the 50th, 100th, 150th, 200th, and 250th rounds, respectively. In this paper, we will compare the number of classes and class distribution of the training data after the occurrence of changes in the 50th round, the 100th, 150th, 200th, 250th, and 300th rounds, respectively. 200, 250, and 300 rounds of model performance comparison.

Table 2 Comparison of accuracy, recall, precision and F1 values for different epoch under class increasing imbalance

Epoch		100	150	200	250	300	Average	△
Accuracy	FCVI	**0.941**	**0.951**	**0.951**	**0.954**	**0.955**	**0.950**	—
	FedRL	0.930	0.932	0.935	0.939	0.941	0.935	⇑0.015
	K-SMOTE	0.916	0.911	0.918	0.925	0.925	0.919	⇑0.031
	FedAvg	0.908	0.910	0.916	0.918	0.926	0.916	⇑0.035
Recall	FCVI	**0.625**	**0.658**	**0.618**	**0.592**	**0.609**	**0.620**	—
	FedRL	0.618	0.606	0.583	0.562	0.548	0.583	⇑0.037
	K-SMOTE	0.544	0.510	0.467	0.468	0.442	0.486	⇑0.134
	FedAvg	0.519	0.479	0.451	0.448	0.419	0.463	⇑0.157
Precision	FCVI	**0.640**	**0.666**	**0.653**	**0.629**	**0.629**	**0.643**	—
	FedRL	0.620	0.615	0.592	0.574	0.550	0.590	⇑0.053
	K-SMOTE	0.567	0.543	0.492	0.483	0.474	0.512	⇑0.132
	FedAvg	0.702	0.512	0.473	0.459	0.448	0.519	⇑0.125
F1	FCVI	**0.624**	**0.657**	**0.622**	**0.596**	**0.610**	**0.622**	—
	FedRL	0.611	0.602	0.579	0.560	0.531	0.577	⇑0.045
	K-SMOTE	0.525	0.492	0.437	0.442	0.416	0.462	⇑0.159
	FedAvg	0.499	0.449	0.413	0.416	0.385	0.432	⇑0.190

Tables 2 and 3 show the four evaluation results of the classification model under different rounds of the method under class increasing imbalance and class decreasing imbalance, respectively, showing the model performance after 50 rounds of occurrence of changes in the number of classes, where '△' indicates the improvement of the method FCVI in this paper compared with other compared methods. It is observed that in class increase imbalance, the method in this paper outperforms other comparative methods in different training rounds, as well as in average accuracy, average recall, average precision, and F1 values of 0.029-0.0157, 0.037-0.125, and 0.030-0.190. Similarly, in class reduction imbalance, the method in this paper outperforms other comparison methods in different training rounds as well as has, 0.013-0.0076, 0.013-0.058, and 0.016-0.100 advantages in average accuracy, average recall, average precision, and F1 values.

Table 3 Comparison of accuracy, recall, precision and F1 values for different epoch underclass decreasing imbalance

Epoch		100	150	200	250	300	Average	△
Accuracy	FCVI	**0.925**	**0.923**	**0.926**	**0.920**	**0.915**	**0.922**	—
	FedRL	0.912	0.911	0.895	0.896	0.895	0.902	⇑0.020
	K-SMOTE	0.897	0.895	0.889	0.879	0.881	0.888	⇑0.034
	FedAvg	0.904	0.896	0.881	0.881	0.877	0.888	⇑0.034
Recall	FCVI	**0.629**	**0.662**	**0.646**	**0.709**	**0.718**	**0.673**	—
	FedRL	0.600	0.635	0.622	0.692	0.730	0.656	⇑0.017
	K-SMOTE	0.540	0.580	0.604	0.627	0.693	0.609	⇑0.064
	FedAvg	0.533	0.561	0.574	0.634	0.684	0.597	⇑0.076
Precision	FCVI	**0.631**	**0.673**	**0.647**	**0.710**	**0.722**	**0.676**	—
	FedRL	0.608	0.644	0.620	0.695	0.740	0.661	⇑0.015
	K-SMOTE	0.567	0.617	0.611	0.645	0.699	0.628	⇑0.049
	FedAvg	0.561	0.595	0.597	0.652	0.686	0.618	⇑0.058
F1	FCVI	**0.630**	**0.662**	**0.643**	**0.707**	**0.718**	**0.672**	—
	FedRL	0.596	0.628	0.614	0.689	0.728	0.651	⇑0.021
	K-SMOTE	0.516	0.561	0.584	0.607	0.678	0.589	⇑0.083
	FedAvg	0.500	0.536	0.547	0.612	0.662	0.572	⇑0.100

Based on the analysis of the above results, this verifies the effectiveness of this paper's method in alleviating the class-variable imbalance problem. It can be concluded that the method in this paper can effectively improve the performance of the model by aggregating servers and edge servers under FL gradient detection method and class variable learning algorithm so that the model can learn new class data and prevent forgetting old class data during the training process.

5. CONCLUSIONS

We proposed the Federated Semi-supervised Learning for Class Variable Imbalance (FCVI) method for solving the class variable imbalance problem in federation learning. Compared to other methods, FCVI does not require the collection of additional model parameters while mitigating the negative impact of class variable imbalance on model training. This work aims to mitigate the data imbalance problem caused by the changing number of classes through the federal gradient monitoring method and the class-variable learning algorithm. First the federal gradient monitoring method is used to monitor and locate the proportion of category changes, and then the data imbalance due to the number of category changes is mitigated by monitoring the resulting proportion of category changes and the class-variable learning algorithm. Experiments show that FCVI can guarantee the performance and execution efficiency of distributed training of deep learning models while adapting to the increase and decrease of the number of categories.

ACKNOWLEDGEMENTS

This work was supported by the basic scientific research expenses program of universities directly under inner mongolia autonomous region, grant/award number: jy20220273.

REFERENCES

[1] Networking, Cisco Visual. "Cisco global cloud index: Forecast and methodology, 2015-2020. white paper." Cisco Public, San Jose (2016): 2016.

[2] Ghosh, Ashish, Debasrita Chakraborty, and Anwesha Law. "Artificial intelligence in Internet of things." CAAI Transactions on Intelligence Technology 3.4 (2018): 208-218.

[3] Shi, Weisong, et al. "Edge computing: Vision and challenges." IEEE internet of things journal 3.5 (2016): 637-646.

[4] Li, Tian, et al. "Federated learning: Challenges, methods, and future directions." IEEE signal processing magazine 37.3 (2020): 50-60.

[5] Brisimi, Theodora S., et al. "Federated learning of predictive models from federated electronic health records." International journal of medical informatics 112 (2018): 59-67.

[6] Lee, Hae Beom, et al. "Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks." arXiv preprint arXiv:1905.12917 (2019).

[7] Fu, Guang-Hui, Lun-Zhao Yi, and Jianxin Pan. "Tuning model parameters in class-imbalanced learning with precision-recall curve." Biometrical Journal 61.3 (2019): 652-664.

[8] Yang, Miao, et al. "Federated learning with class imbalance reduction." 2021 29th European Signal Processing Conference (EUSIPCO). IEEE, 2021.

[9] Albaseer, Abdullatif, et al. "Exploiting unlabeled data in smart cities using federated edge learning." 2020 International Wireless Communications and Mobile Computing (IWCMC). IEEE, 2020.

[10] Jeong, Wonyong, et al. "Federated semi-supervised learning with inter-client consistency & disjoint learning." arXiv preprint arXiv:2006.12097 (2020).

[11] Zhai, Junhai, Jiaxing Qi, and Chu Shen. "Binary imbalanced data classification based on diversity oversampling by generative models." Information Sciences 585 (2022): 313-343.

[12] Naim, Forhad An, Ummae Hamida Hannan, and Md Humayun Kabir. "Effective rate of minority class over-sampling for maximizing the imbalanced dataset model performance." Proceedings of Data Analytics and Management: ICDAM 2021, Volume 2. Springer Singapore, 2022.

[13] Tan, Jingru, et al. "Equalization loss for long-tailed object recognition." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[14] Yang, Yuzhe, and Zhi Xu. "Rethinking the value of labels for improving class-imbalanced learning." Advances in neural information processing systems 33 (2020): 19290-19301.

[15] Guo, Lan-Zhe, and Yu-Feng Li. "Class-imbalanced semi-supervised learning with adaptive thresholding." International Conference on Machine Learning. PMLR, 2022.

[16] Kim, Jaehyung, et al. "Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning." Advances in neural information processing systems 33 (2020): 14567-14579.

[17] Wei, Chen, et al. "Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[18] Li, Tian, et al. "Federated optimization in heterogeneous networks." Proceedings of Machine learning and systems 2 (2020): 429-450.

[19] Ruan, Yichen, et al. "Towards flexible device participation in federated learning." International Conference on Artificial Intelligence and Statistics. PMLR, 2021.

[20] Duan, Moming, et al. "Self-balancing federated learning with global imbalanced data in mobile systems." IEEE Transactions on Parallel and Distributed Systems 32.1 (2020): 59-71.

[21] Wang, Lixu, et al. "Addressing class imbalance in federated learning." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35. No. 11. 2021.

[22] Amini, Massih-Reza, et al. "Self-training: A survey." arXiv preprint arXiv:2202.12040 (2022).

[23] Vedaldi, A., et al. "Convolutional architecture for fast feature embedding." Cornell University (2014).

[24] McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." Artificial intelligence and statistics. PMLR, 2017.

AUTHORS

Zehui Dong received North University of China, and he did Engineering Degree in Software Engineering. Currently, he is pursuing his Master's degree in Software Engineering from Inner Mongolia University of Technology. His research interests include edge computing, federated learning.

Wenjing Liu received the B.S. degree from Inner Mongolia University of Technology, Hohhot, China, in 2011, the M.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2018. She is a college lecturer in Inner Mongolia University of Technology. Her research interests include algorithm design and assessment and artificial intelligence.

Siyuan Liu received the B.S. degree in 2021 from Tianjin Gongye University, Tianjin, China. She is currently pursuing the M.S. degree in Inner Mongolia University, Hohhot, China. Her research interests include edge computing and big data technology.

Xingzhi Chen graduated from Chongqing Institute of Technology with a degree in Software Engineering. He is currently studying for a master's degree in software engineering at Inner Mongolia University of Technology. His research interest covers pattern recognition and single image rain removal.

AUTHOR INDEX

Yu Sun	*19,137,191,201,229*
Yujia Zhang	*19,93,201*
Zehui Dong	*257*