Proceedings of the
International Conference on Artificial Intelligence & Software Engineering, ICAISE 2023 Kochi

15-17 March 2023

COCHIN UNIVERSITY OF
SCIENCE AND TECHNOLOGY

Editor-In-Chief

Prof.(Dr.) Philip Samuel

Associate Editors

Dr. Bijoy Antony Jose
Dr. Jereesh A S
Dr. Shailesh S
Dr. Jeena Kleenankandy

Directorate of Public Relations and Publications *for*
Department of Computer Science, Cochin University of Science & Technology,
Kochi-682022, India

Proceedings of the
International Conference on Artificial Intelligence & Software Engineering, ICAISE 2023 Kochi

15-17 March 2023

Editor-In-Chief

Prof.(Dr.) Philip Samuel

Associate Editors

Dr. Bijoy Antony Jose
Dr. Jereesh A S
Dr. Shailesh S
Dr. Jeena Kleenankandy

Directorate of Public Relations and Publications *for*
Department of Computer Science, Cochin University of Science & Technology,
Kochi-682022, India

ICAISE 2023 Kochi
Proceedings of International Conference on Artificial Intelligence & Software Engineering 2023 Kochi
March 15-17, 2023

Published by
Directorate of Public Relation and Publications
Cochin University of Science And Technology
Kochi, India

For
Department of Computer Science
Cochin University of Science And Technology
Kochi, India
URL: http://dcs.cusat.ac.in/

Editor
Prof. (Dr.) Philip Samuel

ISBN : 978-81-961282-0-3

This book has been published from Camera ready copy/softcopy provided by the contributors. The authors are solely responsible for the contents/claims of the paper compiled in this volume. The publishers/editors do not take any responsibility for the same in any manner.

Printed at
Jose Printers
South Kalamassery, Ph : 2556324

MESSAGE FROM VICE-CHANCELLOR

Prof. (Dr.) K.N Madhusoodanan
Vice-Chancellor,
Cochin University of Science and Technology
Kochi - 682022, India

As we witness the rapid development of AI and software engineering, it is essential to have a platform that promotes collaboration and knowledge sharing among experts in these fields. The International Conference on Artificial Intelligence & Software Engineering (ICAISE 2023) Kochi is aimed to fulfill this need. Participants will have the chance to forge new partnerships that can drive progress in the fields of AI and software engineering. Moreover, this conference can serve as a launching pad for students who aspire to pursue a career in the field of AI. The exposure and knowledge gained from attending this conference can help students to develop a deeper understanding of the field, identify potential research areas, and establish connections with professionals in the industry.

I congratulate all researchers presenting their work at ICAISE 2023 and the team at Department of Computer Science, Cochin University of Science and Technology for setting the stage through this event. I hope this would be an enriching and rewarding experience to both the organizers and its attendees. Wishing you all the very best for a successful conference.

MESSAGE FROM PRO VICE-CHANCELLOR

Prof. (Dr.) P.G.Sankaran
Pro Vice-Chancellor,
Cochin University of Science and Technology
Kochi - 682022, India

Artificial Intelligence (AI) has emerged as one of the most transformative technologies of our time, and its impact on our society is becoming increasingly significant. AI is already revolutionising the way we live, work, and interact with each other, and its potential applications are endless. AI has led to the development of intelligent software systems that can learn from data, reason, and make decisions. These advancements have opened up immense opportunities for academic research as well. The International Conference on Artificial Intelligence and Software Engineering (ICAISE 2023) at Cochin University of Science and Technology would provide an excellent platform for researchers and practitioners working in the field of AI and SE to share their latest research findings and advancements. The conference is an opportunity to explore new ways in which AI can be applied to address real-world problems.

I applaud the organisers of this conference for their efforts in bringing together a diverse group of individuals from around the world to share their knowledge and expertise. May this event foster collaborations and drive innovation, thereby leading to new breakthroughs in the field.

International Conference on Artificial Intelligence & Software Engineering, ICAISE 2023, Kochi

EDITORIAL

With Artificial Intelligence (AI) techniques applied to text processing, a robot can understand and converse in human language. By processing images, a robot can see. With AI applied in software engineering, a robot can create another robot replacing human beings. Industrial robots that substitute assembly line workers, AI-driven product marketing and travel planning, disease and drug discovery, chatbots and virtual assistants, and self-driving cars that replace human drivers etc. are becoming increasingly popular.

The International Conference on Artificial Intelligence & Software Engineering (ICAISE) 2023 Kochi is a major interdisciplinary international conference that provides a platform for researchers, engineers, practitioners, and educators to meet, present, and discuss the latest research results, innovations, trends, experiences, and concerns across various application domains of Artificial Intelligence & Software Engineering. This year's conference featured four exciting themes: Intelligent Computing, Software and Distributed Systems, Device and Systems, and Medical and Health Informatics.

This proceeding includes the research and development activities presented at the International Conference on Artificial Intelligence & Software Engineering, ICAISE

2023, Kochi, organized by the Department of Computer Science, Cochin University of Science and Technology (CUSAT) during 15-17 March 2023 at CUSAT.

As the editor of the proceedings, I use this space to acknowledge the invaluable contributions of all plenary speakers including that of Dr. A K Menon memorial lecture who made this event memorable. I express my sincere gratitude to all organizing committee members, international program committee members, and the reviewers who reviewed the papers presented at the ICAISE 2023 Kochi conference. I thank the World Federation on Soft Computing (WFSC) for their technical sponsorship. The enthusiasm of my students and colleagues at Cochin University of Science and Technology was very much appreciable. I thank our esteemed Vice-Chancellor, Prof. (Dr.) K.N Madhusoodanan, and the authorities of Cochin University of Science and Technology, India for their support without which, the conference could not have been the success that it was.

The proceedings of the International Conference on Artificial Intelligence & Software Engineering, ICAISE 2023, Kochi, compress several years of rich experience in industry, teaching, and research in the areas of Artificial intelligence and Software engineering. This book may lead to technology transfers and practical thrusts that are beneficial for the mankind. I thank all authors of research papers who responded to the call for paper presentation, whose inputs contributed tremendously to the conference. Above all, thanks to almighty God for making this a reality.

Dr.Philip Samuel,
Program Chair - ICAISE 2023 Kochi,
Professor & Head,
Department of Computer Science,
Cochin University of Science and Technology
Kochi-682022, India

ICAISE 2023

International Conference on Artificial Intelligence & Software Engineering - Kochi

15 – 17 March 2023

THEME AREAS

- Intelligent Computing
- Software And Distributed Systems
- Devices And Systems
- Medical And Health Informatics

15 MARCH 2023

9:30 am-11:00 am

Dr. K. N. Madhusoodanan,
Vice Chancellor, CUSAT

Inauguration

Felicitation & A. K. Menon Endowment Scholarship Distribution

Dr. A. K. Menon Endowment Lecture

"Digital Revolution: Enablers for the Past and Future"

11:30 am - 1:00 pm

Dr. P. J. Narayanan
Director
International Institute of Information Technology,
IIIT Hyderabad

ICAISE 2023 | Keynote Speakers

Keynote: "From Grammar Inference to Semantic Inference"

16-March-2023
11.30 am-1.00 pm

Dr. Marjan Mernik
Professor, University of Maribor, Slovenia,
Editor-In-Chief Journal of Computer Languages,
Associate Editor Information Sciences, Applied Soft Computing, Swarm and Evolutionary Computation

Keynote: "Blockchain consensus and running validators for PoS networks"

16-March-2023
2.00 pm-3.00 pm

Mr. Vivek R
Infrastructure Software Engineer,
Chorus One

Keynote: "Enjoyable Programming"

17-March-2023
9.30 am-10.30 am

Dr. Jey Veerasamy
Director, Center for Computer Science Education & Outreach, Professor of Instruction,Department of Computer Science, Erik Jonsson School of Engineering & Computer Science,
University of Texas at Dallas, USA

Keynote: "Exit the Internet and enter the Metaverse?"

17-March-2023
10.30 am-11.30 am

Dr. Vinu Sherimon
College of Computing and Information Sciences,
University of Technology and Applied Sciences,
Muscat, Sultanate of Oman

Keynote: "AI and Intelligent Automation"

17-March-2023
12.00 pm-1.00 pm

Mr. Anbu Ponniah
Chief Architect, Automation
IBM India Software Labs
ISL Kochi

17 MARCH 2023
03:00 pm-04:00 pm
Valedictory and Felicitation to retiring General Chairs

Sri. Muralidharan K B

General Chair - ICAISE 2023 Kochi
Assistant Professor
Department of Computer Science,
Cochin University of Science and Technology

Dr. David Peter S

General Chair - ICAISE 2023 Kochi
Professor
School of Engineering,
Cochin University of Science and Technology

Session Chair: Mr. Sunil Balakrishnan
Chief Values Officer &
Global Head - Center Operations, UST

ICAISE 2023 KOCHI Committee

Programme Chair
Dr. Philip Samuel
Professor & Head
Department of Computer Science, CUSAT

Organizing Chair
Dr. Bijoy A Jose
Associate Professor
Department of Computer Science, CUSAT

Organizing Co-Chair
Dr. Jereesh A. S
Assistant Professor
Department of Computer Science, CUSAT

Finance Chair
Dr. Shailesh S
Assistant Professor
Department of Computer Science, CUSAT

Publicity Chair
Dr. Jeena Kleenankandy
Assistant Professor
Department of Computer Science, CUSAT

Programme Committee

- *Dr. José Valente de Oliveira*
  University of Algarve, Portugal
- *Dr.Lefteris (Eleftherios) Angelis*
  Aristotle University of Thessaloniki, Greece
- *Dr. Zbigniew Suraj*
  University of Rzeszów, Poland
- *Dr. Juan Manuel Carrillo de Gea*
  University of Murcia, Spain
- *Dr. Mathew J. Palakal*
  Indiana University, USA
- *Dr. Chanchal Roy*
  University of Saskatchewan, Canada
- *Dr. Marjan Mernik*
  University of Maribor, Slovenia
- *Dr. Boris Tudjarov*
  Technical University of Sofia, Bulgaria
- *Dr. Juan Flores*
  University of Oregon, USA
- *Dr. Sasikumar Punnekkat*
  Mälardalen University, Sweden
- *Dr. Wei-Chiang Hong*
  Asia Eastern University of Science and Technology
- *Mr. Sunil Balakrishnan*
  UST
- *Dr. G. Santhosh Kumar*
  Cochin University of Science and Technology
- *Dr. Madhu S. Nair*
  Cochin University of Science and Technology

Keynote Address Abstracts

Dr. A. K. Menon memorial lecture by Dr. P. J. Narayanan

Dr. P. J. Narayanan is the Director of IIIT, Hyderabad and a researcher in the areas of 3D vision, computational cameras, and parallel computing. He built the Virtualized Reality system in mid 1990s at the Carnegie Mellon University to capture 3D geometry and appearance of dynamic events. He also was an early adopter of GPUs for several Computer Vision and general computing tasks. He got his Bachelors (1984) from IIT Kharagpur, Masters and Ph.D. (1992) from the University of Maryland, all in Computer Science. He was a research faculty member at the Robotics Institute of CMU from 1992 to 1996 and headed the Vision and Virtual Reality groups of the Centre for Artificial Intelligence and Robotics, Bangalore from 1996 to 2000. He has been with IIIT Hyderabad from 2000 and has been its PG Coordinator, Dean of Research, and, from 2013, the Director. He was the President of ACM India from 2009 to 2014 and is involved with various activities of ACM such as Awards, Technology Policy, etc.

Keynote Title: "Digital Revolution: Enablers for the Past and Future"

Abstract: Advances in different aspects of computing – faster digital computers, memory, algorithms, software, communication speed, protocols, etc., – have brought about a Digital Revolution in the past 50 years. Our world has been transformed in many fundamental ways through the process of digitalization of almost everything, including entertainment, money, engineering, etc. Artificial Intelligence powered by Machine Learning over vast amounts of digital data has been bringing different types of advances in recent years. In this lecture, we will look at the digital journey briefly and speculate on what may be required to exploit these advances for indian society in the coming decades.

Keynote Address by Dr. Marjan Mernik

Marjan Mernik received the MSc and Ph.D. degrees in Computer Science from the University of Maribor in 1994 and 1998, respectively. He is currently a professor at the University of Maribor, Faculty of Electrical Engineering and Computer Science. He was a visiting professor at the University of Alabama at Birmingham, Department of Computer and Information Sciences. His research interests include programming languages, compilers, domain-specific (modeling) languages, grammar-based systems, grammatical inference, and evolutionary computations. He is a member of the IEEE, ACM, and EAPLS. He is the

Editor-in-Chief of the Journal of Computer Languages, as well as associate editor of the Applied Soft Computing Journal, Information Sciences Journal, and Swarm and Evolutionary Computation Journal. He is being named a Highly Cited Researcher for years 2017 and 2018. More information about his work is available at https://lpm.feri.um.si/en/members/mernik/.

Keynote Title: "From Grammar Inference to Semantic Inference"

<u>Abstract:</u> This keynote talk describes a research work on Semantic Inference, which can be regarded as an extension of Grammar Inference. The main task of Grammar Inference is to induce a grammatical structure from a set of positive samples (programs), which can sometimes also be accompanied by a set of negative samples. Successfully applying Grammar Inference can result only in identifying the correct syntax of a language. With the Semantic Inference, a further step is realised, namely, towards inducing language semantics. When syntax and semantics can be inferred, a complete compiler/interpreter can be generated solely from samples. Moreover, from a formal specification, many other tools (editor, compiler, debugger, test engine) can be generated automatically. To solve the problem of Semantic Inference successfully, the Genetic Programming approach was employed, which is a population-based evolutionary search. The first results were encouraging, and we were able to infer S-attributed, L-attributed and Absolutely Non-circular Attribute Grammars.

Keynote Address by Mr. Vivek R.

Vivek R. is a graduate of Model Engineering College, who has accumulated a wealth of experience working with various product-based startups and at ThoughtWorks. He is currently engaged in building and maintaining blockchain validators systems at Chorus One, with a strong focus on distributed systems, networks, and security. Vivek is passionate about exploring these domains and stays up-to-date with the latest trends and developments in the field.

Keynote Title: "Blockchain consensus and running validators for PoS networks".

<u>Abstract:</u> This presentation delves into the consensus mechanisms used in blockchain technology and the complexities involved in operating validator nodes. It begins with an overview of the fundamental concepts of blockchain, including cryptography concepts like hashing and asymmetric keys. The discussion then moves on to Sybil resistance, which leads to an exploration of various consensus mechanisms such as Proof of Work and Proof of Stake. Additionally, the topic of block finality is touched upon, which provides assurance

that past transactions cannot be altered. Different finality mechanisms are examined in this context.

After laying the groundwork with these concepts, the presentation provides an overview of blockchain frameworks like Substrate and the Cosmos SDK, which are used for implementation. The practical aspects of blockchain are also discussed, including key terminologies like clients, full nodes, and validators. Staking is explained, with emphasis on its various types and its practical value. The presentation also covers staking providers and the workings of the staking ecosystem. Ultimately, the presentation provides a comprehensive understanding of the staking mechanism and its conceptual underpinnings.

Keynote Address by Dr. Jey Veerasamy

Dr. Jey Veerasamy is a Professor of Instruction in Computer Science department, University of Texas at Dallas. He also runs the Center for Computer Science Education & Outreach. Dr. Jey completed Ph.D. in CS while working full-time for Nortel & Samsung. He worked in wireless telecom software for 16 years and joined UT Dallas in 2010 to focus on teaching. Dr. Jey organizes Summer Camps & FREE After School Clubs for the school students to learn programming in enjoyable manner & additional hands-on workshops for UTD students to strengthen their technical skills to get better jobs!

Keynote Title: "Enjoyable Programming"

Abstract: The virtual demonstration of "Enjoyable Programming" uses the p5js environment, which comprises a web page with a coding section and a virtual canvas area where the output is displayed or updated as one type in the code. The user-friendly interface is facilitated by a robust graphics library, thereby reducing the amount of coding required. A few examples can be accessed at https://editor.p5js.org/jeyv/sketches for review. The best part is that there is no need for any software installation, and all the necessary work is performed within a web browser.

Keynote Address by Dr.Vinu Sherimon

Dr. Vinu Sherimon is a seasoned educator with 25 years of teaching experience, including 7 years in the university setting and 18 years overseas. Currently, she serves as a faculty member at the University of Technology and Applied Sciences in Muscat, Sultanate of Oman. She has been a Co-Principal Investigator of a research project that developed a clinical decision support system for COVID-19 using clinical ontologies and teleconferencing for primary health centers and satellite clinics of the Royal Oman Police. Her research work has been published in various international journals, such as the International Journal of Research and Reviews in Artificial Intelligence and the International Journal of Computer Science Issues, and she has presented her papers at numerous international and national conferences.

Keynote Title: "Exit the Internet and enter the Metaverse?"

Abstract: A virtual environment or universe known as the “metaverse” allows users to engage in fully immersive interactions with each other and digital items. While there is not currently a single “metaverse”, there are several new virtual platforms that are attempting to develop this kind of experience. The idea of the metaverse is still developing, and as technology develops and consumer demand increases, it is likely that many more platforms and experiences will appear in the upcoming years.

The keynote will start with the history of Metaverse, followed by the underlying technologies of metaverse and their potency. Then the role of NFT (Non-fungible tokens) in Metaverse will be addressed. The applications and use cases of Metaverse and existing Metaverse platforms will be discussed further. The session will conclude with the pros, cons, and the challenges of Metaverse.

Keynote Address by Mr. Anbu Ponniah

Mr. Anbu Ponniah is an Open Group certified distinguished IT Specialist and a Chief Architect at IBM India Software Labs. He has over 26 years of IT experience leading product development and services teams, creating state of the art AI and automation products and solutions. As a mentor and innovator, he leads patent evaluation boards and professional certification boards within IBM.

Keynote Title: “AI and Intelligent Automation”

Abstract: Over the decades, automation has touched most industries – from the factory floor to banking transactions and oil refineries. But intelligent automation enables change at a whole new level. Artificial intelligence (AI) and automation – they become intelligent automation – alter the way humans and machines interact, in terms of how data is analyzed, decisions are made, and tasks and activities within a workflow or system are performed. This talk discusses recent advancements in the field of AI/ML that enable systems to analyze large bodies of operational information, recognize patterns from multiple sources, and execute accordingly.

CONTENTS

1. Genetic Algorithm For Double Roman Domination Problem 1
Himanshu Aggarwal; P. Venkata Subba Reddy

2. Cognitive Skills in Code Comprehension using Machine Learning Methods 7
Divjot Singh; Ashutosh Mishra; Ashutosh Aggarwal

3. Disease Diagnosis and Prediction using DNA sequencing with patterns in Biological Data 15
Banothu Ramji; Raju Bhukya

4. GranReg: Granger-Regression-based Inference of Gene Regulatory Network with Seasonal Differencing 23
Emma P M; Jereesh A S; Santhosh Kumar G

5. A Precise IFCIoT Platform for the Achievement of Semantic Interoperability in Healthcare Data Systems 29
Sony P; Siva Shanmugam G; Sureshkumar N

6. Exploring the potential of Ensemble Based Machine Learning Techniques in Polycystic Ovary Syndrome Diagnosis 35
Mohammed Usman F; Afrah Ayub; Thirumala Akash K; Nitesh Kumar T; Mohammed Riyaz Ahmed

7. sEMG Signal based Hand Gesture Recognition Using Machine Learning 41
Nithin Kurian; Dheeraj S Nair; Shailesh S

8. Network Science And Climate Dynamics - A Mini Review 49
Abhijith S Babu; Divya Sindhu Lekha

9. Multi Block Transformer for Malayalam Language Modeling 53
Rohit T P; Sasi Gopalan; Varsha Shaheen

10. Natural Language Interface In Malayalam For SQL Databases 57
Anjali Augestin ; Geethu Subramanyan C; Kavyadas; Ummu Habeeba P P; Visakh R

11. Strategies of power allocation for QoS parameters in NOMA for 5G wireless communications 63
Lekshmi Nair M; Neelakantan P C

Genetic Algorithm for Double Roman Domination Problem

Himanshu Aggarwal
Computer Science and Engineering
National Institute of Technology Warangal
Warangal, India
hacs21212@student.nitw.ac.in

P. Venkata Subba Reddy
Computer Science and Engineering
National Institute of Technology Warangal
Warangal, India
pvsr@nitw.ac.in

***Abstract*—A variety of domination concepts have been defined to provide better routing and defense strategies under different constraints. A double Roman dominating function (DROMDF) on a simple, undirected graph G is a function $g : V \to \{0, 1, 2, 3\}$ such that every vertex $x \in V$ with $g(x) = 0$ is adjacent to at least two vertices y_1, y_2 with $g(y_1) = g(y_2) = 2$ or a vertex z_1 with $g(z_1) = 3$. Also, a vertex p with $g(p) = 1$ is adjacent to at least one vertex q_1 with $g(q_1) \geq 2$. $\gamma_{dR}(G)$, the double Roman domination number of G, is the smallest possible weight of all possible DROMDFs of G. Determining double Roman domination number of a graph is known to be NP-hard. Hence in this paper, we propose a genetic algorithm based approach for solving double Roman domination problem in which three heuristic algorithms have been proposed and problem specific crossover operator and feasibility function has been developed. Effectiveness of the proposed meta-heuristic algorithm is tested on the random graphs generated using NetworkX Erdős-Rényi model, a popular model for graph generation. Experimental results show that the proposed meta-heuristic algorithm for solving double Roman domination problem gives a near optimal solution in reasonable time.**

***Index Terms*—Domination, Double Roman domination, NP-hard, Genetic algorithm**

I. Introduction

Claude Berge in 1958 initiated the study of domination in graphs [12]. The concept of domination has been an extensively researched topic and one of the fastest growing branches of graph theory with many applications in real life, viz. computer communication networks, social network theory, wireless sensor networks and modelling biological networks [12], [13].
Graphs $G(V, E)$ considered are undirected, connected and simple. Let $V(G)$ represent the vertex set and $E(G)$ represent the edge set of G. $N(v)$ represents the *open neighbourhood* of vertex v, $N(v) = \{u|(u, v) \in E\}$ and $N(v) \cup \{v\}$ represents the *closed neighbourhood* $N[v]$ of vertex v. We refer to [21] for undefined terminology and notations. For a graph G, a dominating set is defined as subset of vertices D such that each vertex of graph which is not in D is adjacent to at least one vertex in D. Domination problem of determining size of a smallest dominating set is known to be NP-hard [12] which implies no polynomial time algorithm exist for general graphs.

Roman domination was introduced in 2004, based on the defense strategies used to protect the the Roman Empire [15]. Since then different variants of Roman domination parameters have been introduced to protect the region under different constraints and are surveyed in [6], [7]. A function $h : V \to \{0, 1, 2\}$ on G with the condition that each vertex $u \in V$ with $h(u) = 0$ has a neighbor v with $h(v) = 2$ is known as the Roman dominating function of G. Optimization version of Roman domination problem is NP-hard [20].

In 2016, Beeler et al. introduced *double Roman domination* in [1]. A function $h : V \to \{0, 1, 2, 3\}$ of graph G such that each vertex $x \in V$ with $h(x) = 0$ is adjacent to at least two vertices y_1, y_2 with $h(y_1) = h(y_2) = 2$ or a vertex z_1 with $h(z_1) = 3$; also, a vertex p with $h(p) = 1$ is adjacent to at least one vertex q_1 with $h(q_1) \geq 2$ is known as double Roman dominating function (DROMDF) on graph G. $h(V) = \sum_{u \in V} h(u)$ is the weight of a DROMDF. The *double Roman domination number* $\gamma_{dR}(G)$ is the value of $\min_h\{w(h) : h$ is a DROMDF of $G\}$ minimum weight of a DROMDF on G [16].
Rest of the paper is organized into five consecutive sections, namely, literature review, genetic algorithm for double Roman domination, implementation using three heuristics, experiments and results and conclusion.

II. Literature Review

Optimization version of double Roman domination problem is defined as below.
Minimum double Roman domination problem (MDR)
Input : A graph G.
Output : $\gamma_{dR}(G)$.
MDR is known to be NP-hard [10] which implies no polynomial time algorithm exist for general graphs. MDR is NP-hard even for subclasses of bipartite graphs [9] but polynomial time solvable for cographs [17]. Different heuristic and meta-heuristic algorithms for solving domination problem have been proposed [3]–[5]. However to the best of our knowledge no efficient approximation algorithms exist, only one meta-heuristic algorithm has been proposed for solving Roman domination problem [14] and an approximation algorithm for MDR problem . However no meta-heuristic algorithms exist

for solving MDR problem, which motivated us to focus on this direction to solve different Roman variants. In this paper, we give a genetic algorithm based solution to solve MDR problem.

III. Genetic Algorithm for Double Roman Domination

In this section, first we describe the working of genetic algorithm.
Genetic algorithm is based on the principle of natural reproduction system. In this, two parent solutions from initial population participates in crossover & mutation and reproduce one child solution. Only the fittest solution will go to next generation [8].

Genetic algorithm starts with creating *initial population* which is a set of feasible solutions in the current generation. Best individuals (solution) are selected based on their fitness evaluated using *fitness function* of the underlying problem which in turn generates intermediate population. Next a problem specific genetic *crossover operator* is used to generate new solution from those solutions to get the best solution. After that *mutation operator* is applied to maintain diversity among population individuals. Eventually initial population is replaced with the new population. Next we list the parameters associated with a genetic algorithm.

A. Generating Parameters for Genetic Algorithm

- Initial population ($init\_pop$) : Genetic algorithm starts with generating initial population which is a set of feasible solutions.
- Number of solutions (num_sol) : This parameter indicates the size of initial population i.e., number of feasible solutions.
- Least Cost obtained in current population ($curr\_least\_val$) : It denotes the minimum cost among all the feasible solutions in the current generation.
- Optimal solution obtained in current population ($curr\_opt\_sol$) : It denotes the optimal solution among all the feasible solutions in the current generation.
- Intermediate population ($inter\_pop$) : After applying crossover operator and mutation a new intermediate population is obtained. Intermediate population will be the initial population for the next generation.
- Best solution ($best\_sol$) : Best solution is the optimal solution obtained as a result on termination of algorithm and and the algorithm terminates either there is no update on best_sol for consecutive number of generations or after the specified number of iterations.
- Optimal cost (opt_val) : It denotes the cost of the best solution.

B. Pseudocode for Genetic Algorithm

Algorithm 1 Genetic Algorithm for DRDP

```
Input: A simple, undirected graph $G$
Output: Best Approximate Solution
  Initialize $num\_sol$
  Create $init\_pop$
  $curr\_least\_val \leftarrow$ minimum cost on $init\_pop$
  $opt\_val \leftarrow curr\_least\_val$
  while termination condition do
      $inter\_pop \leftarrow$ Crossover and Mutation on $init\_pop$
      $init\_pop \leftarrow inter\_pop$
      $curr\_least\_val \leftarrow$ minimum cost on $init\_pop$
    if $opt\_val > curr\_least\_cost$ then
        $opt\_val \leftarrow curr\_least\_cost$
        $best\_sol \leftarrow curr\_opt\_sol$
    end if
  end while
  return best_sol
```

IV. Implementation using three heuristics

We propose three heuristic methods to generate initial population for the genetic algorithm to solve MDR problem. The three heuristics are explained in detail in the following sections.

A. Heuristic 1

In this heuristic, we select a vertex v randomly from the graph, assign it label 3 and all its neighbors are assigned label 0. If the randomly selected vertex v has no neighbors then it is assigned label 2. Next, the vertex v and its neighbors if any are removed from the graph and the process is repeated for the remaining graph until all vertices are labelled.
Pseudocode for Heuristic 1 is given in Algorithm 2 and its working is illustrated with the help of a graph in figure 1.

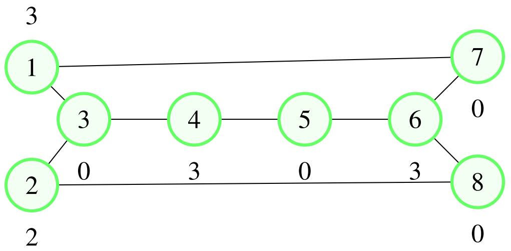


Fig. 1: A graph G labelled by Heuristic 1

3	2	0	3	0	3	0	0

Fig. 2: Solution Representation

A graph $G(V, E)$ with vertex set $\{1, 2, \ldots, n\}$ is provided as input to Algorithm 1. Next $S[1 \ldots n]$ is declared as a solution array and V' is initialize as V. A random vertex u from V' is picked and 3 is assigned to $S(u)$ and 0 to all its neighbours v i.e, $S(v)$ to 0. Next all those vertices which are labelled are removed from V'. If a single vertex u remains then 2 is assigned to $S(u)$ and u is removed from V'. It is easy to verify that S is a DROMDF of G with fitness value i.e., weight $\sum_{k=1}^{n} S(k)$.

For the graph shown in figure 1, $V' = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Suppose that Heuristic 1 picks vertex 4 randomly from V'. Vertex 4 is assigned label 3 i.e., $S(4) = 3$ and its neighbours are assigned label 0 i.e., $S(3) = 0$ and $S(5) = 0$. Now, we have $V' = \{1, 2, 6, 7, 8\}$. If 1 is the next vertex to be picked then set $S(1) = 3$ and assign 0 to its neighbour 7 i.e., set $S(7) = 0$, so $V' = \{2, 6, 8\}$. If 6 is the next vertex to be picked then set $S(6) = 3$ and assign 0 to its neighbour 8 i.e., set $S(8) = 0$, so $V' = \{2\}$. Finally, we pick the reaming vertex 2 and assign it label 2 i.e., $S(2) = 2$ as 2 as no neighbours. Now $V' = \phi$. It is easy to verify that the labelling is a valid DROMDF of G with weight 11. We return S shown in figure 2 as a feasible solution.

Algorithm 2 Pseudo code for Heuristic 1

```
Input: A simple and undirected graph $G$
Output: Feasible Solution $S$
  Declare $S$
  $V'$ = $V$
  while $V' \neq \phi$ do
      Select a vertex $u$ randomly from $V'$ :
      $S(u)$ = 3
      for all vertex $v \epsilon N(u)$ do
          $S(v)$ = 0
      end for
      $V'$ = $V' \backslash N[u]$
      if $|V'|$ = 1 then
          $S(u)$ = 2 such that $u \epsilon V'$
          $V'$ = $V' \backslash \{u\}$
      end if
  end while
  return $S$
```

B. Heuristic 2

This heuristic is similar to Heuristic 1 except for the fact that after removing the randomly selected vertex and its neighbors if the remaining graph has any isolated i.e., degree 0 vertices they are assigned label 2 and are also removed. The process is repeated for the remaining graph until all vertices are labelled.
Pseudocode for Heuristic 2 is given in Algorithm 3 and its working is illustrated with the help of a graph in figure 3.

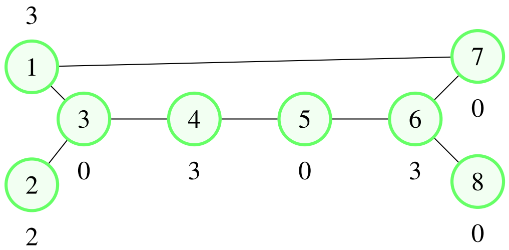


Fig. 3: A graph G labelled by Heuristic 2

3	2	0	3	0	3	0	0

Fig. 4: Solution Representation

For the graph shown in figure 3, $V' = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Suppose that Heuristic 2 picks vertex 4 randomly from V'. Vertex 4 is assigned label 3 i.e., $S(4) = 3$ and its neighbours are assigned label 0 i.e., $S(3) = 0$ and $S(5) = 0$. Now, we have $V' = \{1, 2, 6, 7, 8\}$. Since vertex 2 is the only isolated vertex in the remaining graph, it is labelled as 2 and V' is updated as $V' = \{1, 6, 7, 8\}$. If 1 is the next vertex to be picked then set $S(1) = 3$ and assign 0 to its neighbour 7 i.e., set $S(7) = 0$, so $V' = \{6, 8\}$. Since there are no isolated vertices in the remaining graph, if 6 is the next vertex to be picked then set $S(6) = 3$ and assign 0 to its neighbour 8 i.e., set $S(8) = 0$. Now $V' = \phi$. It is easy to verify that the labelling is a valid DROMDF of G with weight 11. We return S shown in figure 4 as a feasible solution.

Algorithm 3 Pseudo code for Heuristic 2

```
Input: A simple and undirected graph $G$
Output: Feasible Solution $S$
  Declare $S$
  $V'$ = $V$
  while $V' \neq \phi$ do
      Select a vertex $u$ randomly from $V'$ :
      $S(u)$ = 3
      for all vertex $v \epsilon N(u)$ do
          $S(v)$ = 0
      end for
      $V'$ = $V' \backslash N[u]$
      for all vertex $u \epsilon V'$ do
          if $deg(u)$ = 0 then
              $S(u)$ = 2
              $V'$ = $V' \backslash \{u\}$
          end if
      end for
  end while
  return $S$
```

C. Heuristic 3

In this heuristic, we take the advantages of Heuristics 2 and the degree of a vertex. Here, we select a vertex v with maximum degree from the graph by breaking ties arbitrarily. Selected vertex v is assigned label 3 and all its neighbors are assigned label 0. Next, the vertex v and its neighbors if any are removed from the graph. All isolated vertices of the remaining graph are assigned label 2 and these isolated vertices are removed from the remaining graph. The process is repeated for the new graph until all vertices are labelled. Pseudocode for Heuristic 3 is given in Algorithm 4 and its working is illustrated with the help of a graph in figure 5. Algorithm 4 is similar to Algorithm 3 except for the fact that vertex with highest degree is selected after sorting the degrees of vertices of G in non-decreasing order.

For the graph shown in figure 5, $V' = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Clearly vertex 6 has the maximum degree. Hence vertex 6 is selected and assigned label 3 and its neighbors 5, 7 and 8 are assigned label 0 i.e., $S(6) = 3,\ S(5) = 0,\ S(7) = 0$ and $S(8) = 0$. Now, we have $V' = \{1, 2, 3, 4\}$. Since vertex 1 is the only isolated vertex in the remaining graph, it is labelled as 2 and V' is updated as $V' = \{2, 3, 4\}$. Since 3 is the highest degree vertex in the remaining graph it is assigned label 3 and its neighbors 2 and 4 are assigned label 0 i.e., $S(3) = 3,\ S(2) = 0$ and $S(4) = 0$. Now $V' = \phi$. It is easy to verify that the labelling is a valid DROMDF of G with weight 8. We return S shown in figure 6 as a feasible solution.

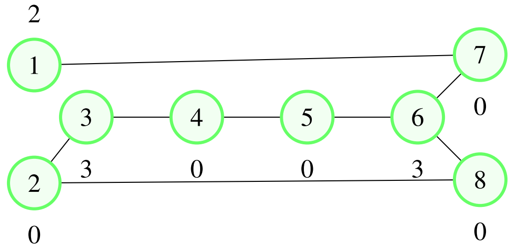


Fig. 5: A graph G labelled by Heuristic 3

2	0	3	0	0	3	0	0

Fig. 6: Solution Representation

Algorithm 4 Pseudo code for Heuristic 3

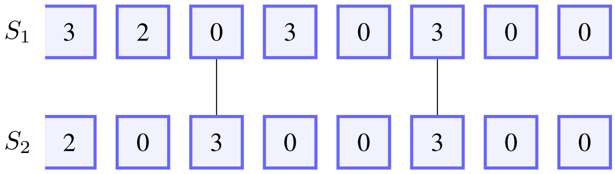
```
Input: A simple and undirected graph $G$
Output: Feasible Solution $S$
  Declare $S$
  $V'$ = $V$
  Sort $V'$ in descending order of their degree
  while $V' \neq \phi$ do
      Select first vertex $u$ from $V'$ such that :
      $S(u)$ = 3
      for all vertex $v \epsilon N(u)$ do
          $S(v)$ = 0
      end for
      $V'$ = $V' \backslash N[u]$
      for all vertex $u \epsilon V'$ do
          if $deg(u)$ = 0 then
              $S(u)$ = 2
              $V'$ = $V' \backslash \{u\}$
          end if
      end for
  end while
  return $S$
```

D. Selection operator

Selection operator is applied to select best possible candidates for crossover and mutation. Initial population i.e approximately 1000 feasible solutions is generated using three heuristics defined above for obtaining a feasible DROMDF of given graph. Next, we select two solutions among 1000 solution for performing crossover operator. Select the first solution using tournament selection operator i.e solution with minimum objective value and second using roulette wheel i.e randomly select a solution out of remaining 999 solutions. After getting two solutions crossover and mutation operations are performed.

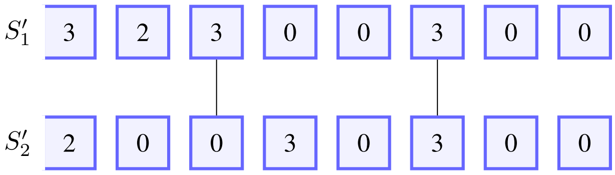
E. Crossover and Mutation Operator

Crossover and Mutation are the two important operation in Genetic Algorithm. Here, we use the two point crossover operation. First, we select two random vertices in first solution and swap the label values between those vertices of first solution with other solution which results in two child solution which may or may not be feasible. Next, we perform the feasibility test to make them feasible. Best two solutions S_1, S_2 are shown in figure 7 and the two child solutions S_1', S_2' generated after two point crossover are shown in figure 8. Clearly S_1' is feasible but not S_2' as zero labelled vertices 2 is not adjacent to vertices with label 3. Next S_2' goes through the feasibility check, zero labelled vertices 2 are assigned label 2 as shown in figure 9.


S_1 3 2 0 3 0 3 0 0

S_2 2 0 3 0 0 3 0 0

Fig. 7: Selected solutions S_1 and S_2


S_1' 3 2 3 0 0 3 0 0

S_2' 2 0 0 3 0 3 0 0

Fig. 8: Solutions S_1' and S_2' after crossover

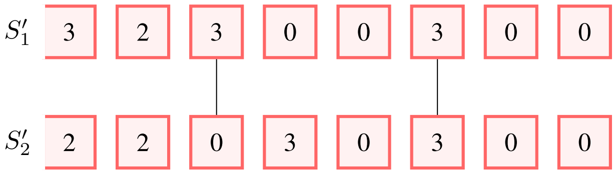
Algorithm 5 Crossover

```
Input: Two random Solutions $S_1$ and $S_2$
Output: Child Solution with better objective value
  Select two random vertices $v_1$ and $v_2$ from $V$.
  $X \leftarrow$set of vertex weights between $v_1$ and $v_2$ from $S_1$
  $Y \leftarrow$set of vertex weights between $v_1$ and $v_2$ from $S_2$
  Swap $X$ and $Y$ in $S_1$ and $S_2$
  $S_1' \leftarrow S_1$
  $S_2' \leftarrow S_2$
  feasible($S_1'$)
  feasible($S_2'$)
  return ($S_1'$, $S_2'$)
```

Algorithm 6 Feasibility check

```
Input: Child Solution $C$
Output: Modified feasible child solution

  for all vertex $u \epsilon C$ do
      if $S(u) = 0$ && $\nexists v \epsilon N(u)$ such that $S(v) = 3$ then
          $S(u) \leftarrow 2$
      end if
  end for
```


S_1' 3 2 3 0 0 3 0 0

S_2' 2 2 0 3 0 3 0 0

Fig. 9: Solutions S_1' and S_2' after attaining feasibility

V. Experiments and Results

Implementation details and the results obtained are discussed in this section. Proposed genetic algorithm is implemented in Python Language and all the experiments are performed on 11$^{\text{th}}$ Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz -1.38 GHz system with 8GB RAM. Performance of the proposed genetic algorithm is tested on random graphs generated using Erdös-Rényi model as it is used by most of the researchers working in this domain [18]. In [16] double Roman domination number of paths and cycles has been obtained and is shown in Table I.

We first tested the performance of the proposed genetic algorithm on path and cycle graphs for $15-100$ vertices. The results obtained are given in Table II. It is observed that as the number of iterations i.e., generations is increased the answer given by our algorithm is close to the optimal solution. Next, the performance of the proposed genetic algorithm is tested on random graphs upto 500 vertices generated using Erdös-Rényi model and shown in Table IV.

Graph	n	$\gamma_{dR}(G)$	**Obtained Results**		
			# of iterations		
			10000	100000	1000000
Path	15	15	15	15	15
	32	33	34	33	33
	68	69	75	72	70
	100	101	115	112	108
Cycle	15	15	15	15	15
	32	32	33	32	32
	67	68	76	73	71
	100	100	114	112	110

TABLE I: Compare optimal results ($\gamma_{dR}(G)$) of above graphs having n vertices with results obtained from Genetic Algorithm.

Since optimal solution for these graphs is not known and no meta-heuristic algorithm has been proposed in the literature to solve MDR problem in the literature, we have considered the upper bounds [17] and lower bounds [11] available and found that obtained results are within the bounds. Proposed genetic algorithm is run for 100000 iterations. The fact that obtained results are close to the lower bounds shows that the proposed algorithm is efficient.

Graph	**Optimal Results**
Path	n, if $n \equiv 0 \pmod 3$
	$n+1$, if $n \equiv 1$ or $2 \pmod 3$
Cycle	n, if $n \equiv 0,2,3,4 \pmod 6)$
	$n+1$, if $n \equiv 1,5 \pmod 6$

TABLE II: Some Standard Graphs with known $\gamma_{dr}(G)$

Graph Parameters	**Lower Bound**	**Upper Bound**
$\lvert V(G)\rvert, \delta, \Delta$	$\lceil\frac{3\lvert V(G)\rvert}{\Delta+1}\rceil$	$\frac{3\lvert V(G)\rvert(1+\ln\frac{2(1+\delta)}{3})}{1+\delta}$

TABLE III: Lower and Upper Bounds on $\gamma_{dr}(G)$

n	p	**LB**	**UB**	**Obtained Result**
25	0.2	9	49	18
	0.5	5	24	9
	0.8	4	16	5
50	0.2	9	60	21
	0.5	5	31	9
	0.8	4	19	6
75	0.2	9	70	26
	0.5	5	30	11
	0.8	4	20	6
100	0.2	9	82	29
	0.5	5	33	12
	0.8	4	21	6
200	0.2	12	95	38
	0.4	6	46	20
	0.5	5	37	15
300	0.2	12	93	45
	0.4	7	49	21
	0.5	6	38	18
400	0.2	12	105	47
	0.4	7	50	24
	0.5	6	40	18
500	0.2	12	96	48
	0.4	7	52	24
	0.5	6	42	20

TABLE IV: Results obtained for some random graphs.

VI. Conclusion

In this paper, we have proposed a genetic algorithm based solution for solving double Roman domination problem which is NP-hard. Since no meta-heuristic algorithm has been proposed in the literature for this problem, effectiveness of the proposed meta-heuristic algorithm is tested on the random graphs generated using NetworkX Erdős-Rényi model, a popular model for graph generation and found that the results obtained are close to the known lower bounds for the problem. This fact emphasizes that the algorithm is efficient. Designing better meta-heuristic algorithms for the problem remains open.

References

[1] R.A. Beeler, T.W. Haynes, S.T. Hedetniemi, *"Double Roman domination,"* Discrete Applied Mathematics, **211**, 23–29 (2016)

[2] Q. Cai, N. Fan, Y. Shi and S. Yao, *"Integer linear programming formulations for double roman domination problem"*, Optimization Methods and Software (2022) 37(1):1-22.

[3] Y. Wang, S. Cai, J. Chen and M. Yin, *"A Fast Local Search Algorithm for Minimum Weight Dominating Set Problem on Massive Graphs,"* Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 13-19 July 2018, Stockholm, Sweden.

[4] S. N. Chaurasia and A. Singh, *"A hybrid evolutionary algorithm with guided mutation for minimum weight dominating set,"*, Appl Intell (2015) 43:512–529

[5] C.N. Giap and D.T. Ha, Parallel genetic algorithm for minimum dominating set problem, International Conference on Computing, Management and Telecommunications, 27-29 April 2014, Da Nang, Vietnam.

[6] M. Chellali, N.J. Rad, S. M. Sheikholeslami, and L. Volkmann, Varieties of Roman Domination II, 17(3), 966–984, 2020.

[7] M. Chellali, N.J. Rad, S. M. Sheikholeslami, and L. Volkmann *Varieties of Roman Domination*, In: T. W. Haynes, S, T. Hedetniemi, M. A. Henning, (eds) Structures of Domination in Graphs, Developments in Mathematics, Springer, Cham, 66, 2021.

[8] J.H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, 1992.

[9] C. Padamutham, and V. S. R. Palagiri *"Complexity of Roman 2-domination and the double Roman domination in graphs,"* AKCE International Journal of Graphs and Combinatorics 2020, 17(3), 1081–1086.

[10] S. Banerjee, M. A. Henning, D. Pradhan, *"Algorithmic results on double Roman domination in graphs",* Journal of Combinatorial Optimization (2020) 39:90–114.

[11] D. R. Poklukar and J. Žerovnik *"Double Roman Domination: A Survey"* Mathematics 2023, 11, 351

[12] T. W. Haynes, S. Hedetniemi, and P. Slater, *"Fundamentals of domination in graphs,"* CRC Press, (1998).

[13] Haynes, T.W., Hedetniemi, S., Slater, P.: Domination in graphs : advanced topics, Markel Dekker, Inc, New York, (1998).

[14] A. Khandelwal, K. Srivastava, and G. Saran, *"On Roman domination of graphs using a genetic algorithm,"* Advances in Intelligent Systems and Computing 1227, (2021).

[15] E. J. Cockayne, P. A. Dreyer, S. M. Hedetniemi, and S.T. Hedetniemi. *"Roman domination in graphs,"* Discrete Math 278 (2004) 11-22.

[16] V. Anu, and S. A. Lakshmanan, *"Double Roman domination number,"* Discrete Applied Mathematics 244, (2018), 198-204.

[17] J. Yuea, M. Wei, M. Li, and G. Liu *"On the double Roman domination of graphs,"* Applied Mathematics and Computation 338, (2018), 669-675.

[18] L. A. Sanchis, *"Experimental analysis of heuristic algorithms for the dominating set problem,"* Algorithmica (2002) 33: 3–18

[19] E. Alvarez-Miranda, and M. Sinnl, *"Exact and heuristic algorithms for the weighted total domination problem,"* Computers and Operations Research 127 (2021) 105157

[20] M. Liedloff, T. Kloks, J. Liu, and S.H. Peng, *"Roman domination in some special classes of graphs,"* Report TR-MA-04-01, November 2004

[21] D.B. West, Introduction to graph theory, Vol. 2, Upper Saddle River: Prentice Hall (2001)

Cognitive skills in Code Comprehension using Machine Learning Methods

Divjot Singh
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology
Patiala, India
dsingh_phd21@thapar.edu

Ashutosh Mishra
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology
Patiala, India
Ashutosh.mishra@thapar.edu

Ashutosh Aggarwal
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology
Patiala, India
Ashutosh.aggarwal@thapar.edu

Abstract

The modern era of technology is filled with latest upgrades and new advancements. Every day, something new is being invented and used from local to global scale. Furthermore, everything is now software dependent, thus it has become necessary that the software remains up-to date with latest upgrades, features and is free from bugs or any other complications. Software maintenance is a crucial step of Software Development Life Cycle (SDLC). To maintain the software, it is necessary that the developers or the programmers handling the software are able to comprehend its source code in a time and cost-efficient way. The comprehension of software code requires good programming and cognitive skills. The researchers have worked on the various techniques like fMRI, eye tracking, EEG signals, code comprehension tasks, etc. to understand the cognitive skills and behavioural features which include cognitive domains like emotions, psychology and Belief-Desire-Intention (BDI). The main aim of our work is to study the various cognitive processes and programming skills of programmers by analysing the various techniques used by the researchers in their work. The study concludes that coding tasks (31%) are the preferable methods for testing the programming and cognitive skills. Time measurement and correctness of response are some of the crucial performance parameters that have been used in numerous researches. The study also concludes that the methods like machine learning and deep learning are most reliable while doing computational work.

Keywords: Software maintenance, BDI, emotions, psychology, code comprehension

1. INTRODUCTION

The computer software and hardware industries have seen revolutionary changes during the past few decades. The paradigm alters rapidly because of the constant innovations and new technologies. With the collapse of a software, a corporation can quickly lose millions or even billions of dollars, thus it is important to stay on top of the latest developments. It has become necessary to maintain the software that we have been working on. Software maintenance is a significant phase in the Software Development Life Cycle model. It is the programmer's obligation to fix any bugs or improve the software so that it can be used effectively.

According to [1], the job of maintaining software is mentally taxing and demands a high IQ. Consequently, it is instructive to study how a programmer deals with software-related issues on a regular basis. Given that a programmer needs to understand the code in order to fix an issue, it is clear that computer programming is the primary component in most software. Understanding source code calls for a wide range of technical and mental abilities. The cognitive qualities that define a programmer are their capacity for abstract thought, memory, and analysis. When considering a programmer's feelings, the importance of this research increases. Problems arise and are resolved at different stages of software maintenance, much like human emotions. Also of importance is the thought process that went into developing this psychology and the way they plan to tackle the problem. In the process of software upkeep, it's vital to take into account the programmer's motivations. His drive and their resolve to find a solution to the problem's intricacy. For the software to be maintained and bugs fixed as quickly as feasible at minimal cost and effort, other cognitive abilities like goals and wishes are crucial. The degree to which a programmer is involved in software maintenance is determined by the Belief-Desire-Intention (BDI) paradigm.

The motivation for our work is to understand and analyse the cognitive skills of the programmers along with the programming skills. Programming skills are the output of the programmers but cognitive skills define how that logic has been achieved. The study of cognitive psychology computes the analytical power of the programmers. The logics and the approaches they take, define the experience of the programmers. Similarly, while debugging a code snippet, the comprehension of each line of the code varies. This triggers various emotional states in the programmers. From feeling confused to frustrated and happy, the emotions play a vital role as a cognitive skill. Sometimes, it becomes difficult to remove a bug and the programmer feels low, at that time it becomes crucial that the programmers stay motivated and use their programming skills and beliefs to stay focused on the task. We have assumed that the participating programmers are well versed with the programming skills, hence we have devoted our attention towards the cognitive skills along with the programming skills.

1.1. Importance of code comprehension in Software Maintenance

The process of making changes to a software product after it has been released and distributed globally is known as software maintenance, and it is an integral part of the Software Development Life Cycle [2]. Bug fixes and performance optimizations are the main foci of post-release software maintenance and development. This is a huge happening right after the construction is finished. It improves the system's performance by eradicating errors, removing unnecessary code, and adding in new, state-of-the-art features. There are four distinct categories of software maintenance mentioned in[3], each of which serves a unique function.

- Adaptive Maintenance
- Corrective Maintenance
- Preventive Maintenance
- Perfective Maintenance

Despite software maintenance being a significant component of the software development process, bugs and other software malfunctions will likely remain in the product. Hence, optimizing the source code and releasing frequent updates and patches for the software are essential. Comprehending the code of a computer program is the process of learning the program's features and how they work. The programmer has an obligation to quickly diagnose the cause of any software issues and provide a fix because of the high value placed on both time and money. The outcome depends on how quickly the programmer can read and comprehend the code.

Understanding the code is a crucial part of software upkeep. The emotional states of programmers are also tested alongside, giving an insightful view point of the identification of behavioural data. Keeping the complexity of the code within reasonable bounds aids in its readability. A developer should strive to write code that is both basic and easy to understand for future reference, both for the sake of understanding and debugging.

1.2 Cognitive Skills

For source code comprehension, cognitive abilities are crucial. A programmer must comprehend the source code to troubleshoot it. To address this, programmers must use critical thinking, analytical power, and reasoning in a short period. Defines the programmer's cognitive psychology. While debugging source code, the programmer may be time-constrained. Pressure can affect a programmer's emotions. A programmer's emotions depend on the situation. Emotional stages affect cognitive skills and comprehension. Anger and irritation can affect programmers' judgement and decision-making while troubleshooting an error. The comprehension process also requires that the programmer believe the source code contains an error and that by amending it, the problem may be fixed. It's learned knowledge. Beliefs lead to wants, such as debugging code and solving the problem. Intentions are their dedication, attention, and sincerity to achieve their goals. All of them involve a programmer's brainpower.

1.2.1 Emotions

Emotions are mental states brought on by neurophysiological changes; they are associated with various modes of cognition, sensibility, and behavior, as well as with varying degrees of satisfaction and dissatisfaction. One's temperament, personality, and capacity for original thought are frequently connected with their state of mind, or emotional state. During the process of updating the software, there will frequently come a time when the programmer is unable to comprehend the issue at hand and will display a wide range of emotional states, including happiness, sadness, surprise, fear, and wrath, amongst others. It's interesting to compare how well programmers do when their mood is either cheerful or sad and they're given the task of understanding some source code. Will it have an impact on their cognitive abilities? It is fascinating to gain an understanding of the diverse ways in which a person's brain operates when they exhibit a range of emotional states while carrying out a task.

1.2.2 Cognitive Psychology

Psychology that focuses on the way our minds work is called cognitive psychology. Perception, logic, memory, focus, language, problem-solving, and learning are just some of the many cognitive functions that are studied in cognitive psychology. If scientists ever want to decipher the workings of the human brain, they will need a much fuller picture of the way people think and take in information. The code can be learned using either a top-down or bottom-up approach. Top-down approaches involve reading and analyzing the code from the top down. To rephrase, this is how a novice programmer would approach the problem. The developer employs a bottom-up approach, which entails looking for already-implemented features in the code before making any judgments about the end result. A professional programmer would address problems in this way. A programmer's viewpoint is typically at the heart of these techniques. Tests of logic, analysis, problem-solving, spatial awareness, critical thinking, and deductive reasoning, on the other hand, can reveal a great deal about a person's attention span, memory, and processing speed. In sum, understanding source code and the cognitive demands of programming push the limits of a programmer's knowledge and skill.

1.2.3 Belief-Desire-Intention (BDI)

BDI software simulates the human mind's cognitive processes in artificial intelligence (AI). The BDI theory of mind explains how humans think and make choices. The BDI framework includes beliefs, desires, and intentions.

- **Belief** is a programmer's knowledge of a software's programming language. It's the rules that define a programming language's syntax and semantics. The phrase belief doesn't mean that what the programmer has learned or believes is true.
- **Desire** drives programmers. It depicts programmer goals or conditions. Programmers' major purpose is to construct source code modules that constitute a software so they can fix problems. Their knowledge or belief fuels their desire.
- **Intention** represents programmers' choices. Intentions are programmer goals. The programmer has begun executing a plan in implemented systems. Intentions fulfil wants. Programmers commit to their work utilizing belief, desire, and intention when given a task. When a defect is found in the source code, programmers' major goal is to debug it. How they achieve this goal utilizing numerous parameters reveals their comprehension intentions.

1.3 Approaches in Code Comprehension

The programmers comprehend several code snippets or modules and use their several cognitive skills while understanding and debugging the code. It is important to understand how the programmers utilise the various skills like emotions, psychology and BDI for code comprehension. The researchers have used various kinds of approaches to understand and observe how the code comprehension is done by the programmer. The researchers have tried to understand the working of a programmer’s brain.

⇒ **Coding tasks and Interviews**- The cognitive skills and programming skills of the programmers are tested using various coding tasks which include code reading, code manipulation, code writing, code debugging, etc. The obtained results are then corroborated in interview sessions.

⇒ **EEG**- The EEG signals are electrical signals which are generated when the neurons of brain are activated. In this way, researchers can conclude which areas of the brain become active.

⇒ **fMRI**- What's better if a person can look inside a brain and see how a brain work. fMRI provide brain images so that it can be identified which parts of the brain gets activated when a cognitive or a programming task is done.

⇒ **Eye tracking**- When a programmer debugs a code or tries to comprehend a code, it is interesting to know where the programmer gazes the most or where he moves his eyes. Eye trackers track the eye movement and provides the pattern of tracking which are analysed by the researchers.

Machine learning and deep learning techniques are being used with the datasets that are being collected from these methods and predictions are made based on these datasets. Machine learning models can be used to classify the level and experience of programmers, while the deep learning techniques can be used to extract features from fMRI, PET scan data images.

2. RESEARCH WORKFLOW

To perform a literature survey of the existing work, a systematic workflow is required [4,5,6]. The sources of information, search criteria and inclusion-exclusion of the articles are some of the important steps to perform a literature review as shown in fig 1.

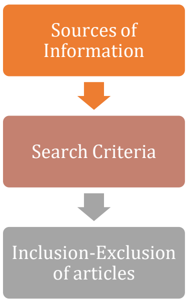


Fig.1 Research Workflow

2.1 Sources of Information

Extensive literature coverage requires a global lens. The most recent information about a study can be found in online research publications. The most important academic journals and reviews are compiled here. Duplicate papers found in the course of the search across the chosen sources were removed manually.

- Science Direct
- ACM Digital Library
- IEEE Explore

2.2 Search Criteria

For searching articles, several different terminologies and metonyms have been used. 'OR', 'AND', and other suitable operators are used to enhance the search criteria. Some search inputs are mentioned below:

- Code comprehension in software maintenance using machine learning
- Code comprehension in software maintenance using deep learning
- Program comprehension in software maintenance
- Emotions of a developer in project development
- Psychology of programmer in the development of projects
- BDI and software maintenance

2.3 Inclusion-Exclusion of articles

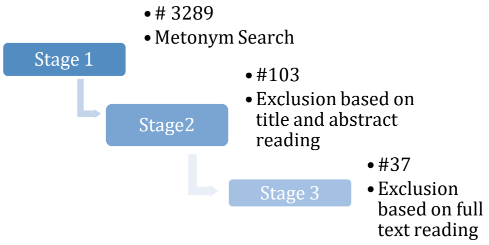

Stage 1
• # 3289
• Metonym Search
Stage2
• #103
• Exclusion based on title and abstract reading
Stage 3
• #37
• Exclusion based on full text reading

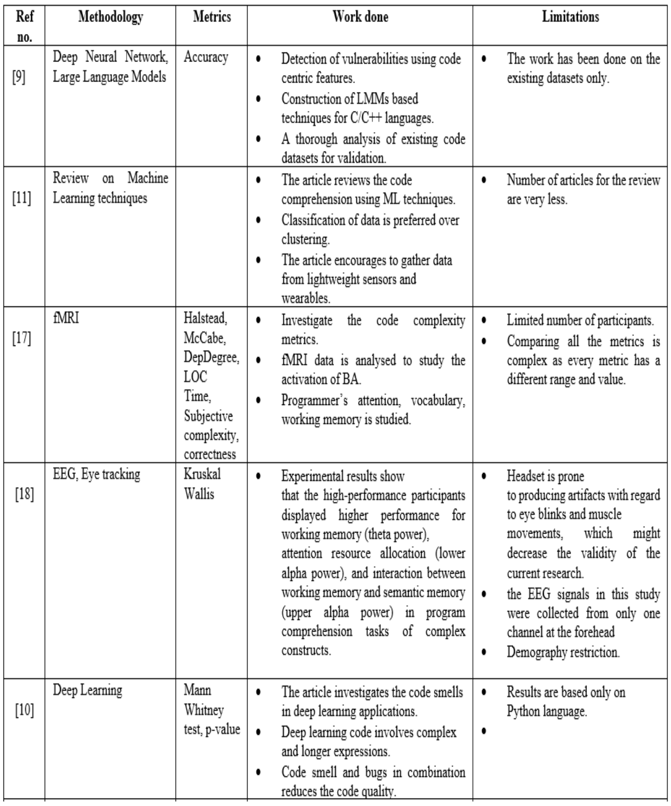
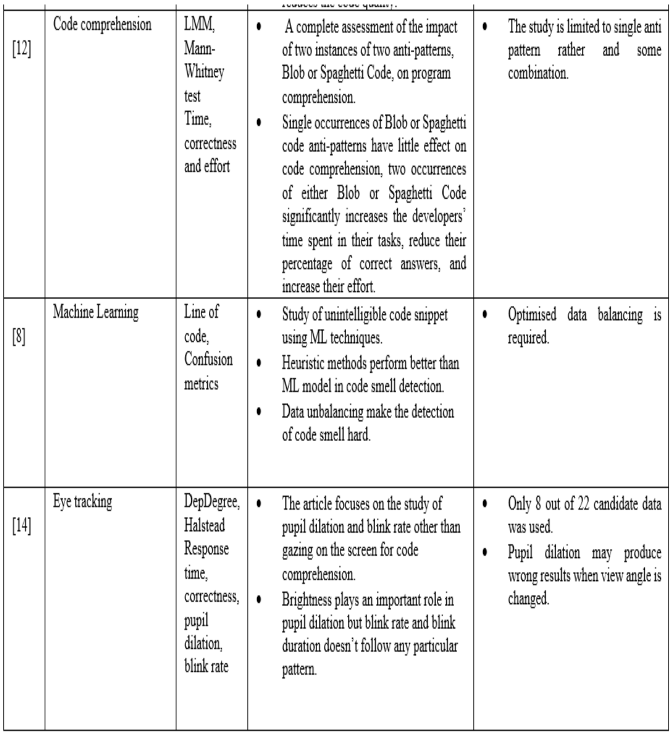
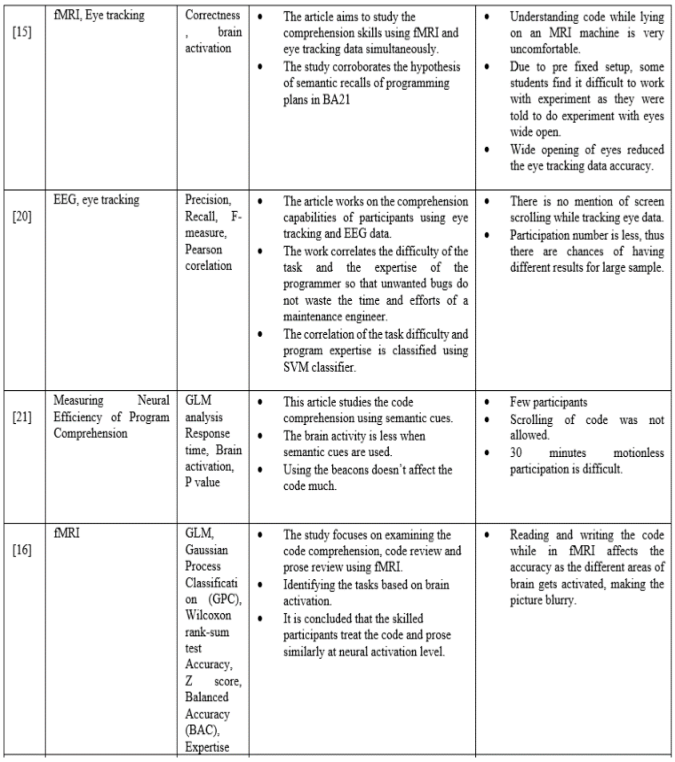
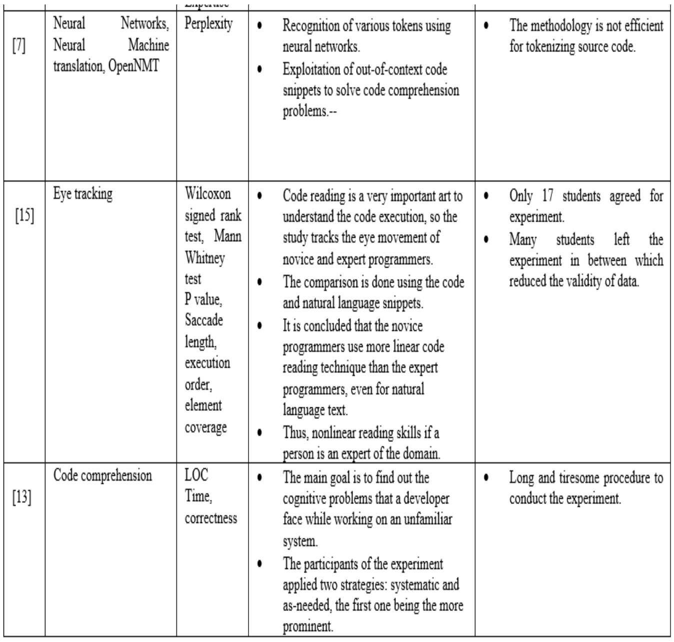
Fig 2. Inclusion-Exclusion of articles

In the first phase of our search, we obtained 3289 research publications by making use of the metonyms discussed above. On the basis of the titles and abstracts, we have chosen 103 of these papers to move on to the next phase. At stage three, 37 of those 103 papers are chosen for the literature review after being further refined by full-text reading. The exclusion-inclusion criteria are shown in fig 2.

3. LITERATURE SURVEY

Researchers have recently shown an interest in the intriguing issue of code comprehension. The current body of research is small, but encouraging. Scientists have utilized multiple approaches to decipher the enigma of the human mind behind deciphering a code, and their conclusions are conclusive. We have studied the articles written by several authors and have concluded what kind of research has been done so far in the paragraph below and through the fig 3.

Fig 3. Literature survey of program comprehension

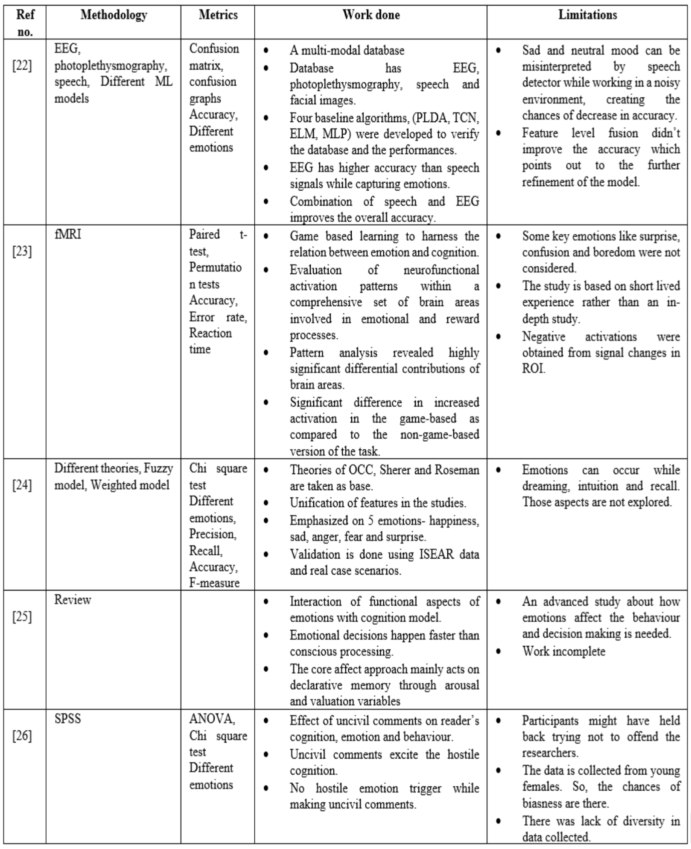
Ref no.	Methodology	Metrics	Work done	Limitations
[9]	Deep Neural Network, Large Language Models	Accuracy	• Detection of vulnerabilities using code centric features. • Construction of LMMs based techniques for C/C++ languages. • A thorough analysis of existing code datasets for validation.	• The work has been done on the existing datasets only.
[11]	Review on Machine Learning techniques		• The article reviews the code comprehension using ML techniques. • Classification of data is preferred over clustering. • The article encourages to gather data from lightweight sensors and wearables.	• Number of articles for the review are very less.
[17]	fMRI	Halstead, McCabe, DepDegree, LOC Time, Subjective complexity, correctness	• Investigate the code complexity metrics. • fMRI data is analysed to study the activation of BA. • Programmer's attention, vocabulary, working memory is studied.	• Limited number of participants. • Comparing all the metrics is complex as every metric has a different range and value.
[18]	EEG, Eye tracking	Kruskal Wallis	• Experimental results show that the high-performance participants displayed higher performance for working memory (theta power), attention resource allocation (lower alpha power), and interaction between working memory and semantic memory (upper alpha power) in program comprehension tasks of complex constructs.	• Headset is prone to producing artifacts with regard to eye blinks and muscle movements, which might decrease the validity of the current research. • the EEG signals in this study were collected from only one channel at the forehead • Demography restriction.
[10]	Deep Learning	Mann Whitney test, p-value	• The article investigates the code smells in deep learning applications. • Deep learning code involves complex and longer expressions. • Code smell and bugs in combination reduces the code quality.	• Results are based only on Python language. •
[12]	Code comprehension	LMM, Mann-Whitney test Time, correctness and effort	• A complete assessment of the impact of two instances of two anti-patterns, Blob or Spaghetti Code, on program comprehension. • Single occurrences of Blob or Spaghetti code anti-patterns have little effect on code comprehension, two occurrences of either Blob or Spaghetti Code significantly increases the developers' time spent in their tasks, reduce their percentage of correct answers, and increase their effort.	• The study is limited to single anti pattern rather and some combination.
[8]	Machine Learning	Line of code, Confusion metrics	• Study of unintelligible code snippet using ML techniques. • Heuristic methods perform better than ML model in code smell detection. • Data unbalancing make the detection of code smell hard.	• Optimised data balancing is required.
[14]	Eye tracking	DepDegree, Halstead Response time, correctness, pupil dilation, blink rate	• The article focuses on the study of pupil dilation and blink rate other than gazing on the screen for code comprehension. • Brightness plays an important role in pupil dilation but blink rate and blink duration doesn't follow any particular pattern.	• Only 8 out of 22 candidate data was used. • Pupil dilation may produce wrong results when view angle is changed.
[15]	fMRI, Eye tracking	Correctness , brain activation	• The article aims to study the comprehension skills using fMRI and eye tracking data simultaneously. • The study corroborates the hypothesis of semantic recalls of programming plans in BA21	• Understanding code while lying on an MRI machine is very uncomfortable. • Due to pre fixed setup, some students find it difficult to work with experiment as they were told to do experiment with eyes wide open. • Wide opening of eyes reduced the eye tracking data accuracy.
[20]	EEG, eye tracking	Precision, Recall, F-measure, Pearson corelation	• The article works on the comprehension capabilities of participants using eye tracking and EEG data. • The work correlates the difficulty of the task and the expertise of the programmer so that unwanted bugs do not waste the time and efforts of a maintenance engineer. • The correlation of the task difficulty and program expertise is classified using SVM classifier.	• There is no mention of screen scrolling while tracking eye data. • Participation number is less, thus there are chances of having different results for large sample.
[21]	Measuring Neural Efficiency of Program Comprehension	GLM analysis Response time, Brain activation, P value	• This article studies the code comprehension using semantic cues. • The brain activity is less when semantic cues are used. • Using the beacons doesn't affect the code much.	• Few participants • Scrolling of code was not allowed. • 30 minutes motionless participation is difficult.
[16]	fMRI	GLM, Gaussian Process Classification (GPC), Wilcoxon rank-sum test Accuracy, Z score, Balanced Accuracy (BAC), Expertise	• The study focuses on examining the code comprehension, code review and prose review using fMRI. • Identifying the tasks based on brain activation. • It is concluded that the skilled participants treat the code and prose similarly at neural activation level.	• Reading and writing the code while in fMRI affects the accuracy as the different areas of brain gets activated, making the picture blurry.
[7]	Neural Networks, Neural Machine translation, OpenNMT	Perplexity	• Recognition of various tokens using neural networks. • Exploitation of out-of-context code snippets to solve code comprehension problems.--	• The methodology is not efficient for tokenizing source code.
[15]	Eye tracking	Wilcoxon signed rank test, Mann Whitney test P value, Saccade length, execution order, element coverage	• Code reading is a very important art to understand the code execution, so the study tracks the eye movement of novice and expert programmers. • The comparison is done using the code and natural language snippets. • It is concluded that the novice programmers use more linear code reading technique than the expert programmers, even for natural language text. • Thus, nonlinear reading skills if a person is an expert of the domain.	• Only 17 students agreed for experiment. • Many students left the experiment in between which reduced the validity of data.
[13]	Code comprehension	LOC Time, correctness	• The main goal is to find out the cognitive problems that a developer face while working on an unfamiliar system. • The participants of the experiment applied two strategies: systematic and as-needed, the first one being the more prominent.	• Long and tiresome procedure to conduct the experiment.

Understanding the whole process of code comprehension is not simple as there are many aspects on which the whole enigma builds upon. Researchers need to have a deep understanding of all the concepts related to it, that is why literature survey is important. In the research work done so far by the different researchers, it is seen that they have tried various approaches and have worked on various points. Machine learning and deep learning are the currently used domains for most of the researchers. From the classic ML models to complex yet more precise neural network models have been used by researchers in [7,8,9,10,11]. Code comprehension requires various coding tasks and these tasks have been used in [12,13] to test the programming skills of the programmers. It is also important to observe where the programmers see the most when the code snippets are in front of them. The tracking of eye movement and the importance of gazing has provided some valuable results in [14,15]. Apart from the eye movement, brain activation is also an important performance parameter to test the cognitive skills of programmers. [16,17] have used fMRI data to look for the patterns of brain activation while performing the code comprehension experiments. To refine the results and provide more precise outcomes, multiple techniques have been used in an experimental work in [18,19,20,21].

3.1 Review of emotions in cognitive study

Emotions are mental states triggered by neurophysiological alterations; they are linked to different kinds of thinking, sensing, and behaving, as well as to degrees of pleasure and displeasure. The state of mind, or emotional state, is often intertwined with one's disposition, personality, or ability to think creatively. It is necessary to understand how far the work has been done in this field which is summed up in fig 4.

Fig 4. Review of emotions in cognitive study



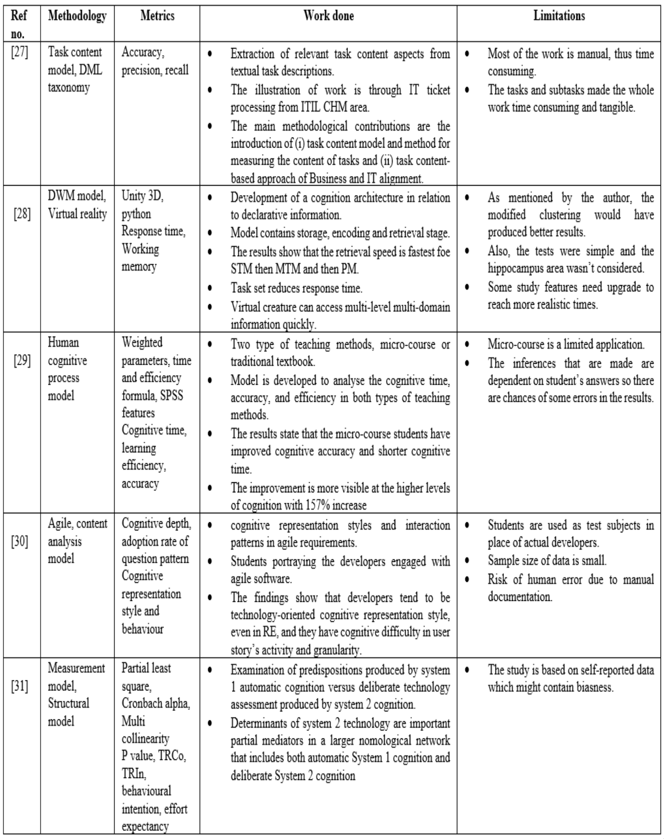
Ref no.	Methodology	Metrics	Work done	Limitations
[22]	EEG, photoplethysmography, speech, Different ML models	Confusion matrix, confusion graphs Accuracy, Different emotions	• A multi-modal database • Database has EEG, photoplethysmography, speech and facial images. • Four baseline algorithms, (PLDA, TCN, ELM, MLP) were developed to verify the database and the performances. • EEG has higher accuracy than speech signals while capturing emotions. • Combination of speech and EEG improves the overall accuracy.	• Sad and neutral mood can be misinterpreted by speech detector while working in a noisy environment, creating the chances of decrease in accuracy. • Feature level fusion didn't improve the accuracy which points out to the further refinement of the model.
[23]	fMRI	Paired t-test, Permutation tests Accuracy, Error rate, Reaction time	• Game based learning to harness the relation between emotion and cognition. • Evaluation of neurofunctional activation patterns within a comprehensive set of brain areas involved in emotional and reward processes. • Pattern analysis revealed highly significant differential contributions of brain areas. • Significant difference in increased activation in the game-based as compared to the non-game-based version of the task.	• Some key emotions like surprise, confusion and boredom were not considered. • The study is based on short lived experience rather than an in-depth study. • Negative activations were obtained from signal changes in ROI.
[24]	Different theories, Fuzzy model, Weighted model	Chi square test Different emotions, Precision, Recall, Accuracy, F-measure	• Theories of OCC, Sherer and Roseman are taken as base. • Unification of features in the studies. • Emphasized on 5 emotions- happiness, sad, anger, fear and surprise. • Validation is done using ISEAR data and real case scenarios.	• Emotions can occur while dreaming, intuition and recall. Those aspects are not explored.
[25]	Review		• Interaction of functional aspects of emotions with cognition model. • Emotional decisions happen faster than conscious processing. • The core affect approach mainly acts on declarative memory through arousal and valuation variables	• An advanced study about how emotions affect the behaviour and decision making is needed. • Work incomplete
[26]	SPSS	ANOVA, Chi square test Different emotions	• Effect of uncivil comments on reader's cognition, emotion and behaviour. • Uncivil comments excite the hostile cognition. • No hostile emotion trigger while making uncivil comments.	• Participants might have held back trying not to offend the researchers. • The data is collected from young females. So, the chances of biasness are there. • There was lack of diversity in data collected.

[22] created a multi-modal database model to research emotions using EEG, voice, photoplethysmography, and facial photographs. Four baseline algorithms—LDPA, TCN, ELM, MLP—are used to test performance. EEG signals captured emotions better. Game-based learning [23] examines emotions and cognition. Compared to non-game task, neurofunctional patterns implicated in emotional processes show considerable activation. [24] used OCC, Sherer, and Roseman ideas to create a weighted model for computing five primary emotions: happy, anger, sadness, fear, and surprise. ISEAR and real-world data confirm the work. Games let us quickly observe our emotions in different situations. Games demand cognitive skills for reaction time. [25] found that declarative memory influences core affect and emotional decisions happen faster than conscious processing. In [26], uncivil comments received online provoke hostile cognition, which affects behaviour and emotions, but when made by the individual, no hostile emotion trigger was seen. Whether we make emotional decisions purposefully or automatically is intriguing.

3.2 Review of psychology in cognitive study

The field of cognitive psychology investigates how the brain works in relation to cognitions such as education and communication. Different people have different learning styles, memory capacities, and modes of communication. Cognitive psychology is the study of how people act based on their perceptions and interpretations of information. Psychologists, like the ancient philosophers Aristotle and Plato, are interested in the acquisition of knowledge, though their methods are significantly more sophisticated. Literature survey for cognitive psychology is shown in fig 5.

Fig 5. Review of psychology in cognitive study



Ref no.	Methodology	Metrics	Work done	Limitations
[27]	Task content model, DML taxonomy	Accuracy, precision, recall	• Extraction of relevant task content aspects from textual task descriptions. • The illustration of work is through IT ticket processing from ITIL CHM area. • The main methodological contributions are the introduction of (i) task content model and method for measuring the content of tasks and (ii) task content-based approach of Business and IT alignment.	• Most of the work is manual, thus time consuming. • The tasks and subtasks made the whole work time consuming and tangible.
[28]	DWM model, Virtual reality	Unity 3D, python Response time, Working memory	• Development of a cognition architecture in relation to declarative information. • Model contains storage, encoding and retrieval stage. • The results show that the retrieval speed is fastest foe STM then MTM and then PM. • Task set reduces response time. • Virtual creature can access multi-level multi-domain information quickly.	• As mentioned by the author, the modified clustering would have produced better results. • Also, the tests were simple and the hippocampus area wasn't considered. • Some study features need upgrade to reach more realistic times.
[29]	Human cognitive process model	Weighted parameters, time and efficiency formula, SPSS features Cognitive time, learning efficiency, accuracy	• Two type of teaching methods, micro-course or traditional textbook. • Model is developed to analyse the cognitive time, accuracy, and efficiency in both types of teaching methods. • The results state that the micro-course students have improved cognitive accuracy and shorter cognitive time. • The improvement is more visible at the higher levels of cognition with 157% increase	• Micro-course is a limited application. • The inferences that are made are dependent on student's answers so there are chances of some errors in the results.
[30]	Agile, content analysis model	Cognitive depth, adoption rate of question pattern Cognitive representation style and behaviour	• cognitive representation styles and interaction patterns in agile requirements. • Students portraying the developers engaged with agile software. • The findings show that developers tend to be technology-oriented cognitive representation style, even in RE, and they have cognitive difficulty in user story's activity and granularity.	• Students are used as test subjects in place of actual developers. • Sample size of data is small. • Risk of human error due to manual documentation.
[31]	Measurement model, Structural model	Partial least square, Cronbach alpha, Multi collinearity P value, TRCo, TRIn, behavioural intention, effort expectancy	• Examination of predispositions produced by system 1 automatic cognition versus deliberate technology assessment produced by system 2 cognition. • Determinants of system 2 technology are important partial mediators in a larger nomological network that includes both automatic System 1 cognition and deliberate System 2 cognition	• The study is based on self-reported data which might contain biasness.

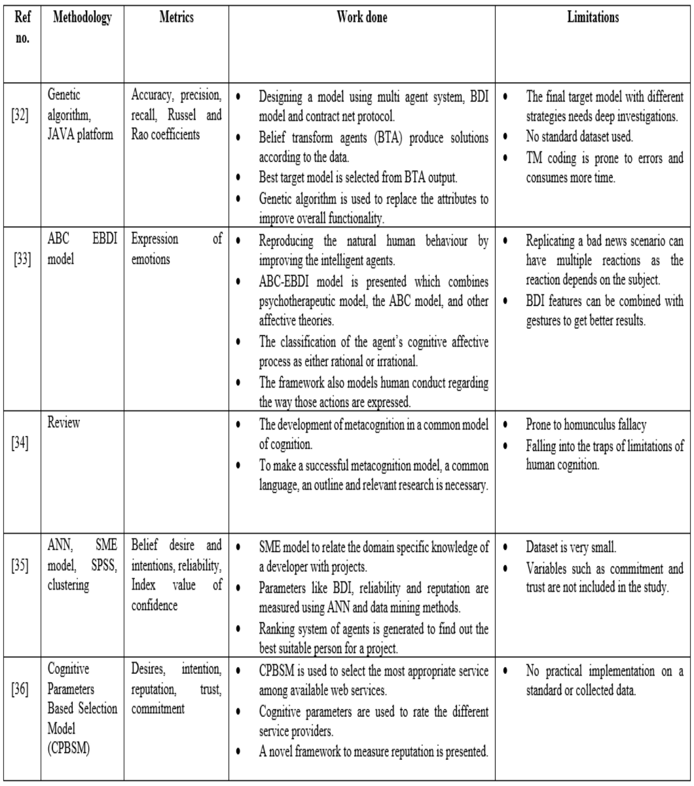
[27] emphasizes task descriptions and interactive decision help. [28] creates a declarative cognitive architecture. The model has storage, retrieval, and encoding steps. Task set reduces response time, and STM retrieves faster than MTM and PM. [29] examines the teaching area to understand collegiate cognitive psychology. The two sorts of teaching approaches are textbook and micro-course. The major goal is to observe cognitive impact using both ways because it influences time, accuracy, and efficiency. Psychology depends on the information that our brain has. The psychology of developers is imitated in [30] as the students are given the role of developers. The agile software records the cognition style

and the interaction patterns. The methods discussed in [31] examines the predispositions produced from automatic cognitive system versus the deliberate technology assessment system. The study observes the behavioural intentions and the effort expectancy of the participants.

3.3 Review of BDI in cognitive study

The BDI model comprises several mental qualities and relationships. Bratman proposes the BDI model, a philosophical framework of practical reasoning. Beliefs, desires, and intentions make up the mental attitude components of the BDI paradigm. Beliefs are knowledge or information about the world. Intentions and actions are processed in order, and the environment is updated. So, to understand the work done in this area, the literature survey is done which is presented in Fig 6.

Fig 6. Review of BDI in cognitive study



Ref no.	Methodology	Metrics	Work done	Limitations
[32]	Genetic algorithm, JAVA platform	Accuracy, precision, recall, Russel and Rao coefficients	• Designing a model using multi agent system, BDI model and contract net protocol. • Belief transform agents (BTA) produce solutions according to the data. • Best target model is selected from BTA output. • Genetic algorithm is used to replace the attributes to improve overall functionality.	• The final target model with different strategies needs deep investigations. • No standard dataset used. • TM coding is prone to errors and consumes more time.
[33]	ABC EBDI model	Expression of emotions	• Reproducing the natural human behaviour by improving the intelligent agents. • ABC-EBDI model is presented which combines psychotherapeutic model, the ABC model, and other affective theories. • The classification of the agent's cognitive affective process as either rational or irrational. • The framework also models human conduct regarding the way those actions are expressed.	• Replicating a bad news scenario can have multiple reactions as the reaction depends on the subject. • BDI features can be combined with gestures to get better results.
[34]	Review		• The development of metacognition in a common model of cognition. • To make a successful metacognition model, a common language, an outline and relevant research is necessary.	• Prone to homunculus fallacy • Falling into the traps of limitations of human cognition.
[35]	ANN, SME model, SPSS, clustering	Belief desire and intentions, reliability, Index value of confidence	• SME model to relate the domain specific knowledge of a developer with projects. • Parameters like BDI, reliability and reputation are measured using ANN and data mining methods. • Ranking system of agents is generated to find out the best suitable person for a project.	• Dataset is very small. • Variables such as commitment and trust are not included in the study.
[36]	Cognitive Parameters Based Selection Model (CPBSM)	Desires, intention, reputation, trust, commitment	• CPBSM is used to select the most appropriate service among available web services. • Cognitive parameters are used to rate the different service providers. • A novel framework to measure reputation is presented.	• No practical implementation on a standard or collected data.

Understanding BDI is crucial to understanding source code. BTA model [32] uses BDI to rank models and select the best. Genetic algorithm improves model correctness. Human behaviour impacts beliefs, wants, and intentions. In [33], the intelligent agents that support those are upgraded to present an ABC-EBDI paradigm. It incorporates the psychotherapy paradigm, the ABC model, and other affective theories. The framework models human conduct in terms of acts and expression. In a typical model of cognition, [34] focuses on the metacognition model. [35] uses a SME model to evaluate developers by BDI, reliability, and reputation and choose the best project. ANN and clustering are utilized to validate the work. Similar to the previous work, [36] uses BDI to rank online services. It also introduces a novel framework to use reputation as a measuring parameter. The literature survey above shows that there is no standardized dataset for these investigations.

4. OBSERVATIONS

4.1 General methods for testing cognitive skills and brain activation

The literature survey of this study has been divided into four parts, code comprehension, emotions, cognitive psychology and BDI. The reason for such division is the diversification of methods and models that has been used in these studies. The methods that have been used for code comprehension have been shown in fig 7. There is a clear preference for evaluating programmers' analytical abilities through coding exercises (31%). Only twenty-four percent of researchers have come up with their own models to analyse human cognition. Other common approaches to gauging mental acuity include electroencephalography (6%), functional magnetic resonance imaging (9%), eye tracking (6%), and a combination of modalities (12%). Methods like as electroencephalography (EEG), eye-tracking, functional magnetic resonance imaging (fMRI), etc., are all part of the several techniques used in a single experiment.

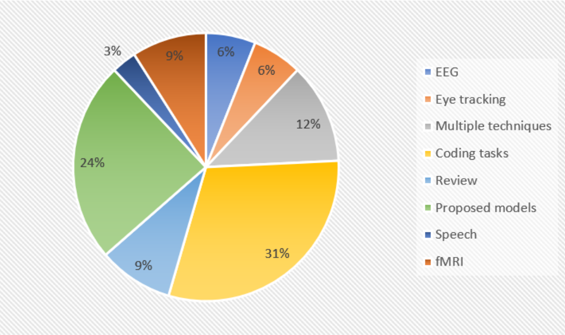


Fig 7. General methods

4.2 Which performance metrics have been used for the research works?

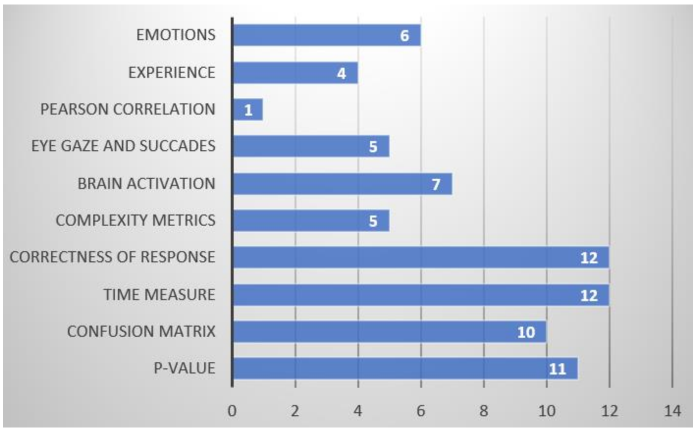


Fig 8. Performance metrics

Performance metrics are examined to gauge a work's effectiveness. The effectiveness of the findings is indicated by the performance metrics. The research works depicted in fig. 8 make use of a variety of performance criteria. The p-value, confusion matrix, time measure, correctness of response, complexity metrics, brain activation, eye gaze and succades, pearson correlation, experience, and emotions are just few of the metrics that have been employed. The numbers show how many papers used each metric.

4.3 Various computational methods in all the approaches

Researchers conduct experiments to draw meaningful conclusions. In order to offer meaningful research, the data gathered from the various approaches must be processed using certain computational models. Some various computational methods have been used in the reviewed articles which are shown below in the Fig 9.

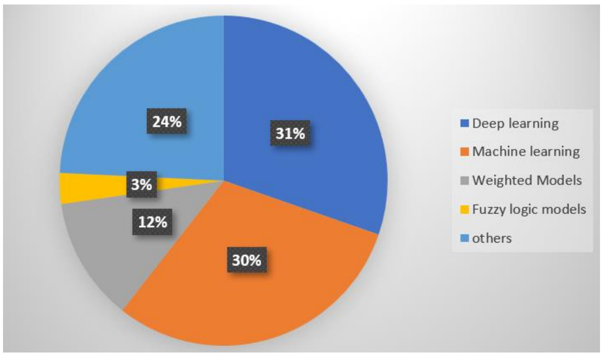


Fig 9. Computational methods

Machine learning has been used in 30% of the articles to extract various features related to data, to do component analysis and for classification. Deep learning (31%) is used to extract the features from the images. It is not necessary that the machine learning and deep learning techniques are only used to extract features or to classify the level of experience of programmers. The surveyed articles have used these techniques for classification and extraction of features. 24% of the reviewed have either used some other computational approach which haven't been mentioned in the paper or there is no computational approach involved in those. 12% have used weighted models and 3% have used fuzzy logic as computational methods.

5. CONCLUSION

An exhaustive literature search was conducted for this article. The surveyed publications cover cognitive abilities like emotion, psychology, and BDI as well as code comprehension. While analysing a code snippet, performance metrics like time and accuracy are crucial. Computational methods such as Deep Learning and Machine Learning are more popular thanks to intelligent computing. Constraints on the number of people who can participate, the size of the code they can work with, and the freedom they have to move around are some of the most prevalent difficulties encountered in experimental work. Methods including functional magnetic resonance imaging, electroencephalography, and eye tracking have emerged as the primary tools for academic research. Finding efficient methods to analyse brain activity with a sufficient sample size can be the focus of future research in this area. Code complexity and speech recognition are two further software and performance metrics that can be used as criteria for evaluating a programmer's cognitive abilities.

REFERENCES

[1] A. Kantahong, "What are Cognitive Skills and Why are They Important? – DEEPGAMMA," pp. 1–6, 2020, [Online]. Available: https://gammaiq.org/2020/07/24/what-are-cognitive-skills-and-why-are-they-important/

[2] G. Parikh, "What is software maintenance really?," *ACM SIGSOFT Softw. Eng. Notes*, vol. 9, no. 2, pp. 114–116, 1984, doi: 10.1145/1012467.1012475.

[3] M. Request, P. Report, S. C. Management, S. Agreement, and S. Q. Assurance, "Chapter 5: Software Maintenance - SWEBOK," pp. 1–13, [Online]. Available: http://swebokwiki.org/Chapter_5:_Software_Maintenance

[4] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," *12th Int. Conf. Eval. Assess. Softw. Eng. EASE 2008*, no. February 2015, 2008, doi: 10.14236/ewic/ease2008.8.

[5] T. Kosar, S. Bohra, and M. Mernik, "A Systematic Mapping Study driven by the margin of error," *J. Syst. Softw.*, vol. 144, no. June, pp. 439–449, 2018, doi: 10.1016/j.jss.2018.06.078.

[6] B. Kitchenham, "Guidelines for performing Systematic Literature Reviews in Software Engineering (Software Engineering Group, Department of Computer Science, Keele …," no. January, 2007.

[7] C. V. Alexandru, S. Panichella, and H. C. Gall, "Replicating Parser Behavior Using Neural Machine Translation," *IEEE Int. Conf. Progr. Compr.*, pp. 316–319, 2017, doi: 10.1109/ICPC.2017.11.

[8] F. Pecorelli, F. Palomba, D. Di Nucci, and A. De Lucia, "Comparing heuristic and machine learning approaches for metric-based code smell detection," *IEEE Int. Conf. Progr. Compr.*, vol. 2019-May, pp. 93–104, 2019, doi: 10.1109/ICPC.2019.00023.

[9] N. Gervasoni, *A Code Centric Evaluation of C / C ++ Vulnerability Datasets for Deep Learning Based Vulnerability Detection Techniques*, vol. 1, no. 1. Association for Computing Machinery, 1999. doi: 10.1145/3578527.3578530.

[10] H. Jebnoun, H. Ben Braiek, M. M. Rahman, and F. Khomh, "The Scent of Deep Learning Code: An Empirical Study," *Proc. - 2020 IEEE/ACM 17th Int. Conf. Min. Softw. Repos. MSR 2020*, pp. 420–430, 2020, doi: 10.1145/3379597.3387479.

[11] M. Castelo-branco, I. Cibit, and R. Couceiro, "A Quick Review on Machine Learning Techniques in Code Comprehension and Code Review Estimated by Neurophysiological Data," pp. 408–413, 2022.

[12] C. Politowski *et al.*, "A large scale empirical study of the impact of Spaghetti Code and Blob anti-patterns on program comprehension," *Inf. Softw. Technol.*, vol. 122, no. January, 2020, doi: 10.1016/j.infsof.2020.106278.

[13] A. Karahasanović, A. K. Levine, and R. Thomas, "Comprehension strategies and difficulties in maintaining object-oriented systems: An explorative study," *J. Syst. Softw.*, vol. 80, no. 9, pp. 1541–1559, 2007, doi: 10.1016/j.jss.2006.10.041.

[14] N. Peitek, J. Siegmund, C. Parnin, S. Apel, and A. Brechmann, "Beyond gaze: Preliminary analysis of pupil dilation and blink rates in an fMRI study of program comprehension," *Proc. - EMIP 2018 Eye Movements Program.*, 2018, doi: 10.1145/3216723.3216726.

[15] T. Busjahn *et al.*, "Eye Movements in Code Reading: Relaxing the Linear Order," *IEEE Int. Conf. Progr. Compr.*, vol. 2015-Augus, pp. 255–265, 2015, doi: 10.1109/ICPC.2015.36.

[16] B. Floyd, T. Santander, and W. Weimer, "Decoding the Representation of Code in the Brain: An fMRI Study of Code Review and Expertise," *Proc. - 2017 IEEE/ACM 39th Int. Conf. Softw. Eng. ICSE 2017*, pp. 175–186, 2017, doi:

10.1109/ICSE.2017.24.

[17] N. Peitek, S. Apel, C. Parnin, A. Brechmann, and J. Siegmund, “Program Comprehension and Code Complexity Metrics: A Replication Package of an fMRI Study,” *Proc. - Int. Conf. Softw. Eng.*, pp. 168–169, 2021, doi: 10.1109/ICSE-Companion52605.2021.00071.

[18] Y. T. Lin, Y. Z. Liao, X. Hu, and C. C. Wu, “EEG Activities during Program Comprehension: An Exploration of Cognition,” *IEEE Access*, vol. 9, pp. 120407–120421, 2021, doi: 10.1109/ACCESS.2021.3107795.

[19] N. Peitek, J. Siegmund, C. Parnin, S. Apel, J. C. Hofmeister, and A. Brechmann, “Simultaneous measurement of program comprehension with fMRI and eye tracking: A case study,” *Int. Symp. Empir. Softw. Eng. Meas.*, 2018, doi: 10.1145/3239235.3240495.

[20] S. Lee, D. Hooshyar, H. Ji, K. Nam, and H. Lim, “Mining biometric data to predict programmer expertise and task difficulty,” *Cluster Comput.*, vol. 21, no. 1, pp. 1097–1107, 2018, doi: 10.1007/s10586-017-0746-2.

[21] J. Siegmund *et al.*, “Measuring neural efficiency of program comprehension,” pp. 140–150, 2017, doi: 10.1145/3106237.3106268.

[22] Q. Wang, M. Wang, Y. Yang, and X. Zhang, “Multi-modal emotion recognition using EEG and speech signals,” *Comput. Biol. Med.*, vol. 149, p. 105907, Oct. 2022, doi: 10.1016/J.COMPBIOMED.2022.105907.

[23] S. Greipl *et al.*, “When the brain comes into play: Neurofunctional correlates of emotions and reward in game-based learning,” *Comput. Human Behav.*, vol. 125, no. January, p. 106946, 2021, doi: 10.1016/j.chb.2021.106946.

[24] S. Jain and K. Asawa, “Modeling of emotion elicitation conditions for a cognitive-emotive architecture,” *Cogn. Syst. Res.*, vol. 55, pp. 60–76, 2019, doi: 10.1016/j.cogsys.2018.12.012.

[25] O. Larue *et al.*, “Emotion in the Common Model of Cognition,” *Procedia Comput. Sci.*, vol. 145, pp. 740–746, 2018, doi: 10.1016/j.procs.2018.11.045.

[26] L. Rösner, S. Winter, and N. C. Krämer, “Dangerous minds? Effects of uncivil online comments on aggressive cognitions, emotions, and behavior,” *Comput. Human Behav.*, vol. 58, pp. 461–470, 2016, doi: 10.1016/j.chb.2016.01.022.

[27] N. Rizun, A. Revina, and V. G. Meister, “Analyzing content of tasks in Business Process Management. Blending task execution and organization perspectives,” *Comput. Ind.*, vol. 130, p. 103463, 2021, doi: 10.1016/j.compind.2021.103463.

[28] L. Martin, K. Jaime, F. Ramos, and F. Robles, “Declarative working memory: A bio-inspired cognitive architecture proposal,” *Cogn. Syst. Res.*, vol. 66, pp. 30–45, 2021, doi: 10.1016/j.cogsys.2020.10.014.

[29] M. Lv, H. Liu, W. Zhou, and C. Zheng, “Efficiency model of micro-course study based on cognitive psychology in the college,” *Comput. Human Behav.*, vol. 107, no. April 2019, p. 106027, 2020, doi: 10.1016/j.chb.2019.05.024.

[30] J. Jia, X. Yang, R. Zhang, and X. Liu, “Understanding software developers’ cognition in agile requirements engineering,” *Sci. Comput. Program.*, vol. 178, pp. 1–19, 2019, doi: 10.1016/j.scico.2019.03.005.

[31] V. Khatri, B. M. Samuel, and A. R. Dennis, “System 1 and System 2 cognition in the decision to adopt and use a new technology,” *Inf. Manag.*, vol. 55, no. 6, pp. 709–724, 2018, doi: 10.1016/j.im.2018.03.002.

[32] A. Siabdelhadi, A. Chadli, H. Cherroun, and A. Ouared, “MoTrans-BDI : Leveraging the Beliefs-Desires-Intentions agent architecture for collaborative model transformation by example,” *J. Comput. Lang.*, vol. 74, no. September 2022, p. 101174, 2023, doi: 10.1016/j.cola.2022.101174.

[33] Y. Sánchez, T. Coma, A. Aguelo, and E. Cerezo, “ABC-EBDI: An affective framework for BDI agents,” *Cogn. Syst. Res.*, vol. 58, pp. 195–216, 2019, doi: 10.1016/j.cogsys.2019.07.002.

[34] J. D. Kralik *et al.*, “Metacognition for a Common Model of Cognition,” *Procedia Comput. Sci.*, vol. 145, pp. 730–739, 2018, doi: 10.1016/j.procs.2018.11.046.

[35] A. Mishra and V. Srivastava, “Cognition based selection and categorization of maintenance engineer (agent) using Artificial Neural Net and Data Mining methods,” *2012 CSI 6th Int. Conf. Softw. Eng. CONSEG 2012*, 2012, doi: 10.1109/CONSEG.2012.6349509.

[36] S. Kumar and R. B. Mishra, “Cognition based Service Selection in Semantic Web Service Composition,” *INFOCOMP J. Comput. Sci.*, vol. 7, no. 3 SE-Articles, pp. 35–41, Sep. 2008, [Online]. Available: https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/227

APPENDIX

Acronyms

SDLC- Software Developement Life Cycle

BDI- Belief-Desire-Intention

fMRI- Functional Magnetic Resonance Imaging

EEG- Electroencephalogram

LOC- Lines of Code

LMM- Linear Mixed Model

ML- Machine Learning

BA- Brodmann Area

GLM- General Linear Model

ANOVA-Analysis of Variance

PLDA- Probabilistic Linear Discriminant Analysis

TCN- Temporal Convolutional Network

ELM- Extreme Learning Machine

MLP- Multi-Layer Perception

STM- Short Term Memory

MTM- Medium Term Memory

PM- Permanent Memory

ANN- Artificial Neural Network

Disease Diagnosis and Prediction using DNA sequencing with patterns in Biological Data

Banothu Ramji
Computer Science and Engineering
National Institute of Technology
Warangal, Telangana.
br22csr2p01@student.nitw.ac.in

Raju Bhukya
Computer Science and Engineering
National Institute of Technology
Warangal, Telangana, India-506004
raju@nitw.ac.in

Abstract: *The basic goal of string matching research is to determine whether specific patterns are present in a sample of text that is equal in size or larger. Pattern matching is required as part of the pattern discovery process in today's culture to detect structural and functional behavior in genes. Pattern matching is common in information systems and data processing, but it's also important in everyday life. String matching algorithms have been used in a variety of applications, including DNA analysis. DNA sequencing is one of the most important disciplines of science today [6]. One of these is disease diagnosis using DNA sequencing, which a safe method of evaluating the chance of a disease is occurring. Multiple genetic mutations are to blame for the disease. Due to the fact that DNA is such a big database, so a good algorithm is needed to check the disease condition. This research suggests the sequential and parallel approaches of the multi pattern matching algorithm using Bloom Filters and Aho-corasick Algorithm corresponding to the Aho-corasick method to determine the likelihood of nucleotide-repeat diseases arising from various DNA sequences. The result analysis shows the comparison and efficiency of the algorithms.*

Keywords: *Pattern matching, Bloom Filters, DNA sequencing, Nucleotide repeat diseases, Aho-corasick Algorithm, Result analysis*

1. Introduction

One of the most crucial fields of research right now is DNA sequencing. Several fields, including forensic science, contemporary agriculture, and contemporary medicine, use DNA sequencing. Medical science is developing day by day to detect diseases and as well as early detection or prediction of diseases such as cancer. Change in sequence of DNA leads to diseases. DNA sequences make large databases hence efficient and fast algorithm is required for DNA sequencing. All species, including humans, have the genetic material known as DNA (Deoxyribonucleic Acid). The DNA of every cell in the body is identical. Adenine (A), guanine (G), cytosine (C), and thymine make up the DNA (T). Base + Sugar + Phosphate = Nucleotide, where each pair of the chemical bases is joined by a sugar and a phosphate molecule. DNA molecules form together thread-like structures called chromosome which are present in nucleus of each cell in the body. The method of DNA sequencing involves figuring out the exact placement of nucleotides within DNA molecules. The letters A, G, C, and T are arranged in various sequences. Disease shows a certain pattern of DNA which can be detected by pattern matching. The amount of DNA extracted from The growth of the organism is exponential. Thus, pattern matching algorithms are crucial in many computational biology applications for data processing involving proteins and genes. Its main goal is to identify a specific pattern within a given DNA sequence. Examination the amount of DNA extracted from the organism is increasing exponentially. In order to match patterns created using Pattern Matching Expressions, pattern matching approaches read the inputs through text strings. Pattern Matching can be used in Identification DNA sequence.

A. Deoxyribonucleic acid (DNA)

Deoxyribonucleic acid, also known as DNA, is a large molecule that houses each organism's individual genetic code. It contains the directions for generating every protein in our bodies, much like a recipe book. The four fundamental "bases" that make up DNA are adenine (A), cytosine (C), guanine (G), and thymine (T) (T). The genome's instructions are created by the order, or sequence, of these nucleotides. The molecule of DNA has two strands. DNA has a distinctive "double helix" form that resembles a twisted ladder. The base sequence is replicated using a template found on each double helix strand of DNA. Cells splits reproducing off springs, this is crucial because each new cell requires an identical duplicate of the old cell's DNA. DNA is responsible for conducting the information required to build and function particular organism. Certain parts of DNA are in charge of turning genes on and off, as well as controlling the amount of that particular protein is produced.

B. DNA Sequencing

DNA sequencing indicates the arrangement in which letters are combined in DNA. Individual genes, entire DNA, complete chromosomes, and entire genomes are sequenced by way of DNA sequencing. To sequence DNA, it must go through a prolonged scientific computing procedure. DNA computation is complicated process involving a lot of precision and complex logic. The state of DNA computation is still being worked out. DNA calculations have not yet surpassed the computing capabilities of current computers. We acquire sequenced DNA in a solution after the final phase of detection and reading, and then the order of base pairs that make up the strand of DNA is identified. It focuses on finding the particular pattern in a given DNA sequence.

C. Disease Diagnosis using DNA Sequencing

Disease diagnosis by DNA sequencing is a challenging medical subject to master. Diseases arecaused by repeating genes; repetitive genes, deleted genes, and the precise presence of disease-affected genes are among other factors. DNA sequencing is a first-stage diagnostic for disease detection that is perfectly safe.

DNA databases are massive and growing day after day. As a result, analyzing data sets is not a manual process. Exact and inexact model matching algorithms, as well as single and multi faceted model matching algorithms, are the most common categories. Model correspondence techniques' main task is to identify known models from massive data sets. Some inaccuracies may be regarded as per application need in the inexact pattern matching process but exact pattern matching has no errors recognized in the outcome. When there are several match patterns to scan for, a single pattern matching algorithm executes more than one string at a time. Users will find it more difficult to get critical information from sequences as the bulk of the data expands. As a result, faster pattern matching algorithms require more efficient and resilient procedures. It It is one of the most significant areas in bio informatics that has been researched. The method is designed to get better as the data set expands. Because bio informatics requires precise findings, accurate and multi-string pattern matching techniques are utilized. With multi-string patterns to be searched, more than one string is passed in a single pass by a single pattern matching algorithm, which passes one string at a time.

2. Literature Review

The Agrep Algorithm for Approximate Nucleotide Sequence Matching was proposed by Hongjian Li et al. in [1]. Currently used pattern matching algorithms can only match a limited number of genomic sequences. The Agrep method's CUDA implementation provides a rough sequence matching algorithm that searches huge genomes for patterns of length up to 64 and edit distance up to 9. Its memory demand was decreased by using two bits for each binary representation of a single character. Therefore it is possible to load multiple genomes simultaneously. Its high sensitivity is guaranteed by strictly scanning through the entire genome. High sensitivity, which gauges the percentage of actual positives that are accurately identified, is another benefit of CUDA Agrep. It is determined as the proportion of mapped reads to all reads. CUDA Agrep is assured not to overlook any potential approximation matching because the reference genome is thoroughly examined for each query without the use of any heuristic methods. Also, an email crawler and an AJAX MVC website for real-time internet searching were constructed.

A novel accurate multiple pattern matching approach using DNA sequence and pattern pair was introduced by Raju Bukya, DVLN Somayajulu in [2]. There is no need to build the indexes for the input sequence again once they have been created. The algorithm for each pattern starts with the character that matches the pattern, reducing the need for pointless comparisons of other characters. It performs well for applications using DNA sequences. In certain of the scenarios of the suggested algorithm, the number of comparisons lowers as the size of the pattern increases. In terms of the quantity of comparisons and CPC (Comparations per Character) ratio, the algorithm suggested in this study performs better than some of the other widely used algorithms. In compared to algorithms like MSMPMA, Brute-force, Tri Match, Nave string matching, and IKPMPM methods, the suggested approach performs well in two metrics, CPC ratio and number of comparisons. EPMSPP approach gives very good performance related tothe other older methods.

In [4] Boyer-Moore Algorithm, when a mismatch is identified, it conducts larger shift increment. In terms of scanning, it differs significantly from Naive. It traverses the string going right to left, unlike Naive, in which the last character of P is matched with the very first character of T. If a character matches, the pointer is advanced to left to the pattern's remaining elements. If a conflict is found in T at character c that does not exist in P, P is advanced right by m places and adjusted to a next character following c. P is moved right such that c is aligned with rightmost appearance of c in P if c is part of P. But the worst time complexity still remains O (m+n).

In [5] Knuth-Morris-Pratt, for the specified pattern P, Finite State Automata (FSM) model M is first generated.If there exist pattern in the text, it is approved, else it's ignored. The KMP algorithm initially generates a supplementary LPS of size m (the same as the pattern size) that is utilized to skip characters during matching. LPS represents the Longest Prefix Suffix. It also servesas a suffix. KMP has the advantage of already knowing some of the next characters in the following window's text. Knowing this, we can avoid matching characters that we already know will match. The sole drawback of the KMP method is that doesn't indicate the count of occurrence of the pattern has occurred. Dynamic programming is one among the highly utilized algorithm in computer science. It entailed recursively addressing consecutive recurrence relations, in which smaller issues are addressed in order to answer the larger problem. The worst time complexity is O(n).

S. Hasib, M. Motwani, and A. Saxena discussed the importance of the Aho-Corasick String Matching Algorithm in Real World Applications in [7]. The Aho-Corasick algorithm has been found to be particularly effective for multiple pattern matching and to be applicable to a wide range of problems. However, it has been found that the algorithm's performance suffers both in terms of time and space as the size of the automata increases significantly. The algorithm's complexity increases linearly with the length of the patterns, the length of the text being searched, and the number of output matches. Due to the ability to match all keywords simultaneously in a single pass, it is found to be appealing for huge amounts of keywords. This technique can be used to address a number of issues, including intrusion detection, plagiarism detection, bioinformatics, digital forensics, and text mining, among others. The technique of intrusion detection uses an intrusion detection system to find intrusions (IDS).

In [3] Burton Howard Bloom created the space-efficient probabilistic data structure known as the Bloom filter in 1970. It is used to determine whether an element is a part of a set. False 8 positive matches are possible, but false negative matches are not; in other words, a query returns either potentially in the set or unquestionably out of the set. Although this can be handled with the counting Bloom filter variation, elements can only be added to the set; the more things added, the higher the risk of false positives. Bloom filters, compared to other data structures for expressing sets, such as self-balancing binary search trees, attempts, hash tables, or straightforward arrays or linked lists of the entries, have a significant space advantage despite the possibility of false positives. For the most part, this necessitates keeping at least the data items themselves, which can range in size from a few bits for small integers to a large number of bits for strings (tries are an exception since they can share storage between elements with equal prefixes). The actual storage must be handled separately because Bloom filters do not store the data pieces in any way. Nevertheless, because Bloom filters don't actually store the data pieces, a different solution must be offered for their actual storage.

Sandeep U.Mane and Ketaki H. Pangu suggested a disease diagnosis method based on DNA sequencing in [6]. A sequential approach based on the GPGPU. In order to determine the likelihood of nucleotide repeat diseases and some cancer types from DNA sequencing in parallel approach using GPGPU and compare results between sequential and parallel approach, the paper presents design and implementation of the Aho-corasick exact and multi string pattern matching algorithm. This article discusses the GPGPU-based Aho-corasick multi-string pattern matching algorithm, which produces better results in a shorter amount of time. By contrasting the two methods, the authors come to the conclusion that the GPGPU-based parallel algorithm performs better than the sequential technique when the number of patterns rises and as well. The fastest speedup possible was about two times faster than the sequential version.

3. Problem Statement

As mentioned earlier DNA is such a big database and growing genetic databases so a good and faster pattern matching algorithm is needed to check the disease condition for diagnosing nucleotide repeat diseases. According to study in [9], a parallel optimized Aho-corasick precise and multi-string pattern matching algorithm with bloom filters should be created and put into use to estimate the likelihood of nucleotide repeat disorders using DNA sequencing. Also propose implementation utilizing sequential and parallelapproach. And analyze and compare the results obtained from Parallel failure less aho-corasick algorithm using bloom filters and aho-corasick. Collective data setanalysis to extract inferences from overall results.

4. Proposed Solution

This research proposes a modification of the Aho-corasick [7] Pattern Matching Algorithm for calculating the number of occurrences of specific patterns in a DNA sequence. It consists of two parts: the generation of finite-state automata from input keywords, and the processing of the DNA sequence in a single pass utilizing these automata. The algorithm's main goal is to find and identify all substrings of Input patterns that are keywords of Input sequence.

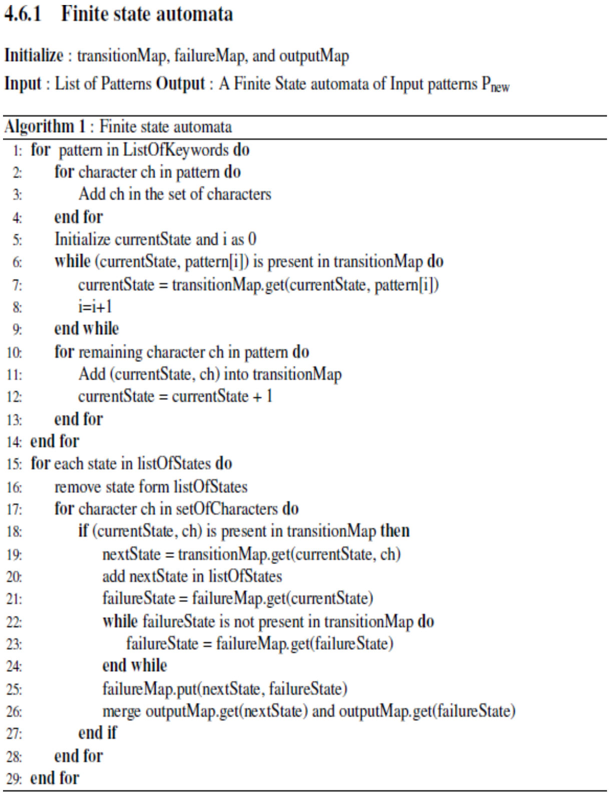
Improvements: Minimizing Number of times accessing the finite state automata by using inexact pattern. Matching first before accessing the Finite State automata. . Minimizing the number of computations by making chunks of the given input and performing aho-corasick algorithm on the chunk separately by assigning athread.

Chunking Input data in order to make it multithreaded: Before to parallel processing, the DNA sequence file is split into many blocks. As a result, the file's data will be split into many data chunks. Each block is given a thread, which is then assigned to it according to its availability. After that, each block is sent to automation to look for patterns.[8] In Parallel block aho-corasick algorithm we divide the input sequence into block chunks and use threads for each block. Each child process runs aho-corasick algorithm for each block. As we are using parallel processing it increases the execution speed compared to aho-corasick algorithm. In parallel block aho-corasick we use less number of threadscompared to parallel failure less aho-corasick so parallel block aho-corasick algorithm will have less thread over head compared to parallel failure less aho-corasick algorithm.

In [10] pfac we have made more no of comparisons compared to aho-corasick. In aho-corasick everything is computed serially nothing is done in parallel, so here we are splitting data into blocks and assigning each block toa thread and for each block aho-corasick is used by this we are reducing no of comparisons performed.

B. *Using Bloom Filters to reduce search space*

The proposed improvement is by using of bloom filters we create a filter for a unique length using all patterns of that length and use that filter to check for all the possible texts of that length generated from sequence if that text passes the filter only we perform exact pattern matching. Once we enter aho-corasick we will follow aho-corasick until we reach start state because of failures. [11] This we will reduce the number of times accessing the finite automata and also bloom filter requires less space compared to other data structures as bloom filters do not store data items.



4.6.1 Finite state automata

Initialize : transitionMap, failureMap, and outputMap

Input : List of Patterns **Output** : A Finite State automata of Input patterns P_{new}

Algorithm 1 : Finite state automata

```
for pattern in ListOfKeywords do
    for character ch in pattern do
        Add ch in the set of characters
    end for
    Initialize currentState and i as 0
    while (currentState, pattern[i]) is present in transitionMap do
        currentState = transitionMap.get(currentState, pattern[i])
        i=i+1
    end while
    for remaining character ch in pattern do
        Add (currentState, ch) into transitionMap
        currentState = currentState + 1
    end for
end for
for each state in listOfStates do
    remove state form listOfStates
    for character ch in setOfCharacters do
        if (currentState, ch) is present in transitionMap then
            nextState = transitionMap.get(currentState, ch)
            add nextState in listOfStates
            failureState = failureMap.get(currentState)
            while failureState is not present in transitionMap do
                failureState = failureMap.get(failureState)
            end while
            failureMap.put(nextState, failureState)
            merge outputMap.get(nextState) and outputMap.get(failureState)
        end if
    end for
end for
```

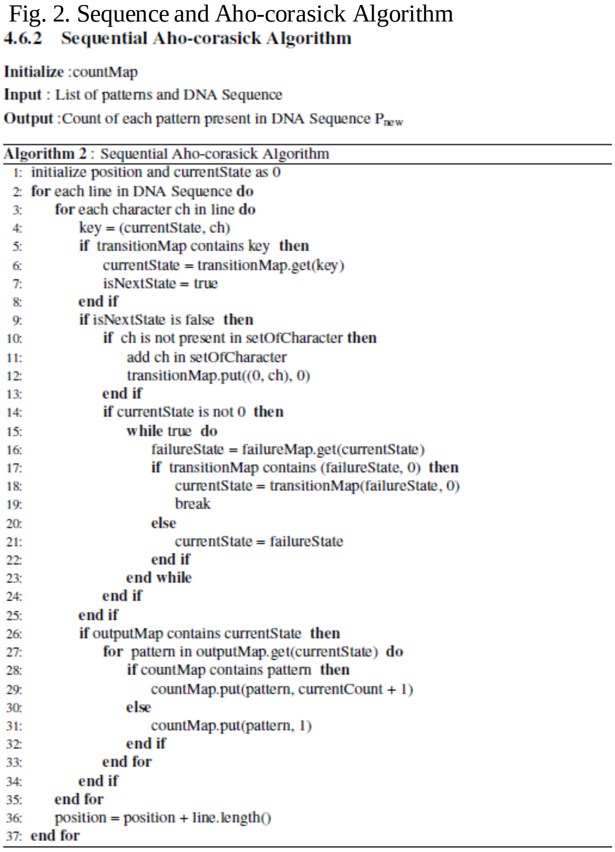
Fig. 1.Finite State Automata

C. Implementation

1) Aho-corasick algorithm: In Aho-corasick algorithm first nucleotide sequences (patterns needed to search) are given as input using these patterns finite state automata is constructed. [12] A finite automaton contains go to and failure links .In finite state automata, the go to function populates a transition map that stores the transition from one state to the next. Failure function creates a failure map according to the Aho-corasick Algorithm. Failure transition. For any undefined transitions, the start state will revert to start.

2) Parallel failure less aho-corasick algorithm: [17] Parallel failure less aho-corasick algorithm is avariation of aho-corasick algorithm in which we use size of the input pattern number of parallel threads each thread corresponding to each base of input sequence . Each thread checks whether there is a pattern starting from corresponding base in the trie. In this algorithm thread ends when there is a failure state and when it finds the pattern. In parallel failure less aho-corasick algorithm there will not be any failure transition states. Failure links are not needed in the The trie [15].This algorithm needs more computation than aho-corasick algorithm as we use pre-computed failure states in aho-corasick it has less computations but as we are using many threads it will execute faster than aho-corasick. refer algorithm in Fig.2.

Fig. 2. Sequence and Aho-corasick Algorithm



4.6.2 Sequential Aho-corasick Algorithm

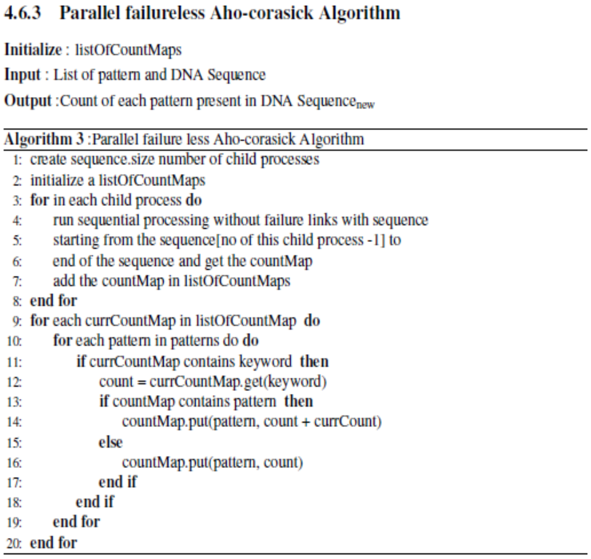
Initialize :countMap

Input : List of patterns and DNA Sequence

Output :Count of each pattern present in DNA Sequence P_{new}

Algorithm 2 : Sequential Aho-corasick Algorithm

```
initialize position and currentState as 0
for each line in DNA Sequence do
    for each character ch in line do
        key = (currentState, ch)
        if transitionMap contains key then
            currentState = transitionMap.get(key)
            isNextState = true
        end if
        if isNextState is false then
            if ch is not present in setOfCharacter then
                add ch in setOfCharacter
                transitionMap.put((0, ch), 0)
            end if
            if currentState is not 0 then
                while true do
                    failureState = failureMap.get(currentState)
                    if transitionMap contains (failureState, 0) then
                        currentState = transitionMap(failureState, 0)
                        break
                    else
                        currentState = failureState
                    end if
                end while
            end if
        end if
        if outputMap contains currentState then
            for pattern in outputMap.get(currentState) do
                if countMap contains pattern then
                    countMap.put(pattern, currentCount + 1)
                else
                    countMap.put(pattern, 1)
                end if
            end for
        end if
    end for
    position = position + line.length()
end for
```



4.6.3 Parallel failureless Aho-corasick Algorithm

Initialize : listOfCountMaps

Input : List of pattern and DNA Sequence

Output :Count of each pattern present in DNA Sequence_{new}

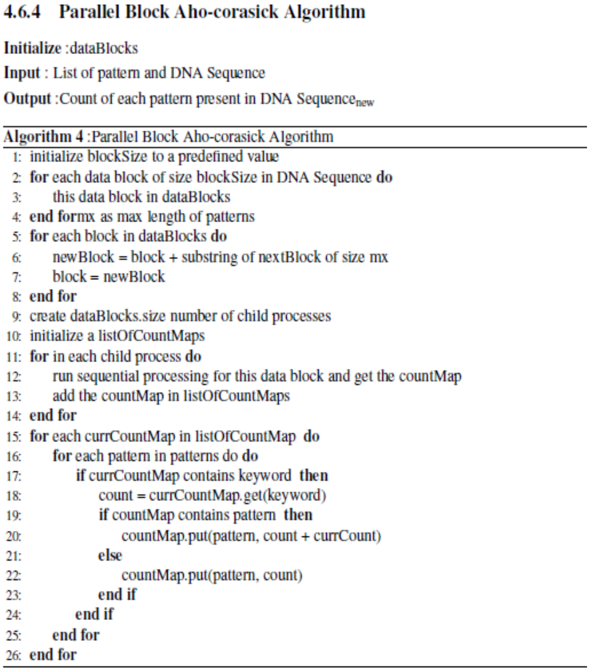
Algorithm 3 :Parallel failure less Aho-corasick Algorithm

```
create sequence.size number of child processes
initialize a listOfCountMaps
for in each child process do
    run sequential processing without failure links with sequence
    starting from the sequence[no of this child process -1] to
    end of the sequence and get the countMap
    add the countMap in listOfCountMaps
end for
for each currCountMap in listOfCountMap do
    for each pattern in patterns do do
        if currCountMap contains keyword then
            count = currCountMap.get(keyword)
            if countMap contains pattern then
                countMap.put(pattern, count + currCount)
            else
                countMap.put(pattern, count)
            end if
        end if
    end for
end for
```

Fig. 3. Parallel failure less aho-corasick Algorithm

3) Parallel Hashed failure less Aho-corasickAlgorithm: The proposed improvement is by using of bloom filters we create a filter for a unique length using all patterns of that length and use that filter to Check for all the possible texts of that length generated from. Sequence from that starting point if only text passes the filter we perform exact pattern matching.

Before creating a thread in pfac if we check for hash values of sequences generated from that starting point. This we will reduce the number of times accessing the finite automata and also bloom filter requires less space compared to other data structures as bloom filters do not store data items. Refer algorithm in Fig.5.



4.6.4 Parallel Block Aho-corasick Algorithm

Initialize :dataBlocks

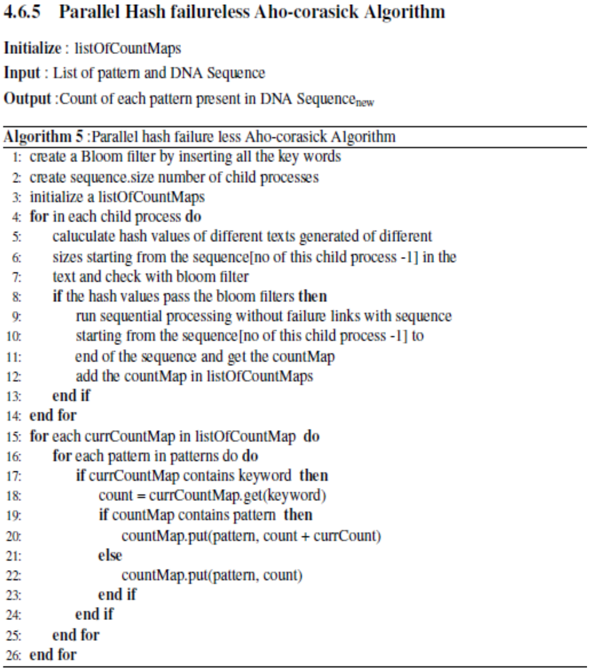
Input : List of pattern and DNA Sequence

Output :Count of each pattern present in DNA Sequence$_{new}$

Algorithm 4 :Parallel Block Aho-corasick Algorithm

```
initialize blockSize to a predefined value
for each data block of size blockSize in DNA Sequence do
    this data block in dataBlocks
end formx as max length of patterns
for each block in dataBlocks do
    newBlock = block + substring of nextBlock of size mx
    block = newBlock
end for
create dataBlocks.size number of child processes
initialize a listOfCountMaps
for in each child process do
    run sequential processing for this data block and get the countMap
    add the countMap in listOfCountMaps
end for
for each currCountMap in listOfCountMap  do
    for each pattern in patterns do do
        if currCountMap contains keyword  then
            count = currCountMap.get(keyword)
            if countMap contains pattern  then
                countMap.put(pattern, count + currCount)
            else
                countMap.put(pattern, count)
            end if
        end if
    end for
end for
```

Fig. 4. Parallel Block aho-corasick Algorithm



4.6.5 Parallel Hash failureless Aho-corasick Algorithm

Initialize : listOfCountMaps

Input : List of pattern and DNA Sequence

Output :Count of each pattern present in DNA Sequence$_{new}$

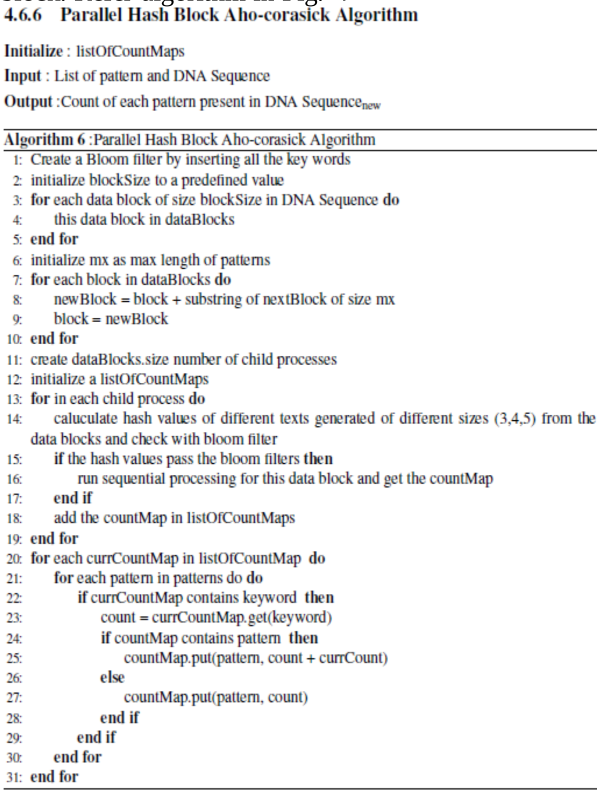
Algorithm 5 :Parallel hash failure less Aho-corasick Algorithm

```
create a Bloom filter by inserting all the key words
create sequence.size number of child processes
initialize a listOfCountMaps
for in each child process do
    caluculate hash values of different texts generated of different
    sizes starting from the sequence[no of this child process -1] in the
    text and check with bloom filter
    if the hash values pass the bloom filters then
        run sequential processing without failure links with sequence
        starting from the sequence[no of this child process -1] to
        end of the sequence and get the countMap
        add the countMap in listOfCountMaps
    end if
end for
for each currCountMap in listOfCountMap  do
    for each pattern in patterns do do
        if currCountMap contains keyword  then
            count = currCountMap.get(keyword)
            if countMap contains pattern  then
                countMap.put(pattern, count + currCount)
            else
                countMap.put(pattern, count)
            end if
        end if
    end for
end for
```

Fig. 5. Parallel Hash failure less Aho-corasick Algorithm

4) Parallel block Aho-corasick Algorithm: In the parallel variant of the Aho-corasick algorithm, we break the input DNA sequence into a fixed number of blocks, refer the Fig 3. The block size is a predefined value and it depends on the number of the parallel processing units of the system. Each child process will

We create a child process for each block of the DNA sequence. [13] All these child processes will run in parallel in a multi- processing system. Each child process will run the sequential variant of the Aho-corasick algorithm and populates the count of occurrences of each pattern in that particular data block. Refer algorithm in Fig. 4



4.6.6 Parallel Hash Block Aho-corasick Algorithm

Initialize : listOfCountMaps

Input : List of pattern and DNA Sequence

Output :Count of each pattern present in DNA Sequence$_{new}$

Algorithm 6 :Parallel Hash Block Aho-corasick Algorithm

```
Create a Bloom filter by inserting all the key words
initialize blockSize to a predefined value
for each data block of size blockSize in DNA Sequence do
    this data block in dataBlocks
end for
initialize mx as max length of patterns
for each block in dataBlocks do
    newBlock = block + substring of nextBlock of size mx
    block = newBlock
end for
create dataBlocks.size number of child processes
initialize a listOfCountMaps
for in each child process do
    caluculate hash values of different texts generated of different sizes (3,4,5) from the
data blocks and check with bloom filter
    if the hash values pass the bloom filters then
        run sequential processing for this data block and get the countMap
    end if
    add the countMap in listOfCountMaps
end for
for each currCountMap in listOfCountMap  do
    for each pattern in patterns do do
        if currCountMap contains keyword  then
            count = currCountMap.get(keyword)
            if countMap contains pattern  then
                countMap.put(pattern, count + currCount)
            else
                countMap.put(pattern, count)
            end if
        end if
    end for
end for
```

Fig. 6. Parallel Hash Block aho-corasick Algorithm

5) Parallel hash block Aho-corasick Algorithm: The DNA sequence file is split into many blocks. As a result, the file's data will be split into many data chunks. Each block is given a thread, which is then assigned to it according to its availability. After that, each block is sent to automation to look for patterns. Aho-corasick algorithm is run on each block. Refer algorithm in Fig 6.

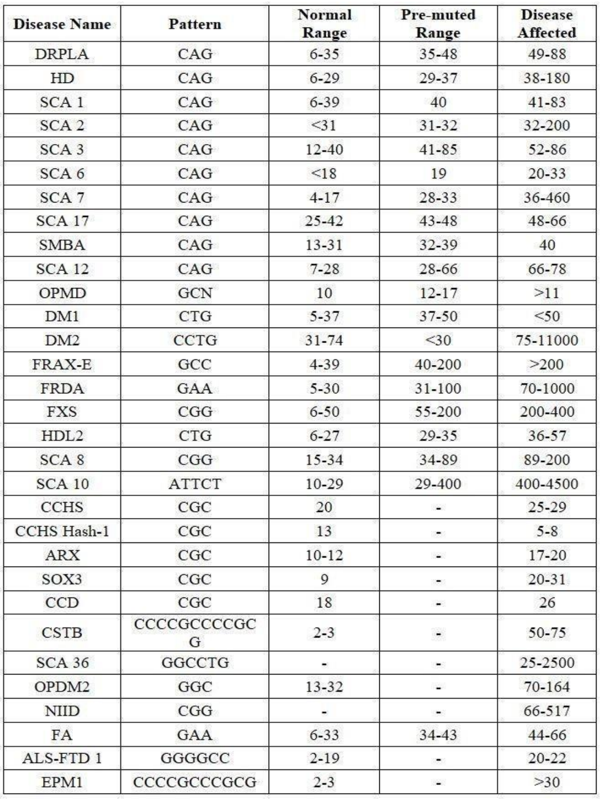
5. Result Analysis

1) Dataset: In this research we have used standard datasets from National center of biotechnology information (NCBI) [8] and some standard Genome projects [9]. The genes used for the purpose of study were downloaded from NCBI having gene id 1760, 2175, 2395, 2332, 8106, 25814, 4287 [14]. All these genes are classified as 'Homo sapiens' under organism.

2) Results: Table 1 shows the diseases caused by nucleotide repeats as well as the lengths of the repeats and possible molecular biochemical processes that induce sickness. The table presents several patterns together with their related ranges and diseases, allowing us to deduce if a certain DNA sequence is affected by specific diseases, pre-mutated, or normal. [15] The frequency of a pattern in a DNA sequence or

genome is represented by the ranges here. If we wish to determine the likelihood of nucleotide repeat disorders based on these standard ranges, we should compare the mentioned standard ranges.

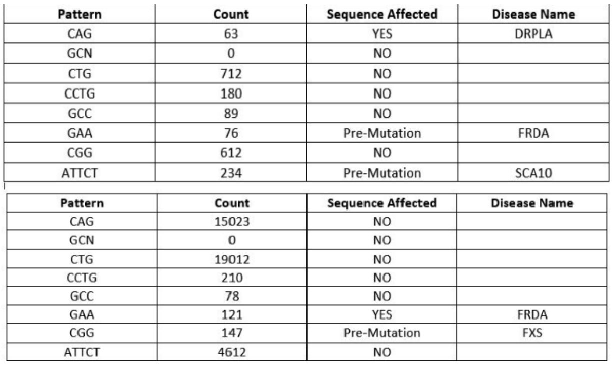
[16] We infer the medical condition of the patient based on the collected results as disease(s) infected, permuted, or disease-free. The pre- muted predicts the likelihood of diseases occurring in the primary stage, allowing the patient to become aware and receive treatment. As a result, early state diagnosis is beneficial in subsequent treatment.



Disease Name	Pattern	Normal Range	Pre-muted Range	Disease Affected
DRPLA	CAG	6-35	35-48	49-88
HD	CAG	6-29	29-37	38-180
SCA 1	CAG	6-39	40	41-83
SCA 2	CAG	<31	31-32	32-200
SCA 3	CAG	12-40	41-85	52-86
SCA 6	CAG	<18	19	20-33
SCA 7	CAG	4-17	28-33	36-460
SCA 17	CAG	25-42	43-48	48-66
SMBA	CAG	13-31	32-39	40
SCA 12	CAG	7-28	28-66	66-78
OPMD	GCN	10	12-17	>11
DM1	CTG	5-37	37-50	<50
DM2	CCTG	31-74	<30	75-11000
FRAX-E	GCC	4-39	40-200	>200
FRDA	GAA	5-30	31-100	70-1000
FXS	CGG	6-50	55-200	200-400
HDL2	CTG	6-27	29-35	36-57
SCA 8	CGG	15-34	34-89	89-200
SCA 10	ATTCT	10-29	29-400	400-4500
CCHS	CGC	20	-	25-29
CCHS Hash-1	CGC	13	-	5-8
ARX	CGC	10-12	-	17-20
SOX3	CGC	9	-	20-31
CCD	CGC	18	-	26
CSTB	CCCCGCCCCGC G	2-3	-	50-75
SCA 36	GGCCTG	-	-	25-2500
OPDM2	GGC	13-32	-	70-164
NIID	CGG	-	-	66-517
FA	GAA	6-33	34-43	44-66
ALS-FTD 1	GGGGCC	2-19	-	20-22
EPM1	CCCCGCCCGCG	2-3	-	>30

Fig. 7. Table 1

Table 1 illustrates the number of repeat patterns (keywords) in patient1's gene. We can see from the table that the chances of being affected by FRDA, FXS are higher because GAA has a frequency of 121 and CGG has a frequency of 147. Count 121 of GAA is in the impacted range, while CGG is in the pre-mutated area, according to the frequency table. As a result, the sequence is FRDA-affected and may result in FXS.



Pattern	Count	Sequence Affected	Disease Name
CAG	63	YES	DRPLA
GCN	0	NO	
CTG	712	NO	
CCTG	180	NO	
GCC	89	NO	
GAA	76	Pre-Mutation	FRDA
CGG	612	NO	
ATTCT	234	Pre-Mutation	SCA10

Pattern	Count	Sequence Affected	Disease Name
CAG	15023	NO	
GCN	0	NO	
CTG	19012	NO	
CCTG	210	NO	
GCC	78	NO	
GAA	121	YES	FRDA
CGG	147	Pre-Mutation	FXS
ATTCT	4612	NO	

Fig. 8. Table 2

Table 2 indicates the number of repeat patterns (keywords) in patient2's gene. We can see from the table that the chances of being affected by FRDA, DRPLA, and SCA10 are higher because GAA is 76, ATTCT is 234, and CAG is 63. GAA is in the pre- mutation range, according to the frequency chart. GAG is in the impacted range, but ATTCT is in the pre- mutated range. As a result, the sequence is DRPLA- affected, with the possibility of FRDA and SCA10.

Graph 1 indicates the improvement in time on hashing the sequence before performing PFAC. The difference in time is less when the file size is small. As the no of characters in the sequence increases we can see the effects of hashing Fig. 9. Table 3 and the difference in time taken by PFAC and Hashed- PFAC increases.

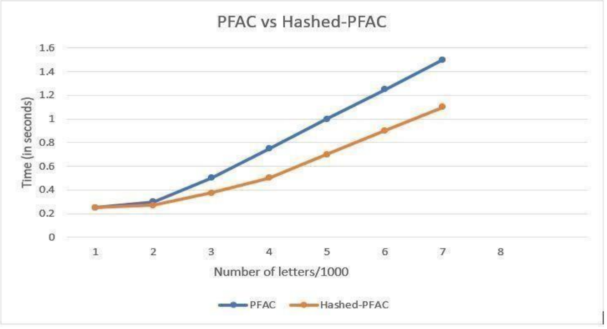


Fig. 9. Graph 1

Graph 2 indicates the improvement in time on hashing the sequence before performing Chunked-Aho-corasick. The difference in time is less when the file size is small. As the no of characters in the sequence increases we can see the effects of hashing and the difference in time taken by PFAC and Chunked-PFAC increases.

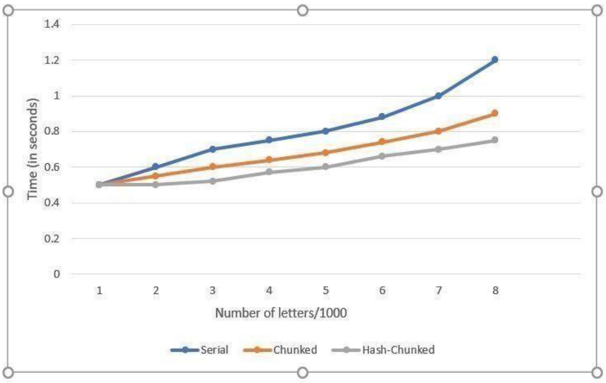


Fig. 10. Graph 2

6. Conclusion and Future Work

The report examines a simple and effective nucleotide repeat identification approach. Based on his or her DNA, the obtained data is utilized to evaluate ifan individual has a chance of developing the disease(s) in the future. The presented chucked hash parallel Aho-corasick algorithm performs better than all the other .As we are using more number of threads we create more over head using pfac and hashed pfac. In future we can use pattern division and sequence division and assign to threads for improved performance. We can implement this algorithms can implemented in GPGPU. The collective analysis can be designed for community genome testing and analyzing the larger dataset to achieve key findings such as often occurring diseases, malignancies, and trends in the results, among other things.

References

[1] Hongjian Li, Bing Ni, Man-Hon Wong, Kwong-Sak Leung "A fast CUDA implementation of Agrep algorithm for approximate nucleotide sequence matching" IEEE 9th Symposium on Application Specific Processors , vol. 34, pp 978-1-4577-1212-8, 2011.

[2] Raju Bhukya and DVLN Somayajulu., "Exact multiple pattern matching algorithm using DNA sequence and pattern pair," In: International Journal of Computer Applications, vol. 17.8, pp. 32–38, 2021.

[3] Burton H Bloom. "Space/time trade-offs in hash coding with allowable errors," In: Communications of the ACM, vol. 59, pp. 422– 426, 1970.

[4] Raju Bhukya, Balram Parmer, Anand Kulkarni," An Index Based Skip Search Multiple Pattern Matching Algorithm", International Journal on Computer Science and Engineering, vol. 3, pp. 1510-1517, 2011.

[5] Donald E Knuth, James H Morris Jr, and Vaughan R Pratt, "Fast pattern matching in strings," In: SIAM journal on computing, vol. 6, pp. 323–350, 1977.

[6] Sandeep U Mane and Ketaki H Pangu., "Disease di- agnosis using pattern matching algorithm from DNA sequencing: a sequential and GPGPU based approach," In: Proceedings of the International Conference on Informatics and Analytics, pp. 1– 5, 2016.

[7] Maleeha Najam et al., "Pattern matching for DNA sequencing data using multiple bloom filters," In: BioMed research international, vol. 27, no. 3, pp. 1-17, 2019.

[8] Mukku Nisanth Kartheek, Munaga VNK Prasad, Raju Bhukya, "Local optimal oriented pattern for person independent facial expression recognition," Twelfth international conference on machine vision (ICMV 2019), vol. 11433, pp. 195-202, 2020.

[9] Chandra Mohan Dasari, Raju Bhukya., "Explainable deep neural networks for novel viral genome prediction," Applied Intelligence, vol. 52, pp. 3002-3017, 2022.

[10] Chandra Mohan Dasari, Raju Bhukya., "Intersspp: investigating patterns through interpretable deep neural networks for accurate splice signal prediction," Chemometrics and Intelligent Laboratory Systems, vol. 206, pp. 104-144, 2020.

[11] Santhosh Amilpur, Raju Bhukya, "Edeepssp: explainable deep neural networks for exact splice sites prediction," *Journal of Bioinformatics and Computational Biology*, vol. 18, pp. 2050024, 2020.

[12] Chandra Mohan Dasari, Santhosh Amilpur, Raju Bhukya, "Exploring variable-length features (motifs) for predicting binding sites through interpretable deep neural networks," Engineering Applications of Artificial Intelligence, vol. 106, pp.104485,2021.

[13] Kaushik Bhargav Sivangi, Chandra Mohan Dasari, Santhosh Amilpur, Raju Bhukya, "NoAS- DS: Neural optimal architecture search for detection of diverse DNA signals," Neural Networks, vol. 147, pp. 63-71, 2022.

[14] Chandra Mohan Dasari, Raju Bhukya, "Map Reduceparadigm: DNA sequence clustering based on repeats as features," Expert Systems, vol. 29, pp. e12827, 2022.

[15] Mukku Nisanth Kartheek, Munaga VNK Prasad, Raju Bhukya., "Chess pattern with different weighting schemes for person independent facial expression recognition," Multimedia Tools and Applications, vol. 81,pp. 22833-22866, 2022.

[16] Lin C., Chang C., and Wang, Z., "Reversible Data Hiding Scheme Using Adaptive Block Truncation Coding Based on an Edge-Based Quantization Approach," Symmetry, vol. 11, no. 6, pp. 765, 2019.

[17] Chandra Mohan Dasari, Raju Bhukya, "Disease Diagnosis based on Various Pattern Frequency from Extracted Exons" IEEE 3rd Global Conference for Advancement in Technology(GCAT), vol. 29, pp. 978-1-6654-6855, 2022.

GranReg: Granger-Regression-based Inference of Gene Regulatory Network with Seasonal Differencing

Emma Paul
Bioinformatics Lab, Department of Computer Science
Cochin University of Science and Technology
Kochi, India
emmapaul1993@cusat.ac.in

Jereesh A. S.
Bioinformatics Lab, Department of Computer Science
Cochin University of Science and Technology
Kochi, India
jereesh@cusat.ac.in

G. Santhosh Kumar
Bioinformatics Lab, Department of Computer Science
Cochin University of Science and Technology
Kochi, India
san@cusat.ac.in

***Abstract*— Gene network inference deals with the construction of gene regulatory networks from genomic data and is one of the long-standing critical problems in computational biology. Regression techniques have proven to be one of the most efficient methods for inference. This study proposes a granger causality and regression-based pipeline for inferring GRNs. The Dialogue for Reverse Engineering Assessments and Methods (DREAM) challenge provides benchmark datasets on which algorithms for network construction can be tested. Our method resorts to datasets provided in the DREAM4 challenge. A two-way pipeline, first with seasonal differencing and granger causality and the second with regression importance scores was used to calculate the final network edge scores. The result of the proposed computational framework works effectively in terms of AUPR and AUROC values on size 10 and size 100 networks of the DREAM4 challenge and is competitive to the existing algorithms. The AUPR and AUROC values are comparable to GreyNet in size 10 network and outperforms GreyNet for size 100 dataset.**

Keywords—Gene Expression, Gene regulatory network, Regression, Granger causality.

I. Introduction

Gene Regulatory Network (GRN) inference is a major research area in computational biology. Genes code for a large number of products that our body requires, these products are in turn able to affect other products and can even involve self-loops affecting the very same gene from which it all started. Inferring this entire regulatory network interaction from gene expression data using experimentation is infeasible. Computational methods strategically reduce the search space to the most probable sets of relations which can then be experimentally verified. This enables speeding up the advancement of understanding disease prognosis and disease progress prediction. The paper's main contribution is a simple dual pipeline capable of improving the accuracy of GRN inference on the benchmark DREAM4 datasets compared to recent works. The first part focuses on statistical analysis utilizing seasonal differencing followed by granger causality, which has been found to perform well on gene regulatory networks [1]. Granger causality helps to find out if a time series variable is capable of forecasting another one. The second part of the pipeline contains a feature scoring using the regression technique. The final score is aggregated to form the network.

The paper covers the background, briefly describing the commonly used methods for gene regulatory network construction, followed by materials and methods, which discusses the dataset and methodology. The results and discussion section compare the proposed approach with existing techniques, followed by a conclusion and references.

II. Background

A. Information-theoretic approaches

This category includes methods that incorporate techniques like correlation, entropy [2], mutual information (MI) [3], conditional mutual information (CMI) [4], conditional mutual inclusive information (CMI2) [5] and so on. The major works in this area are relevance network [6], MIBNI [7], ARACNE [8], CLR [9] and MRNET [10].

B. Bayesian network models

Bayesian networks (BNs) comprise both the diagram and the conditional probability table. There is a conditional probability table given the parents for each variable in the diagram. The arrows in BN denote the "forward" probability and the usage of Bayes' rule finds the inverse probability, by taking the product of the prior with the likelihood. As for the variables without parents, their prior probability can summarize their cause. Dynamic Bayesian Networks (DBNs) comprise a class of techniques that can elegantly deal with the loop taboo of BNs. DBNs tactfully solve the DAG problem by expanding the variable set based on specific time points. The major works in this area are BGRMI [11], FBISC [12], KBOOST [13], IMBDANET [14], Intelligent verification with Machine Learning and Model checking [15].

C. Regression models

Regression and feature selection methods deal with network inference by considering one target gene at a time and recognizing potential regulators that best predict the target gene state [15]. GENIE3 [18], GENIE3-time [21], GNIPLR [20], KBOOST [13], BTNET [19], BMALR [17] and TIGRESS [16] are some of the main works under regression models.

III. Materials and Methods

The gene regulatory network which is reconstructed from gene expression data can be represented using G = (V, E) where V represents the vertices and E represents the edges. The edge $e_{ij} \in E$, represents the interaction between gene g_i to gene g_j. Let D denote the time series dataset for the network reconstruction problem,

$$D= \{D_1, D_2, . . ., D_N\} \qquad (1)$$

where D_i denotes the time series data for sample i. Each sample in this scenario is assumed to have an equal number of time points, T.

The samples Di, where i= 1 to N is then concatenated to form a single time series input with N*T time points, which will be referred to as P. Dataset D, can be now denoted at row level as,

$$g(t_k) = \{g_1(t_k), g_2(t_k), \ldots, g_M(t_k)\} \qquad (2)$$

where $g_i(t_k)$ denotes the expression values of gene i at time point k. M denotes the total number of genes in the dataset. $k \in P$ denotes the time point index.

A. Dataset used

Dream 4[22][23][24] network inference challenge data set was used for the analysis.

B. Gene expression data

Microarray keeps track of thousands of genes in parallel and produces output raw files in image format which can then be transformed into gene expression matrices after data pre-processing. In these matrices the table rows represent samples and the column represents the genes; the number in the cell characterizes the value of that specific gene in the corresponding row sample. While static data has no time component, temporal aka time-series data have interactions that vary across time. Each link can have attributes denoting its active time and interaction strength at different chronological points. For this study, the main focus is on time series datasets. DREAM4 challenge dataset has five size 10 and ten size 100 node datasets. Size 10 has 10 genes with time series information on 21-time points. Size 100 has 100 genes on 21-time points.

C. Performance metrics:

We validate our method with other state of art methods like GreyNet [28], BiXGBoost [29] and Jump3 [30] by AUROC and AUPR scores. AUROC corresponds to the area under the ROC curve based on TPR and FPR. AUPR is the area under the PR based on precision and recall. Another comparison with GreyNet at 2%,5%,10%,15% and 20% threshold was performed using additional metrics such as F1-Score, Accuracy and MCC (Matthew's correlation coefficient). The density of the gene regulatory network varies from organism to organism, the choice of threshold hence is not specific to any organism since that is not the main purpose of this study.

$$\text{Recall} = \text{TPR} = \frac{TP}{TP+FN} \qquad (3)$$

$$\text{FPR} = \frac{FP}{FP+TN} \qquad (4)$$

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (5)$$

$$\text{F}_1 \text{ score} = \frac{2.TP}{2.TP++FN+FP} \qquad (6)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FN+FP} \qquad (7)$$

$$\text{MCC} = \frac{TP.TN-FP.FN}{\sqrt{(TP+FP).(TP+FN).(TN+FP).(TN+FN)}} \qquad (8)$$

Here TPR denotes the true positive rate, FPR represents False positive rates, TP is the true positive, TN is the true negative FP is the false positive and FN is the false negative. The AUPR and AUROC values for each method are averaged over 10 runs for each network.

D. Preprocessing

The dataset of size 10 samples was concatenated to form a 105x10 matrix where 105 represents the time points and 10 represents the gene count. Similarly, size 100 samples were concatenated to get a 210x100 matrix.

E. Granger causality pipeline

Traditional approaches to time series analysis work with stationary input assumptions. Differencing is a technique used to stabilize the mean of the time series and compute the differences between successive observations. Seasonal differencing is performed between observation and one at the lagged time point in the previous season [27], here the seasonal period is set 21-time points which represents a sample. In this case, the seasonal differencing can be represented as

$$g'_k = g_k - g_{k\text{-lag}} \qquad (9)$$

The lag value of seasonal differencing hence corresponds to 21-time points. This step is followed by the KPSS stationarity test [26] to see if the series is stationary. A time series variable X is judged to be the cause of another variable Y if past values of X help in predicting future values of Y, i.e, the past values of Y and X achieve greater forecasting power to predict future Y than what would have been achieved by past values of Y alone. Granger causality was applied to the stationary data and significant interactions were scored with a value of 1 (p-value <= 0.05). As Granger causality is not true causality and since only pairwise linear interactions are covered with the test the remaining insignificant interactions were given a score value of 0.5. In the granger causality pipeline, one variable is set as a regulator and is used to predict the other one which is the target. On each loop, an edge is covered and after the full run, the edge score for the fully connected network is obtained.

F. Regression pipeline:

The regression pipeline uses the same concatenated dataset but after simple exponential smoothing [27] using an exponential window. Exponential smoothing is given by the formula,

$$e_t = \alpha x_t + (1-\alpha)e_{t-1} \qquad (10)$$

Multiple regression analyses were used to calculate the feature importance values, including Random Forest, XGBoost, SVM, Lasso and Ridge. Out of these methods, Random Forest performed the best. The comparisons below with existing approaches are done using the random forest method. On each iteration of the random forest, one gene is selected as the target and the rest posing as regulators. The random forest approach then calculates the feature importance scores for each regulator-target interaction. These feature scores are used to calculate the final network. This way the GRN inference is divided into multiple subproblems. Finally, the feature importance scores from each subproblem are integrated along with granger causality significance scores to create the final network. The scores from both branches were multiplied to obtain the final interaction score. The workflow of the proposed method is provided in Fig. 1.

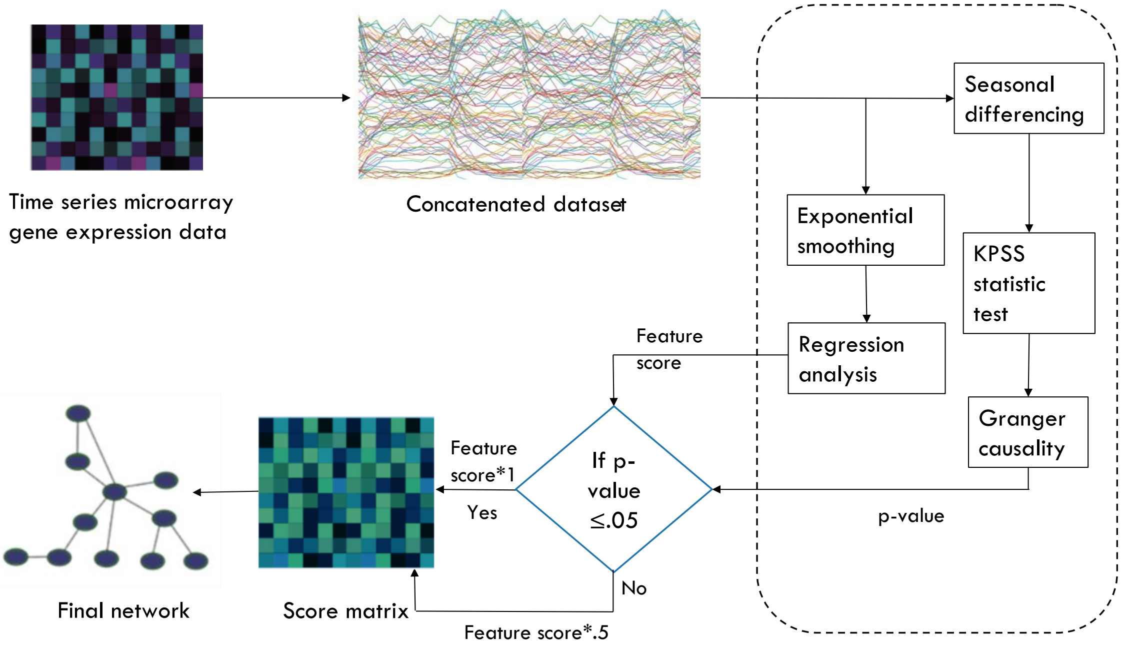


Fig. 1: GranReg workflow

IV. Results and Discussion

Existing techniques named GreyNet, BiXGBoost and Jump3 were applied to time series data, the results below are obtained by averaging on 10 runs. The radar plot in Fig. 2 shows the AUPR and AUROC values of the Proposed method over the existing approaches in the size 10 and size 100 datasets.

The AUPR and AUROC values for the GreyNet, BiXGBoost, Jump3 and random network against the proposed method are provided in Table 1. After considering the threshold of 2%, 5%, 10%, 15% and 20%, Precision, Recall, F1 score, MCC and Accuracy values were calculated. These are provided in Fig. 3 and Fig. 4.

As the number of nodes increases there is a clear drop in precision value. The FP vs FN plot in Fig. 5 and Fig. 6 shows that the number of false positives increases as the threshold is increased. This causes a drop in precision. Also, as the size of the dataset increases more edges are detected as false positives. FN decreases as the threshold is increased and for Recall values of size 100 dataset increases compared to Precision values.

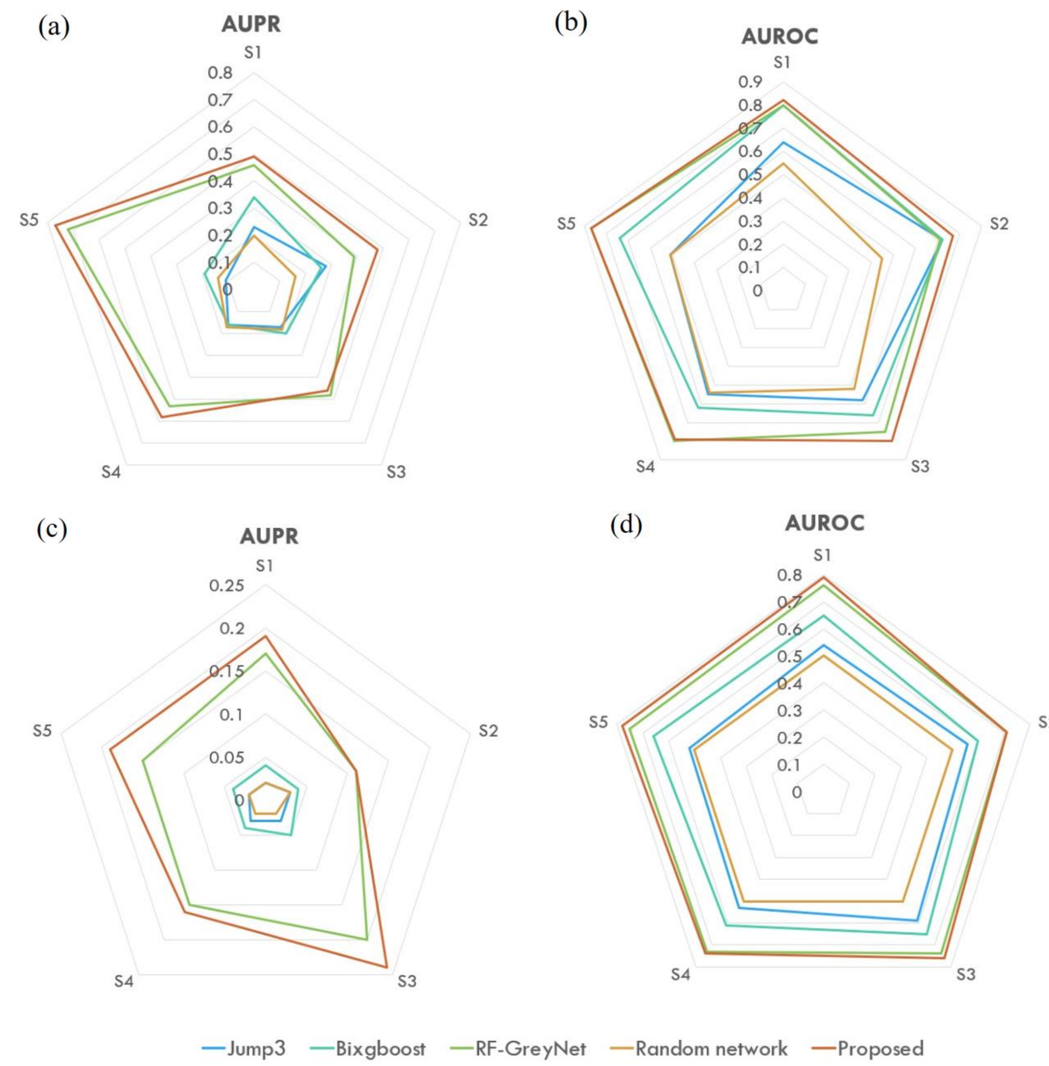


Fig. 2: AUPR and AUROC value comparison between the proposed method against GreyNet, BiXGBoost, Jump3 and Random network for size 10 and size 100 dataset for 5 subnetworks (S1, S2, S3, S4, S5). Fig. 2(a) and 2(b) correspond to the size 10 network and Fig. 2 (c) and 2(d) correspond to the size 100 dataset.

TABLE 1: AUPR AND AUROC SCORE COMPARISON BETWEEN 3 EXISTING METHODS, THE PROPOSED METHOD AND RANDOM NETWORK CONSTRUCTION. THE MEAN AUPR AND AUROC VALUES ON 10 RUNS WITH A 95% CONFIDENCE INTERVAL ARE PROVIDED BELOW.

Size 10		S1	S2	S3	S4	S5
Jump3	AUPR	0.23±0.009	0.28±0.007	0.17±0.002	0.16±0.005	0.11±0.003
	AUROC	0.64±0.012	0.72±0.016	0.58±0.006	0.55±0.013	0.51±0.017
BiXGBoost	AUPR	0.34±0.01	0.26±0.008	0.2±0.008	0.16±0.008	0.19±0.014
	AUROC	0.8±0.008	0.72±0.01	0.66±0.018	0.62±0.022	0.74±0.026
GreyNet- RF	AUPR	0.46±0.006	0.39±0.005	**0.48±0.006**	0.53±0.006	0.72±0.006
	AUROC	0.8±0.003	0.71±0.003	0.75±0.003	**0.8±0.002**	**0.87±0.003**
Proposed method	AUPR	**0.49±0.012**	**0.48±0.012**	0.46±0.006	**0.58±0.022**	**0.77±0.012**
	AUROC	**0.82±0.004**	**0.77±0.008**	**0.8±0.004**	0.79±0.004	**0.87±0.008**
Random approach	AUPR	0.2±0.037	0.16±0.041	0.18±0.06	0.17±0.043	0.14±0.028
	AUROC	0.55±0.095	0.45±0.094	0.52±0.09	0.54±0.076	0.51±0.066
Size 100		S1	S2	S3	S4	S5

Jump3	AUPR	0.02±0.001	0.03±0.0	0.03±0.001	0.03±0.003	0.02±0.0
	AUROC	0.54±0.004	0.56±0.004	0.59±0.003	0.53±0.002	0.52±0.002
BiXGBoost	AUPR	0.04±0.002	0.04±0.002	0.05±0.002	0.04±0.002	0.04±0.002
	AUROC	0.65±0.012	0.6±0.012	0.65±0.016	0.61±0.018	0.66±0.014
GreyNet- RF	AUPR	0.17±0.008	**0.11±0.005**	0.2±0.004	0.15±0.004	0.15±0.006
	AUROC	0.76±0.001	**0.71±0.002**	0.74±0.001	0.73±0.002	0.75±0.002
Proposed method	AUPR	**0.19±0.006**	**0.11±0.003**	**0.24±0.002**	**0.16±0.004**	**0.19±0.005**
	AUROC	**0.79±0.001**	**0.71±0.002**	**0.76±0.002**	**0.74±0.002**	**0.78±0.001**
Random approach	AUPR	0.02±0.002	0.03±0.001	0.02±0.002	0.02±0.002	0.02±0.002
	AUROC	0.5±0.017	0.5±0.016	0.5±0.011	0.5±0.018	0.5±0.021

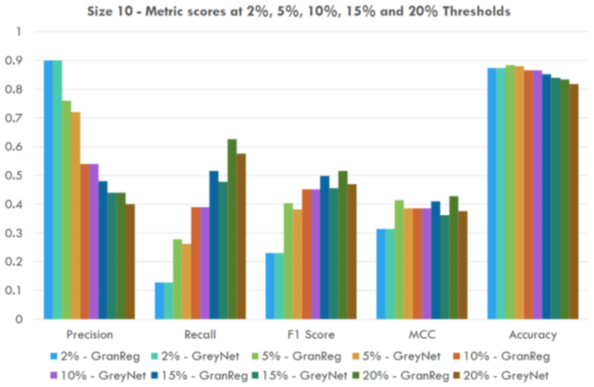


Fig. 3: Precision, recall, F1 score, MCC and Accuracy values were calculated at 2%, 5%, 10%, 15% and 20%. The values are averaged over the 5 sub-networks S1, S2, S3, S4 and S5 of the size 10 dataset

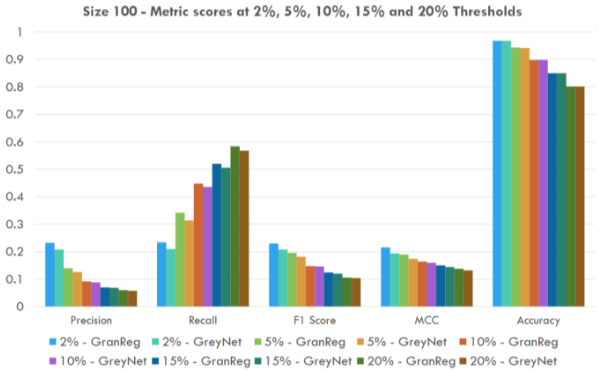


Fig. 4: Precision, recall, F1 score, MCC and Accuracy values were calculated at 2%, 5%, 10%, 15% and 20%. The values are averaged over the 5 sub-networks S1, S2, S3, S4 and S5 of the size 10 dataset.

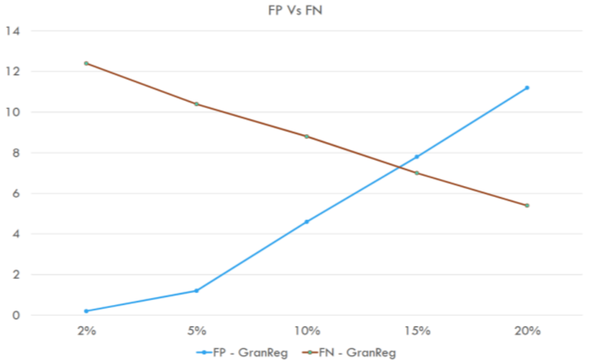


Fig. 5: FP vs FN plot for size 10 datasets. The thresholds used are 2%, 5%, 10%, 15% and 20% and the FP and FN values are averaged for the 5 subnetworks under size 10.

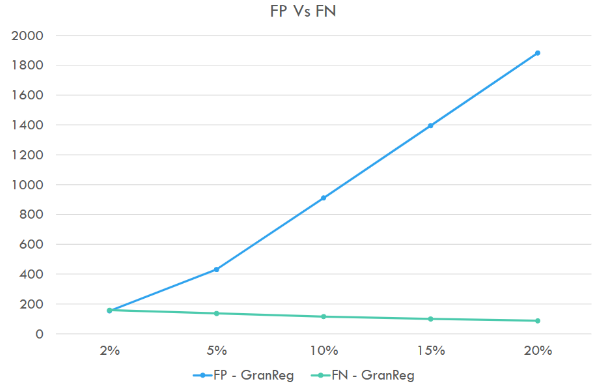


Fig. 6: FP vs FN plot for size 100 datasets. The thresholds used are 2%, 5%, 10%, 15% and 20% and the FP and FN values are averaged for the 5 subnetworks under size 100.

As the major focus of the work is to improve accuracy, threshold detection was carried out at 5 different values, at 2%, 5%, 10%, 15% and 20% of edge ranking. The AUROC and AUPR values were used to represent the threshold invariant evaluation. Fine-tuning the threshold for evaluating the network can improve the interpretability of the results.

Another limitation of the study is that the granger causality used for the study is pairwise linear, extending it to a nonlinear and multi-variable level can improve the results by discovering non-linear interactions and multiple gene-based regulations. Granger causality is not true causality, for example, if there is a mediator variable through which effects are transferred then such interaction will not be realized by the granger causality. As this approach uses the stationarity assumption to use granger causality, it will be a difficult condition to be satisfied by every gene expression dataset. More time points in the dataset can improve the accuracy of the results. Also extending this work to real datasets can provide more insights into how it fairs in the specific organism or disease-oriented datasets.

V. Conclusion

GRN inference is an important research area that holds great hopes for understanding the regular working of the human body and disease pathology. In this study, we took advantage of the DREAM4 gene expression dataset as a benchmark for comparing the inferred scores. The usage of a parallel pipeline with nonlinear regression and linear statistical prediction branches improves the performance of GRN construction to a considerable degree. For size 100 dataset the proposed method outperforms Greynet in all 5 networks. Future work focuses on extending this work to overcome the above limitations.

References

[1] G.H.F. Tam, C. Chang, and Y.S. Hung, August. “Application of Granger causality to gene regulatory network discovery”. In 2012 IEEE 6th International Conference on Systems Biology (ISB), pp. 232-239, 2012.

[2] C. E. Shannon, “A mathematical theory of communication”. The Bell system technical journal, 27(3), 379–423, 1948.

[3] G. Altay, and F. Emmert-Streib, “Revealing differences in gene network inference algorithms on the network level by ensemble methods”, Bioinformatics, 26,1738–1744, 2010.

[4] A. D. Wyner, “A definition of conditional mutual information for arbitrary ensembles”. Information and Control, 38(1), 51-59, 1978.

[5] S. Frenzel , & B.Pompe, “Partial mutual information for coupling analysis of multivariate time series”, Physical review letters, 99(20), 204101, 2007.

[6] A. J. Butte, P. Tamayo, D. Slonim, T. R. Golub, , and I. S. Kohane, “Discovering functional relationships between rna expression and chemotherapeutic susceptibility using relevance networks”, Proceedings of the National Academy of Sciences of the United States of America, 97(22),12182–12186. 2000.

[7] A. J. Butte, and I. S. Kohane, “Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements”, Biocomputing 2000, 7,418–429 1999.

[8] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Dalla Favera, and A. Califano, “Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context”, BMC bioinformatics, 7(1),1–15, 2006.

[9] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, , and T. S. Gardner, et al. “Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles”, PLoS biology, 5(1), e8. 2007.

[10] P. E. Meyer, K. Kontos, F. Lafitte, and G. Bontempi, “Information-theoretic inference of large transcriptional regulatory networks”, EURASIP journal on bioinformatics and systems biology, 2007(1), 1–9, 2007.

[11] L. F. Iglesias-Martinez, W. Kolch, and T. Santra, “Bgrmi: A method for inferring gene regulatory networks from time-course gene expression data and its application in breast cancer research”, Scientific Reports, 6(1), 1–12, 2016.

[12] Z. Narimani, H. Beigy, A. Ahmad, A. Masoudi-Nejad, and H. Fröhlich, “Expectation propagation for large scale bayesian inference of non-linear molecular networks from perturbation data”, PloS one, 12(2), e0171240, 2017.

[13] L. F. Iglesias-Martinez, B. De Kegel, and W. Kolch, “Kboost: a new method to infer gene regulatory networks from gene expression data”, Scientific Reports, 11(1),1–13. 2021.

[14] W. Liu, Y. Jiang, L. Peng, X. Sun, W. Gan, Q. Zhao, and H. Tang, “Inferring gene regulatory networks using the improved markov blanket discovery algorithm”, Interdisciplinary Sciences: Computational Life Sciences, 14(1), 168–181, 2022.

[15] H. Richards, Y. Wang, T. Si, H. Zhang, and H. Gong, “Intelligent learning and verification of biological networks”, In Advances in Artificial Intelligence, Computation, and Data Science, pages 3–28. Springer, 2021.

[16] A. C. Haury, F. Mordelet, P. Vera-Licona, and J. P. Vert, “Tigress: trustful inference of gene regulation using stability selection”, BMC systems biology, 6(1),1–17, 2012.

[17] X. Huang, and Z. Zi, “Inferring cellular regulatory networks with bayesian model averaging for linear regression (bmalr)”, Molecular BioSystems, 10(8), 2023–2030, 2014.

[18] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts, “Inferring regulatory networks from expression data using tree-based methods”, PloS one, 5(9), e12776, 2010.

[19] S. Park, J. M. Kim, W. Shin, S. W. Han, M. Jeon, H. J. Jang, and J. Kang, et al. “Btnet: boosted tree based gene regulatory network inference algorithm using time-course measurement data”, BMC systems biology, 12(2),69–77, 2018.

[20] Y. Zhang, X. Chang, and X. Liu, “Inference of gene regulatory networks using pseudo-time series data”, Bioinformatics, 37(16),2423–2431, 2021.

[21] V. A. Huynh-Thu, Machine learning-based featureranking: statistical interpretation and gene network inference, PhD thesis,Universit ́e de Li`ege, Li`ege, Belgium, 2012.

[22] D. Marbach, R. Prill, T. Schaffter, C. Mattiussi, D. Floreano, and G. Stolovitzky, “Revealing strengths and weaknesses of methods for gene network inference”, PNAS, 107(14),6286–6291, 2010.

[23] D. Marbach, T. Schaffter, C. Mattiussi, and D. Floreano, “Generating realistic in silico gene networks for performance assessment of reverse engineering methods”, Journal of Computational Biology, 16(2),229–239, 2009.

[24] R. Prill, D. Marbach, J. Saez-Rodriguez, P. Sorger, L. Alexopoulos, and X. Xue, et al. “Towards a rigorous assessment of systems biology models: the dream3 challenges”, PLoS ONE, 5(2), e9202, 2010.

[25] R. J. Hyndman, and G. Athanasopoulos, "Forecasting: Principles and Practice”, 2nd edn , OTexts, Melbourne, Australia, 2018. [Online]. Available: https://otexts.com/fpp2/ [Accessed Jan. 6, 2023].

[26] Y. Shin, and P. Schmidt, “The KPSS stationarity test as a unit root test”, Economics Letters, 38(4), 1992, pp.387-392.

[27] R.G. Brown, “Exponential smoothing for predicting demand”, cambridge, mass., arthur d. little. Book Exponential Smoothing for Predicting Demand, 1956.

[28] G. Chen, and Z. Liu. "Inferring causal gene regulatory network via GreyNet: From dynamic grey association to causation.", Frontiers in Bioengineering and Biotechnology, 2022.

[29] R. Zheng, M. Li, X. Chen, F. X. Wu, Y. Pan, and J. Wang, "BiXGBoost: a scalable, flexible boosting-based method for reconstructing gene regulatory networks.", Bioinformatics 35.11: 1893-1900, 2019.

[30] V. A. Huynh-Thu, and S. Guido, "Tree-based learning of regulatory network topologies and dynamics with Jump3.", Gene Regulatory Networks. Humana Press, New York, NY. 217-233, 2019.

A Precise IFCIoT Platform for the Achievement of Semantic Interoperability in Healthcare Data Systems

Sony P, Siva Shanmugam G, Sureshkumar N
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore, India
sony.p2016@vitstudent.ac.in, sivashanmugam.g@vit.ac.in, sureshmcame@gmail.com

***Abstract*—Fast decision support systems and accurate diagnosis have become significant in the rapidly growing healthcare sector. Quick and reliable retrieval of healthcare data becomes progressively more difficult as the number of heterogeneous medical IoT devices linked to the human body increases. Semantic interoperability is one of the fundamental prerequisites for the Healthcare Internet of Things(HIoT). The state of art solutions of HIoT suffers latency and bandwidth issues. Fog computing, a cloud computing extension, not only addresses the latency issue but also offers additional advantages, including resource mobility and on-demand scaling. Fog computing is included in the suggested method to reduce latency and network bandwidth usage in a system that offers semantic interoperability in healthcare organizations. The proposed framework can reduce the computation overhead of heterogeneous healthcare data. The simulation results show that fog computing can reduce delay, network usage, and energy consumption.**

***Index Terms*—Fog Computing, Healthcare IoT, Semantic interoperability, ontology, umls**

I. Introduction

In the healthcare sector, semantic interoperability refers to a healthcare system's ability to read and process medical data generated or controlled by another healthcare system. In the Internet of Things age, healthcare data must be monitored in real-time, relevant decisions made, and alerts issued in seconds. Furthermore, the data generated by IoT devices is massive, primarily in big data. Medical data from IoT devices and unstructured electronic health records involve complex processing to provide timely health warnings. Developing all the solutions in the cloud causes a lot of network congestion and computational delays.

A patient requires real-time monitoring when he receives an infusion, necessary treatments, or surgery or is in an intensive care unit or ambulatory services. Real-time monitoring may occasionally save lives in critical situations such as cardiac emergencies, strokes, and so on. Real-time monitoring in healthcare necessitates an immediate response. When data is collected from IoT devices and delivered to the cloud for processing, the real-time monitoring system is hampered by its inability to receive a timely response from the cloud. In most circumstances, IoT Healthcare devices do not have enough computing power and Fog Computing technology was conceived to meet these requirements.

Fog computing is used in many sectors, including agriculture, healthcare, logistics, transportation, etc. Fog computing methodology can be used in healthcare for providing health monitoring [1] and real-time notifications and can play an important role in decision support systems [2]. In healthcare, fog computing can be used to monitor conditions such as high blood pressure [3], arthritis [4], diabetes [2], and many more. It is an excellent resource for geriatric care [5].

In healthcare, like disease monitoring, some other applications require prompt results. Semantic interoperability solution is one among them. When medical records are transported from one healthcare environment to another, such as when a patient is unconscious or when a patient is transferred from one hospital to another for rapid treatment or surgery, the semantic interoperability solution must be absolutely time critical.

A. Problem Definition

We all live in a world that demands quick and precise processing. The proposed goal is to create a rapid and accurate semantic interoperability solution for healthcare data systems. The proposed solution should address the following specific objectives.

- The suggested solution must handle all lexical, syntactic, and semantic challenges found in medical documents.
- The recommended approach should minimize bandwidth consumption and delay.
- The solution should consider all types of medical documents, including unstructured, structured, and IoT MD documents.

The following is how the rest of the article is organized. Section II contains a review of the literature. The suggested architecture is represented in Section III. The implementation details are presented in Section IV. Section V displays the findings and discussions. Section VI contains the conclusion and future scope.

II. Literature Review

Even before 2010, several researchers pointed out that cloud computing was vulnerable to latency and bandwidth difficulties. At an ACM conference in 2014, CISCO presented Fog computing [6] to the world.

To overcome latency and bandwidth issues, Malik et al. [7] proposed a three-layer fog computing architecture. The bottom layer was the Device layer, the middle layer was Fog, and the upper layer was the Cloud layer. When the number of sensing devices increases considerably, problems such as the processing time of time-critical IoT applications will increase due to network congestion caused by offloading data to the cloud and uploading data from many IoT generators, according to [7].

Some authors have proposed a Fog computing-based paradigm to diagnose diseases. Sood et al. [3] proposed Fog based healthcare framework to identify and monitor hypertension attacks in human beings, and they used an artificial neural network to predict hypertension.

Internet of Things and Fog Computing presented a real-time deployment of an E-health system for monitoring the health of older people [8]. They used the My signals HW V2 platform and an Android app as a fog server to collect pharmacological and health parameters regularly. The parameters were collected, evaluated, and cached before being sent to the cloud via a specific REST API using the JSON data model. Data distribution, communication, and management layers are the three layers in the suggested design. The data distribution layer makes use of MongoDB [8].

A decision support system for healthcare IoT was created using soft computing and fog computing principles. The six levels of the Fog computing architecture in the suggested research [2] were the physical and virtualization layer, monitoring layer, pre-processing layer, temporary storage layer, security layer, and transport layer.

Ahmed et. al [9] has suggested another fog computing-based healthcare monitoring system. The suggested system consisted of three layers: a sensor network, a fog, and a cloud layer. The sensor network layer contains a variety of sensors, such as E.C.G. sensors, B.P. sensors, and temperature sensors. The fog layer handled data security, compression, notification service, data analysis, and local storage. The cloud layer handled big data processing, ample data storage, and disease prediction. One of the suggested technique's primary drawbacks was that it only applied to images.

For handling diabetic healthcare IoT data, David et al. [10] compare cloud computing with edge/fog computing environments. The bottom layer comprises diabetic instruments with sensors, while the intermediate layer includes intelligent devices, hubs, routers, and gateways. The top layer, the Cloud layer, contains all computing components.

Hassen et al. [5] created a real-time home hospitalization system using the healthcare and environmental parameters of five persons aged 56 to 61. The data was gathered using My signal hw V2 in their proposed system, while android phones and tablets used fog servers. The system's main drawback was that the dataset's size was too small.

Several authors, Markus A et.al [11],Ashouri et al. [12], Neha R K et.al [13], Mohan et al. [14] to mention a few, conducted various analyses on the quality of simulators. They claimed iFogSim was one of the best simulators accessible due to its event-driven and open-source nature and implementation language. According to Markus et.al [11], 63% of simulators are event-driven and 70% of them are written in the language java.

According to [12], iFogSim was the optimal tool for modeling environments that demand faster response durations, higher processing usage, and lower bandwidth and energy consumption.

III. Proposed Architecture

Four layers comprise a layered Fog-based semantic interoperability approach in the Healthcare IoT, as shown in Fig.1. In this architecture, physical layer devices such as wearable sensors, implanted sensors, wireless and wired IoT devices, environmental sensors, sensors attached to the human body, and other intelligent healthcare IoT devices can transmit sensed medical data to edge nodes along with their geospatial locations. The physical layer includes unstructured EHR also.

The layer nearer to the physical layer is called the edge layer. The edge layer often uses resources like mobile phones, tablets, laptops, workstations, etc., as computing facilities. The two computing modules in the edge layer are modules for data granularity and modules for ontology integration. The medical IoT data received from intelligent devices require extra processing with the help of the UMLS ontology [15] for storing and indexing the IoT medical data. The Data Granularity module is in charge of processing IoT medical data.

The processed data in the edge layer is transferred to the fog layer, comprised of fog nodes designated as fog gateways. These gateways served as a router, switches, and other computational services, as well as providing storage. Some fog gateways have been renamed Fog proxy, providing communication with the cloud layer's third layer. The sensors on the human body collect data from individuals, which may be used to analyze and diagnose various illnesses. After being transformed into an intermediate format comprehensible by all healthcare professionals, it can be readily translated to the target formats, where the processed data is stored in the cloud. In this case, the fog node serves as a link between the cloud and the end devices.

Smartphones, tablets, or desktops PC can act as fog gateways. The computational and storage capacities of these devices are limited. The sensed data from the physical layer is transmitted to the cloud for additional processing and calculation after the initial essential computation. Every hospital has one or more fog nodes, which send the analyzed data to the cloud. If the caregiver needs access to previous data, the fog node may get it from the cloud and make it available. Suppose a caregiver needs medical information from another hospital.

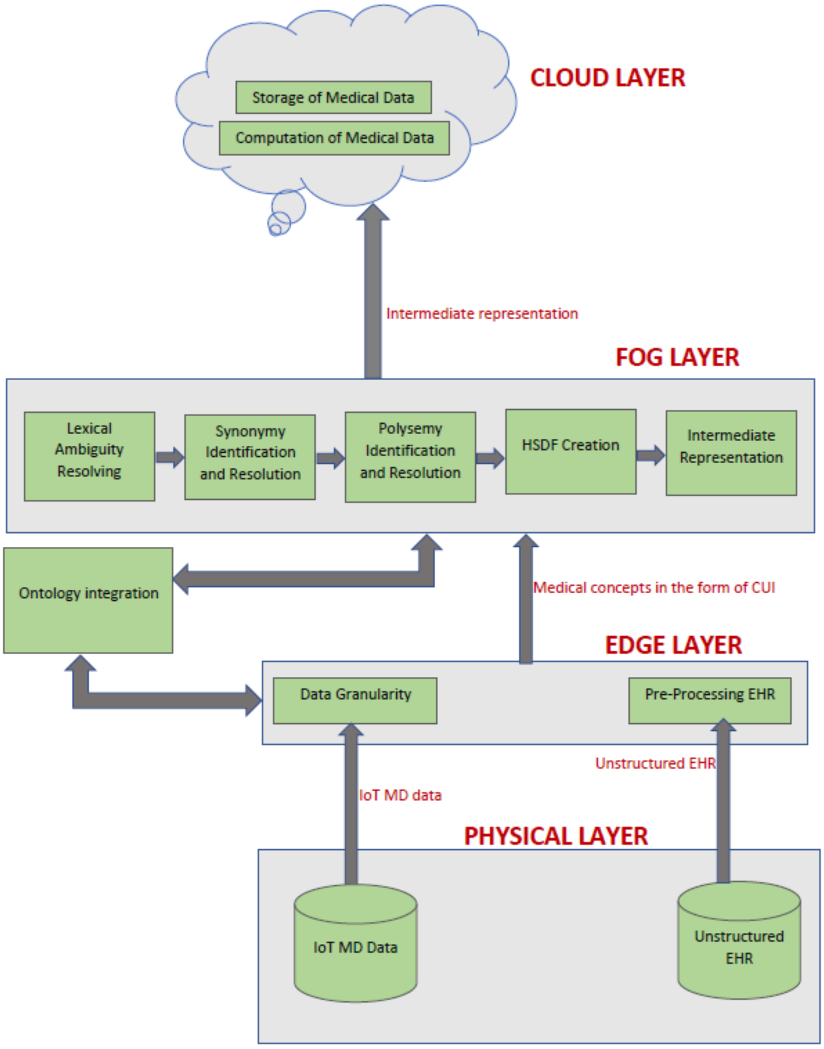

CLOUD LAYER
Storage of Medical Data
Computation of Medical Data
Intermediate representation
FOG LAYER
Lexical Ambiguity Resolving
Synonymy Identification and Resolution
Polysemy Identification and Resolution
HSDF Creation
Intermediate Representation
Ontology integration
Medical concepts in the form of CUI
EDGE LAYER
Data Granularity
Pre-Processing EHR
Unstructured EHR
IoT MD data
PHYSICAL LAYER
IoT MD Data
Unstructured EHR

Fig. 1. Fog-based Integrated HIoT Architecture.

TABLE I
SAMPLE IoTMD DATA

Age 12 yrs	Allergies Vancomycin			
28-04-2021	7.30 A.M	8 A.M	8.30 A.M	9 A.M
Fentanyl 15mcg	40			
Midazolan 0.9mg		0.7mg		
Paracetamol 0.5mg			0.5mg	
O2				
SPO2			98	96
EtCO2			4	3.8
Temperature(F)	99.3	99.3	100.3	
Pulse	80	80	80	93

In that case, the fog node can retrieve the intermediate form from the cloud and send it to the caregiver, who can decode it into the necessary format. All data packets received are not transmitted to the cloud computing module; only applications with time flexibility are routed to the top levels.

The cloud can translate medical data collected from numerous fog nodes into an intermediate format to maintain interoperability. It may store the processed data for a more extended period. Latency can be decreased because the initial processing is done on the fog node.

IV. IMPLEMENTATION

The proposed system is implemented using the simulator ifogsim2.0 [16] and the National Library of Medicine's Ontology UMLS [17].

Fig.2 depicts a sample simulation result. The physical devices read three types of input data: IoTMD data, EHR data, and a combination of the two. A sample IoTMD data and IoTMD matrix is shown in Table I and II respectively. In iFogsim, sensors serve as input devices. The input data is first sent to physical layer devices for pre-processing. The

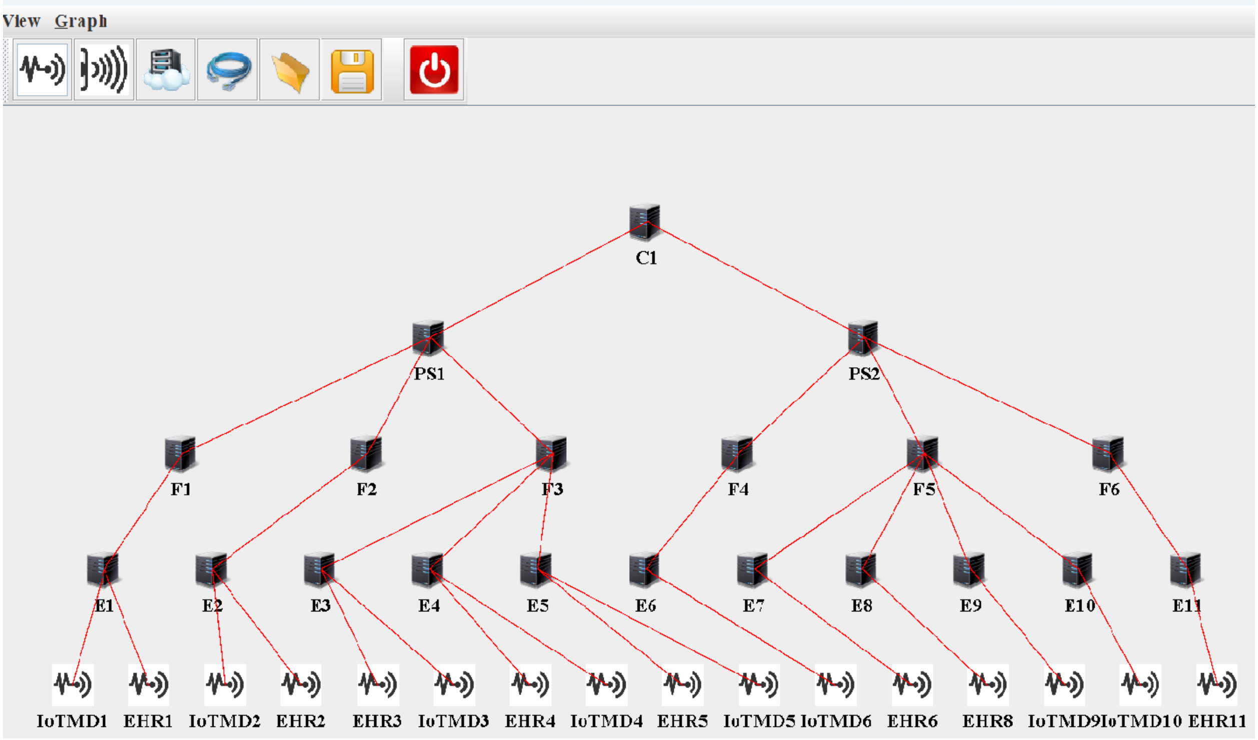


Fig. 2. A Sample Simulation Result.

pre-processed data is subsequently sent to the fog devices containing the semantic interoperability module. The fog devices generate an intermediate form (Healthcare Sign Description Framework), which is subsequently sent to the cloud devices to be stored. Sensors and actuators serve as physical devices in iFogsim. Fog devices include cloud devices, proxy servers, fog devices, and edge devices. Different levels are utilized to identify these devices. Level-1 fog devices are renamed cloud devices, level- 2 fog devices are referred to as proxy servers, level-3 fog devices are referred to as fog devices themselves, and level-4 fog devices are referred to as edge devices. In iFogsim, any number of physical and fog devices are possible.

iFogsim also has the ability to add an eternal database to the system. The domain ontology UMLS is used as an external database in this case. The modules of edge and fog devices process using the UMLS ontology. The external databases can be created as events in the controller class or in the main function. Using the main function, we can also store all of the data obtained after the simulation in the database. Here we created an external database UMLS in the application's main function.

The medical concepts are identified by the concept processing algorithm represented in [18]. After resolving the linguistic issues, HSDF(Health Sign Description Framework) creation microservice is called. HSDF creation algorithms specified in the article [19] are used to create healthcare signs.

A. Configuration Environment

1) Duration of the experiment: 35000seconds

TABLE II
IoT MD MATRIX GENERATED FROM THE AUTOMATA

UHID	TimeStamp	item	Value
253657	28/04/2021 7.30 A.M	Temp	99.3
253657	28/04/2021 8.30 A.M	Temp	100.3
253657	28/04/2021 7.30 A.M	Pulse	80
253657	28/04/2021 9 A.M	Pulse	93
247589	28/04/2021 7.30 A.M	Temp	102
357426	28/04/2021 7.30 A.M	Temp	100.4
253657	28/04/2021 7.30 A.M	Food intake Idly	20gm
247589	28/04/2021 8.30 A.M	Urine Output	0.5litre

TABLE III
CONFIGURATION PARAMETERS.

Parameters	Cloud	Proxy	Fog	Edge
Count	8	4	16	52
Speed(in MIPS)	2000-2500	2000-2500	2000-2500	400
RAM(GB)	16	16	8	2
Uplink(Mbps)	80	20	80	150
Downlink(Mbps)	80	40	130	200
Busy Power(MW)	106.223	106.223	106.223	67.56
Idle Power(MW)	82.34	82.34	82.34	61.78

2) No of location change events: 98
3) Network architecture: heterogeneous
4) Deployment: Grid and Uniform
5) No of patients in one cluster: 10
6) No of biosensors in one cluster: 48

An Intel Core 2 Duo CPU operating at 2.33 GHz, 16 GB of

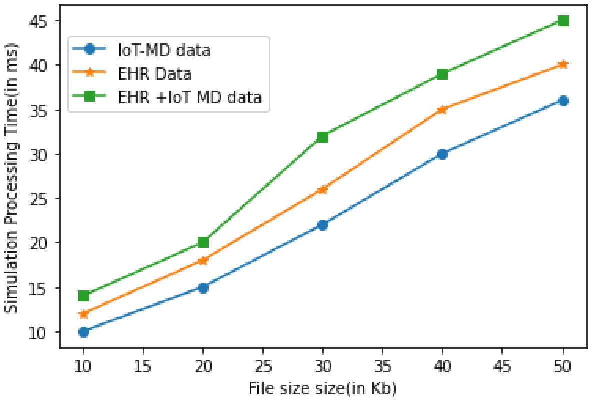


Fig. 3. Simulation Time

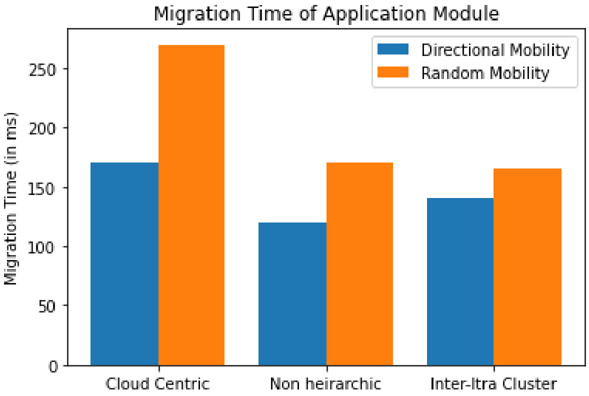


Fig. 4. Migration Time of Application Module.

RAM, and iFogSim 2.0 are used to execute the simulations. Table III provides the configuration parameters. Healthcare sensors can operate in three modes: sustained, sleep-awake, and periodic. When a patient is in a critical care unit, in an emergency, or has surgeries or procedures, the patient's operating mode is set to *Sustained*. *Sleep Awake mode* is set when the person is in a normal situation. Sensors are set to *Periodic mode* when health data is required on demand.

For the model simulation, three separate datasets are used. The first is structured EHR data, the second is data from IoT medical devices, and the third combines both. IoT MD data set is generated from the patients residing in 25 different states of India. For each user, we create smartphones as a gateway. Through tier-1 nodes in the Fog layer, the gateway may connect with tier-0 nodes in the cloud layer. We use the fractional selectivity of the input-output relationship with a module set to 1.0 [16]. Cloud-centric, non-hierarchical, and intra/inter-cluster migrations are the migration methodologies employed.

The Clustering class implemented in IFogSim can be used to cluster Fog nodes in order to improve storage and computing capacity. The IoT MD data set comprises the locations of each Fog node; locations are parsed using the data parser class. Each node can access information about its neighboring nodes' communication range, storage and computing capacity, and bandwidth.

V. Results and Discussion

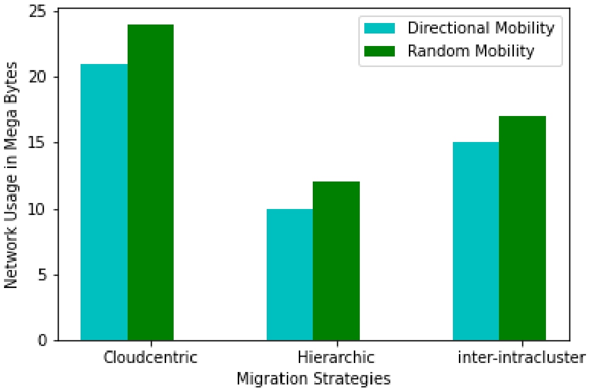


Fig. 5. Network usage during migration.

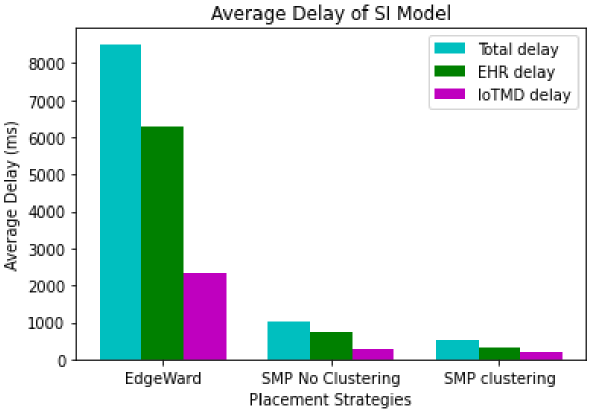


Fig. 6. Average Delay.

We assess simulation time, migration time of application modules, network usage during migration, and average delay to evaluate the performance of the proposed system. Simulation time is the period between the submission of the document to the creation of the intermediate form. Simulation time is shown in Fig.3. Simulation time is less for the documents that contain data "*only from IoT Medical devices*," while simulation time for documents that contain *only EHR data* is much more. It is maximum for the documents that contain both.

We use migration time and network usage to assess the performance of mobility schemes in different clustering environments. In all three methods (cloud-centric, non-hierarchic, and inter-cluster), the random mobility scheme has higher migration time and network usage than the directional mobility scheme. The migration time of microservice modules using random and directional mobility strategies is illustrated in Fig.4.

Network bandwidth usage during random and directional microservice migration is displayed in Fig.5. During the cloud migration, numerous fog nodes and gateways are engaged in the cloud-centric strategy, consuming more network bandwidth. As the number of intermediary Fog nodes during the migration process is reduced, the intra and inter-cluster strategy saves network utilization. The non-hierarchical approach has fewer network resources than the other two techniques. Random mobility scheme has more network bandwidth than directional mobility in all three systems. The network load increases as the number of healthcare IoT devices linked to the human body grows. The increased usage of networks for cloud environments can cause network congestion and application performance deterioration. The use of fog nodes can minimize network traffic. If fog computing is implemented, a similar issue can be averted.

In iFogSim2, three different placement algorithms are used: Edgeward, SMP No Clustering, and SMP Clustering. The Edgeward placement technique solely addresses vertical placement, ignoring horizontal scalability and clustering among Fog nodes, and SMP No clustering uses horizontal scalability instead of clustering the fog nodes. The SMP clustering technique uses iFogsim2's microservice orchestration and clustering functionalities. Fig.6 demonstrates that the SMP-clustering placement approach has the shortest average latency.

VI. Conclusion and Future Scope

Healthcare data is available in various forms, languages, and representations depending on the organization. Semantic interoperability is one of the most significant research issues in the healthcare business. As medical data is so large, most hospitals store it in the cloud. However, network congestion and overall performance delay are present in a cloud environment.

This article proposes a semantic interoperability solution for healthcare IoT in the fog environment. The proposed system uses structured and unstructured medical documents as input, and the proposed method uses an ontology *UMLS* developed by the National Library of Medicine(NLM). As a result of its observations, the proposed fog-based semantic interoperability model reduces latency and network congestion.

We intend to adapt the SI model to the dew computing paradigm in the future, as dew computing is better suited to real-time applications with minimal latency.

References

[1] W.-H. H. H. C. S. S. R. Anand Paul, Hameed Pinjari, "Fog computing-based iot for health monitoring system," *Journal of Systems and Software*, vol. 2018, pp. 1687–725X, 2018. [Online]. Available: https://doi.org/10.1155/2018/1386470

[2] M. Abdel Basset, G. Manogaran, A. Gamal, and V. Chang, "A novel intelligent medical decision support model based on soft computing and iot," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4160–4170, 2020.

[3] S. K. Sood and I. Mahajan, "Iot-fog-based healthcare framework to identify and control hypertension attack," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1920–1927, 2019.

[4] S. Tanwar, J. Vora, S. Kaneriya, S. Tyagi, N. Kumar, V. Sharma, and I. You, "Human arthritis analysis in fog computing environment using bayesian network classifier and thread protocol," *IEEE Consumer Electronics Magazine*, vol. 9, no. 1, pp. 88–94, 2020.

[5] H. Ben Hassen, N. Ayari, and B. Hamdi, "A home hospitalization system based on the internet of things, fog computing and cloud computing," *Informatics in Medicine Unlocked*, vol. 20, p. 100368, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352914820302860

[6] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM computer communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

[7] S. Malik and K. Gupta, "Smart city: A new phase of sustainable development using fog computing and IoT," *IOP Conference Series: Materials Science and Engineering*, vol. 1022, no. 1, p. 012093, jan 2021. [Online]. Available: https://doi.org/10.1088/1757-899x/1022/1/012093

[8] D. W. . H. B. Ben Hassen, H., "An e-health system for monitoring elderly health based on internet of things and fog computing," *Health Inf Sci Syst*, vol. 7, no. 24, 2019.

[9] A. I. T. A. A. Ahmed Elhadad, Fulayjan Alanazi, "Fog computing service in the healthcare monitoring system for managing the real-time notification," *Journal of Healthcare Engineering*, vol. 2022, pp. 2040–2295, 2022.

[10] D. C. Klonoff, "Fog computing and edge computing architectures for processing data from diabetes devices connected to the medical internet of things," *Journal of Diabetes Science and Technology*, vol. 11, no. 4, pp. 647–652, 2017, pMID: 28745086. [Online]. Available: https://doi.org/10.1177/1932296817717007

[11] A. Markus and A. Kertesz, "A survey and taxonomy of simulation environments modelling fog computing," *Simulation Modelling Practice and Theory*, vol. 101, p. 102042, 2020, modeling and Simulation of Fog Computing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X1930173X

[12] M. Ashouri, F. Lorig, P. Davidsson, and R. Spalazzese, "Edge computing simulators for iot system design: An analysis of qualities and metrics," *Future Internet*, vol. 11, no. 11, p. 235, 2019.

[13] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47 980–48 009, 2018.

[14] N. Mohan and J. Kangasharju, "Edge-fog cloud: A distributed cloud for internet of things computations," *2016 Cloudification of the Internet of Things (CIoT)*, pp. 1–6, 2016.

[15] O. Bodenreider, "The unified medical language system (umls): integrating biomedical terminology," *Nucleic acids research*, vol. 32, no. suppl_1, pp. D267–D270, 2004.

[16] R. Mahmud, S. Pallewatta, M. Goudarzi, and R. Buyya, "ifogsim2: An extended ifogsim simulator for mobility, clustering, and microservice management in edge and fog computing environments," *Journal of Systems and Software*, vol. 190, p. 111351, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121222000863

[17] O. Bodenreider, "The unified medical language system (umls): integrating biomedical terminology," *Nucleic acids research*, vol. 32, no. suppl_1, pp. D267–D270, 2004.

[18] P. Sony and N. Sureshkumar, "Concept-based electronic health record retrieval system in healthcare iot," in *Cognitive Informatics and Soft Computing*. Springer, 2019, pp. 175–188.

[19] P. Sony and S. Nagarajan, "Semantic interoperability model in healthcare internet of things using healthcare sign description framework," *International Arab Journal of Information technology*, vol. 19, p. 111351, 2022. [Online]. Available: https://iajit.org/portal/index.php/archive/volume-19-2022/july-2022-no-4

Exploring the potential of Ensemble Based Machine Learning Techniques in Polycystic Ovary Syndrome Diagnosis

Mohammed Usman F[1] Afrah Ayub[2], Thirumala Akash K[3], Nitesh Kumar T[4] and Mohammed Riyaz Ahmed[5]
[1,2]School of Computing and Information Technology, *REVA University*, Bengaluru, India
[3,4,5]School of Multidisciplinary studies, *REVA University*, Bengaluru, India
farooqmohammedusman@gmail.com, afrahayub1907@gmail.com, akashthirumala@gmail.com,
iam.nitesh.t@gmail.com and riyaz@reva.edu.in

***Abstract*—Polycystic ovary syndrome (PCOS) is a common hormonal disorder that impacts women during the reproductive period. The accurate diagnosis of PCOS is challenging due to its wide spectrum of presentation. Machine learning has the potential to play a significant role in the early detection and diagnosis of PCOS. In this paper, we present a machine-learning approach for the early detection of PCOS. We used demographic data to train a machine-learning model and the model was able to achieve high accuracy in detecting PCOS. This can help in identifying the subtle changes in hormones and other biomarkers that are indicative of PCOS. We also discussed the importance of data scaling and normalization in machine learning to improve the model's performance. Our results show that machine learning can be used as an effective tool for the early detection of PCOS and can aid in accurately diagnosing this disorder. However, it will exponentially increase the effective medication for diagnosis of PCOS. Additionally, machine learning should be used as an adjunct to clinical examination, hormonal assays, and ultrasound.**

***Index Terms*—Demographic data, Harmonal Disorder, Machine Learning, Normalization, Polycystic ovary syndrome**

I. Introduction

PCOS is a common hormonal disorder that primarily affects women of reproductive age. Early detection of PCOS is important because it can lead to early treatment and management of the condition. PCOS symptoms include irregular menstruation, excessive hair growth, acne, and obesity. PCOS can also be diagnosed through physical examination, such as pelvic ultrasound and blood tests measuring hormone levels [1]. A definitive diagnosis of PCOS is usually made when at least two criteria are met: irregular periods, clinical or biochemical signs of hyperandrogenism, and polycystic ovaries. If you suspect you have PCOS, you must visit a healthcare provider for proper diagnosis and treatment.

Machine learning is a relatively new technology that has been applied to the early detection of polycystic ovary syndrome (PCOS). It is a type of artificial intelligence that enables computers to learn and predict based on data. Machine learning algorithms can be trained on large datasets of patient information, such as ultrasound images, hormone levels, and demographic data, in the context of PCOS to identify patterns and predict a patient's likelihood of having PCOS Fig.1.

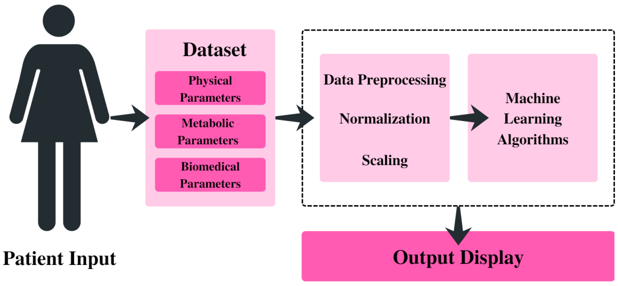


Fig. 1: Block Diagram representation of Polycystic Ovary Syndrome Diagnosis using Machine Learning

It could aid in the early detection of PCOS [2]. Several studies have been conducted in recent years to develop machine-learning models to detect PCOS early [2].

These studies have used various techniques such as normalization, scaling, correlation, feature extraction, and classification to improve the accuracy of PCOS diagnosis. Scaling and normalization are techniques used in machine learning to preprocess data and ensure that input values are within a consistent range. These techniques can improve the performance of machine learning models by making the input data more consistent and easier for the model to process [3]. For example, one study used a deep learning algorithm to analyze ultrasound images of the ovaries and achieved high accuracy in detecting PCOS. Another study used demographic data and hormone levels to predict PCOS with an accuracy of over 90% [4]. However, it's important to note that these studies are still in the early stages, and more research is needed to validate the use of machine learning in the early detection of PCOS in clinical practice. Additionally, machine learning is not a substitute for traditional diagnostic methods and should be used as an adjunct to clinical examination, hormonal assays, and ultrasound.

In the context of Polycystic ovary syndrome (PCOS), correlation can be used to identify relationships between different variables and their impact on the development and progression of the disorder. By identifying these correlations, researchers

TABLE I: Literature Survey

Authors	Techniques Used	Summary	Limitations
Bhat, S. A. (2021) [5]	J48, Random Forest, and Bagging	In this study, decision tree algorithms such as J48, Random Forest, and Bagging were used to predict PCOS. The authors found that Random Forest performed best in terms of accuracy and sensitivity.	Many machine learning studies in the field of medicine suffer from small sample sizes, which can impact the generalizability of the results.
Nabi, N. (2021) [6]	K-nearest neighbors, decision tree, support vector machine, and artificial neural networks.	In this study, several machine learning algorithms such as K-nearest neighbors, decision tree, support vector machine, and artificial neural networks were compared to predict PCOS. The authors found that artificial neural networks performed best in terms of accuracy.	The machine learning model have performed well on the training data but poorly on the test data, indicating overfitting to the training data.
Ahmetaevi, A. (2022) [7]	Artificial neural networks	In this study, an artificial neural network (ANN) was trained and tested on a dataset of 517 patients to predict PCOS. The authors found that the ANN had good accuracy for PCOS prediction.	Only ANNs were considered and no comparison with other algorithms.
Danaei Mehr, H. (2022) [8]	Hybrid machine learning approach (decision tree and support vector machine algorithms).	In this study, a hybrid machine learning approach was developed using a combination of decision tree and support vector machine algorithms to predict PCOS. The authors found that the hybrid approach performed better compared to individual algorithms.	The sample may not be representative of the population as a whole, leading to biased results.
Aggarwal, S.. (2023) [9]	Support vector machine and decision tree algorithms.	In this study, both support vector machine (SVM) and decision tree algorithms were used to predict PCOS. The authors found that SVM performed better compared to decision tree algorithms.	Small sample size and limited algorithms considered.

and healthcare providers can better understand the underlying causes of PCOS and develop more effective methods for predicting, diagnosing, and treating the disorder. Correlation is a statistical measure that describes the relationship between two or more variables in a dataset Fig.2. However, it is worth noting that correlation does not imply causality, and further research is needed to establish the causality of these relationships [10].

Ensemble-based machine learning techniques, such as Random Forest and Extra Trees classifiers, have been used to predict Polycystic ovary syndrome (PCOS). These techniques involve combining multiple decision trees to improve the accuracy and robustness of the model. Random Forest is an ensemble technique that builds multiple decision trees and combines them to make a final prediction. It works by selecting a random subset of features for each tree and then averaging the predictions of all the trees in the forest [11]. This technique can reduce the overfitting of individual decision trees and improve the model's overall accuracy. The extra Trees classifier is similar to Random Forest, but it uses even more randomization to select features for each tree. As a result, it can lead to more diverse decision trees and a more robust model overall.

II. Background

A common endocrine condition that affects women of reproductive age is polycystic ovary syndrome (PCOS). It is characterized by symptoms, including irregular menstrual cycles, hyperandrogenism (elevated levels of male hormones), and multiple small cysts on the ovaries.

A. History behind Polycystic Ovary Syndrome

Among women of reproductive age, polycystic ovary syndrome (PCOS) is a typical endocrine condition. The first description of a condition resembling PCOS was made in the early 1700s by an Italian physician, Antonio Vallisneri, who observed enlarged ovaries with multiple small cysts in women with infertility. In the early 1900s, German gynecologist Karl von Rokitansky described a similar condition: ”polycystic ovary disease.” In the 1930s, American endocrinologist Irving Stein and gynecologist Michael Leventhal coined the term ”polycystic ovary syndrome” to describe the complex of symptoms, including irregular periods, excess androgen (male hormone) production, and the presence of multiple small cysts on the ovaries [12]. However, it wasn't until the 1990s that PCOS was widely recognized as a common disorder, and research on the condition has continued to this day.

Polycystic ovary syndrome (PCOS) prediction using machine learning has been an active area of research in recent years. A few studies are worth mentioning :

Zhang et.al (2021) aims to use Raman spectroscopy and machine learning algorithms to screen polycystic ovary syndrome (PCOS). The authors performed Raman spectroscopy on both follicular fluid and plasma samples and used machine learning algorithms to analyze the resulting spectral data. The algorithms were trained on samples from women with and without PCOS to identify features associated with the disease. Their approach can potentially serve as a non-invasive method for PCOS screening. The study by Zhang et al. (2021) is a preliminary study, and further research is needed to validate the results and assess the clinical utility of this approach [13]. Silva et al. (2022) aim to identify new phenotypes of polycystic ovary syndrome (PCOS) using machine learning models and to evaluate the relationship between these phenotypes and clinical and laboratory variables. The authors used machine learning algorithms to analyze data from a cohort of women with PCOS, including demographic, clinical, and laboratory variables, to identify subgroups or phenotypes within the population. This study is a preliminary exploration of the use of machine learning algorithms in the study of PCOS, and larger, more comprehensive studies will be needed to validate

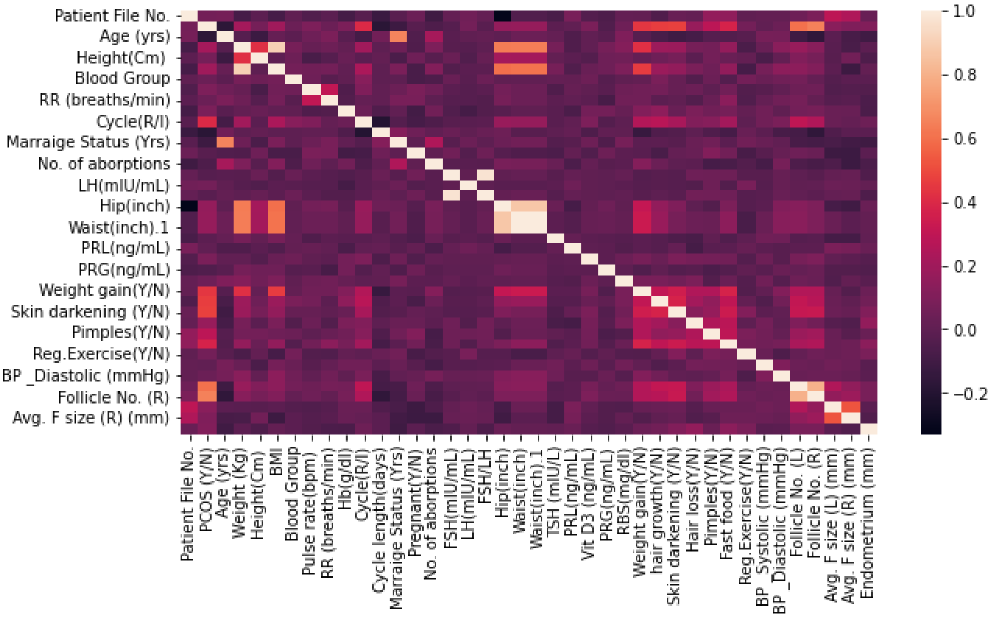


Fig. 2: Demonstrating the relationship between the attributes: Correlation Matrix

the results and determine the clinical utility of this approach [14]. Bharati et.al (2020) presents a study using machine learning algorithms to diagnose polycystic ovary syndrome (PCOS). The authors collected data on demographic, clinical, and laboratory variables from a cohort of women and used machine learning algorithms to develop a model for the diagnosis of PCOS. This study is a preliminary exploration of the use of machine learning algorithms in diagnosing PCOS, and larger, more comprehensive studies will be needed to validate the results and determine the clinical utility of this approach [15].

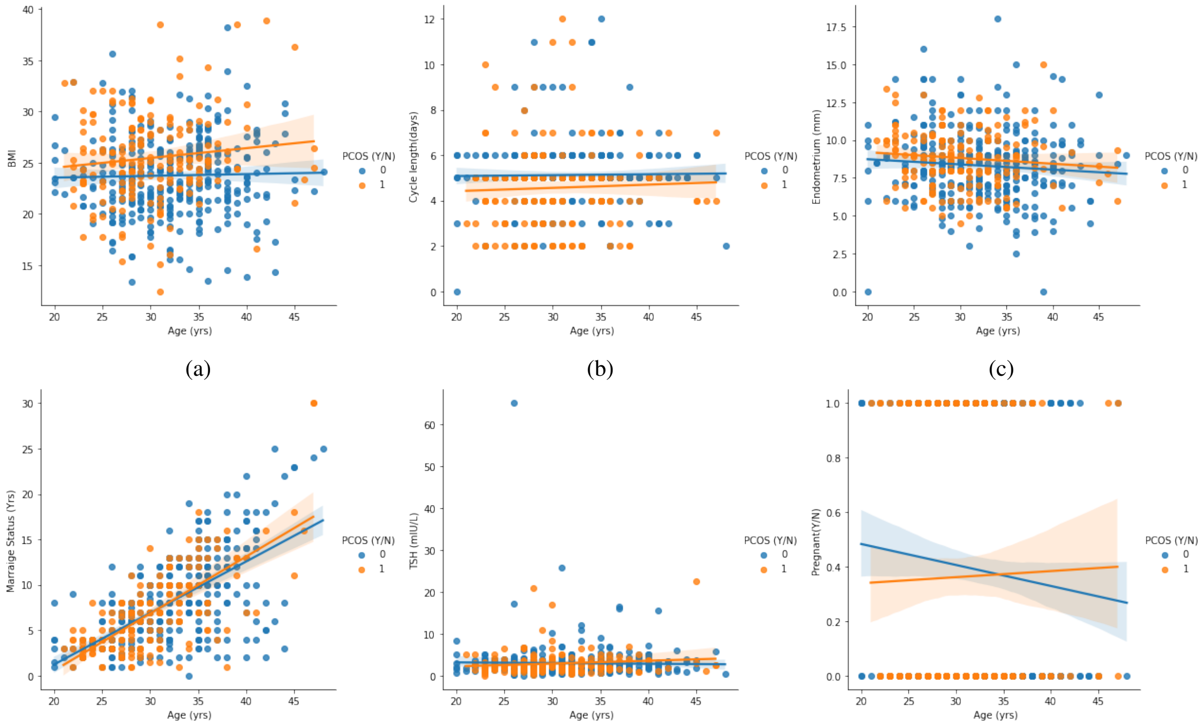
These studies suggest that machine learning can be a useful tool for predicting PCOS. However, more research is needed to confirm the results and to explore other algorithms and larger sample sizes.

B. *Types of Polycystic Ovary Syndrome*

Polycystic ovary syndrome (PCOS) is a complex disorder that affects the ovaries and can present with a wide range of symptoms. There are several types of PCOS, and it's important to note that not all women with PCOS will have the same symptoms or test results. 1. Classic PCOS: This type of PCOS is characterized by multiple cysts on the ovaries, irregular periods, and high levels of androgens (male hormones) such as testosterone. 2. Ovarian Androgen Excess: This type of PCOS is characterized by high levels of androgens, with or without multiple cysts on the ovaries [16]. 3. Insulin-Resistant PCOS: This type of PCOS is characterized by insulin resistance and obesity, with or without multiple cysts on the ovaries. 4. Inflammatory PCOS: This type of PCOS is characterized by an increased inflammation marker in the blood and is associated with a higher risk of metabolic complications such as diabetes and cardiovascular disease. 5. Hidden PCOS: This type of PCOS is characterized by PCOS-related symptoms but with normal ovaries on ultrasound. It's important to note that some women may have a combination of these types of PCOS, and the diagnosis of PCOS often depends on the presence of certain symptoms, physical examination, hormonal levels, and ultrasound findings [17].

C. *Causes for Polycystic Ovary Syndrome*

The exact cause of polycystic ovary syndrome (PCOS) is not fully understood. However, it is thought to be a combination of genetic, hormonal, and environmental factors [8]. Some proposed causes of PCOS include 1. Insulin resistance: Many women with PCOS have high levels of insulin, a hormone that regulates blood sugar. High insulin levels can lead to increased androgen (male hormones) production by the ovaries, which can contribute to developing PCOS symptoms [18]. 2. Hormonal imbalances: PCOS is characterized by an imbalance in the levels of certain hormones, such as luteinizing hormone (LH) and follicle-stimulating hormone (FSH). It can lead to ovulation problems and the formation of cysts on the ovaries. 3. Genetics: PCOS tends to run in families, which suggests that there may be a genetic component to the disorder. Studies have identified several genes that may be associated with PCOS. 4. Environmental factors: Environmental factors such as obesity, poor diet, and lack of physical activity can also play a role in developing PCOS. It's important to note that not all women with PCOS will have all of these risk factors. Additionally, some women may develop PCOS without any known cause [19].



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 3: a)Pattern of weight gain (BMI) over years in PCOS and Normal , b)Length of menstrual phase in PCOS vs normal, c)Pattern of endometrium over years in PCOS and Normal, d) Pattern of marriage status over years in PCOS and Normal, e)Pattern of TSH over years in PCOS and Normal , f)Pattern of Pregnancy over years in PCOS and Normal

D. *Symptoms of Polycystic Ovary Syndrome*

Polycystic ovary syndrome (PCOS) is characterized by a combination of symptoms that can vary from person to person as shown from the analysis in Fig.3. However, some of the most common symptoms of PCOS include the following: 1. Irregular periods: Women with PCOS may have infrequent, irregular, or heavy periods due to hormonal imbalances. 2. Excess hair growth: PCOS can cause an increase in the production of male hormones, which can lead to excess hair growth on the face, chest, back, and other body parts. 3. Acne: Hormonal changes in PCOS can also lead to acne and other skin problems. 4. Obesity: Obesity and weight gain are common in women with PCOS due to the hormonal imbalances and insulin resistance associated with the condition. 5. Infertility: PCOS can affect ovulation, making it difficult for women to conceive. 6. Headaches: women with PCOS can experience headaches as a symptom. 7. Depression and anxiety: women with PCOS can experience depression and anxiety as symptoms [20]. It is worth noting that some women may have mild symptoms or no symptoms at all. Therefore, it's important to consult a healthcare provider if you suspect you have PCOS.

III. Experimental Set up

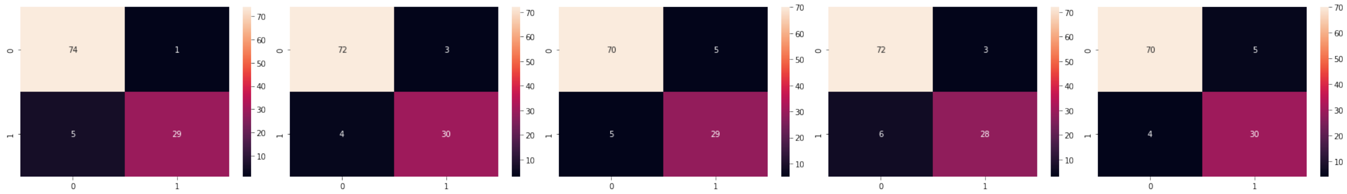
A. *Data Collection*

The data for predicting PCOS (polycystic ovary syndrome) using ensemble-based methods, is collected across 10 different hospitals from Kerala,India on various physiological and demographic characteristics of patients diagnosed with the condition. It includes hormone levels, body mass index, age, and family medical history. The experiment also needs a control group of patients without PCOS to compare the data. The data collected is standardized in a systematic way to ensure that it is accurate and can be used to train your ensemble models. The data is cleaned and preprocessed before being used for model training.

B. *Data Pre-Processing*

1) Normalization: Normalization is a technique used in machine learning to adjust input data values so that they fall within a specific range, typically between 0 and 1. It is also known as Min-Max Scaling. This technique is used when the data has a large range of values, such as with data that includes very large and very small numbers. The normalization process involves transforming the data by subtracting the

minimum value and dividing it by the range, resulting in a new set of values between 0 and 1. It can be expressed mathematically as (x-min) / (max-min), Where x is the original



(a) (b) (c) (d) (e)

Fig. 4: Confusion Matrix (Clockwise: FP, TN, FN, TP) for a)RandomForestClassifier , b)ExtraTreeClassifier, c)ADA Boost, d) Gradient Boost, e) XGBosst

value, min is the minimum value of the dataset, and max is the maximum value of the dataset. Normalization can improve the performance of machine learning models by making the input data more consistent and easier for the model to process. It can also help to prevent certain machine learning algorithms from being sensitive to the scale of the input data, which can produce poor results if the input data is not properly normalized. It's important to note that normalization should only be applied to data that is not already in a range between 0 and 1 and should not be applied to binary data (0,1) or categorical data.

2) Scaling: Data scaling is a technique used in machine learning to adjust input data values so that they fall within a similar range. It is important because many machine learning algorithms are sensitive to the scale of the input data and can produce poor results if the input data is not properly scaled. Several data scaling techniques can be used in machine learning, including 1. Min-Max Scaling: Also known as normalization, this technique scales the data so that all values fall between 0 and 1. It is often used when the data has a large range of values, such as with data that includes very large and very small numbers. 2. Standardization: This technique scales the data with a mean of 0 and a standard deviation of 1. It is often used when the data does not have a well-defined range, such as with data that is normally distributed. 3. Robust Scaling: This technique is similar to Min-Max Scaling, but it considers the data outliers. It is used when the data set has outliers that can affect the overall scaling. 4. Logarithmic scaling: This technique is used when the data set has a skewed distribution; it can help balance large values' effect on the dataset. Overall, the concept of scaling a dataset in machine learning is to preprocess the data to ensure that the input values are within a similar range; this can improve the model's performance and make it more robust to variations in the data.

C. Model Selection

Ensemble methods are techniques that combine multiple models to improve the performance of predictions. Some popular ensemble methods for PCOS detection include:

- Random Forest: An ensemble method generates a set of decision trees and selects the mode of classification (classification) or mean prediction (regression) of the individual trees. The formula for Random Forest is:

$$RF(X) = mode(T1(X), T2(X), ..., Tn(X)) \tag{1}$$

Where RF is the Random Forest model, X is the input data, and T1, T2, ..., and Tn are the individual decision trees.

- Extra Trees Classifier: It is an extension of Random Forest, where each tree is grown using random subsets of both features and samples. The formula for Extra Trees Classifier is similar to Random Forest.
- XGBoost is an optimized version of the Gradient Boosting algorithm for classification and regression problems. XGBoost stands for "Extreme Gradient Boosting" and uses a gradient descent algorithm to minimize the loss function. The formula for XGBoost is:

$$F(X) = F(X) + *h(X) \tag{2}$$

where F(X) is the current model, is the learning rate, and h(X) is the new decision tree added to the model.

- AdaBoost: It is a boosting algorithm that adapts to the difficulty of the training instances. It works by weighting the instances, putting more weight on difficult-to-classify instances. The formula for AdaBoost is:

$$F(X) = F(X) + *h(X) \tag{3}$$

where F(X) is the current model, is the weight of the new classifier, and h(X) is the new decision tree added to the model.

- Gradient Boosting: It is an ensemble method that creates a collection of decision trees, where each tree is grown using the gradient descent algorithm to minimize the loss function. The formula for Gradient Boosting is similar to XGBoost.

IV. Results and Discussions

The results of this study indicate that machine learning-based ensemble methods can effectively diagnose and predict Polycystic ovary syndrome (PCOS). When compared to traditional methods, the ensemble models achieved high accuracy, precision, recall, and F1-score. Furthermore, ensemble methods improved the predictions' robustness by reducing the models' variance and bias. In terms of the specific ensemble methods used, the results showed that Random Forest, Extra Trees Classifier, XGBoost, AdaBoost, and Gradient Boosting all performed well for PCOS detection. However, it was found

TABLE II: Performance of Ensemble Based Machine Learning Algorithms

Algorithms	Accuracy	Precision	Recall	F1-Score
RandomForest	94.0%	94.0%	97.0%	95.0%
ExtraTreesClassifier	94.0%	97.0%	94.0%	95.0%
XGBoost	92.0%	95.0%	93.0%	94.0%
GradientBoost	91.0%	92.0%	95.0%	93.0%
AdaBoost	91.0%	93.0%	93.0%	93.0%

that Rabdom Forest and Extra Tree Classifier had the highest accuracy of 94% and performed the best overall performance.

In terms of feature selection, it was found that clinical and demographic features such as age, body mass index (BMI), and hormonal levels were the most relevant for PCOS detection. Additionally, using a combination of features resulted in better performance than using individual features. Followed by, In terms of model optimization, fine-tuning the parameters of the ensemble models and experimenting with different ensemble methods and combination strategies improved the performance of the models in which Random Forest and Extra Tree Classifier were found to be the best-performing ensemble methods for PCOS detection. Additionally, the study has shown that clinical and demographic features and model optimization are key factors in achieving good performance for PCOS detection.

V. Conclusion

Machine learning-based ensemble methods have been proposed as a promising approach for diagnosing and predicting PCOS. The combination of multiple models, such as decision trees, random forests, and neural networks, can improve the accuracy and robustness of the predictions. This research has highlighted the methodology of using machine learning-based ensemble methods for PCOS diagnosis and prediction, including data preparation, feature selection, model selection, model building, model evaluation, and model optimization. It has also presented an overview of several popular ensemble methods, including Random Forest, Extra Trees Classifier, XGBoost, AdaBoost, and Gradient Boosting. Furthermore, the evaluation metrics such as accuracy, precision, recall, F1-score, and other parameters have been discussed. Though the preliminary research shows enormous potential for the early prediction of PCOS, yet it will be a replacement for Doctors. However, it will exponentially increase an effective medication for the diagnosis of PCOS.

VI. Acknowledgement

The authors acknowledge the support from REVA University for the facilities provided to carry out research.

References

[1] Rachana, B., Priyanka, T., Sahana, K. N., Supritha, T. R., Parameshachari, B. D., & Sunitha, R. (2021). Detection of polycystic ovarian syndrome using follicle recognition technique. Global Transitions Proceedings, 2(2), 304-308.

[2] Sreejith, S., Nehemiah, H. K., & Kannan, A. (2022). A clinical decision support system for polycystic ovarian syndrome using red deer algorithm and random forest classifier. Healthcare Analytics, 2, 100102.

[3] Krishnaveni, V. (2019). A roadmap to a clinical prediction model with computational intelligence for pcos. International Journal of Management, Technology and Engineering, 9(2), 177-185.

[4] Shetty, D., Varma, J., Navi, S., & Ahmed, M. R. (2020). Diving deep into deep learning: history, evolution, types and applications. The International Journal on Media Management, 9, 2278-3075.

[5] Bhat, S. A. (2021). Detection of polycystic ovary syndrome using machine learning algorithms (Doctoral dissertation, Dublin, National College of Ireland).

[6] Nabi, N., Islam, S., Khushbu, S. A., & Masum, A. K. M. (2021, July). Machine Learning Approach: Detecting Polycystic Ovary Syndrome & It's Impact on Bangladeshi Women. In 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE.

[7] Ahmetaevi, A., Alielebi, L., Bajri, B., Bei, E., Smajovi, A., & Deumi, A. (2022, June). Using Artificial Neural Network in Diagnosis of Polycystic Ovary Syndrome. In 2022 11th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-4). IEEE.

[8] Danaei Mehr, H., & Polat, H. (2022). Diagnosis of polycystic ovary syndrome through different machine learning and feature selection techniques. Health and Technology, 12(1), 137-150.

[9] Aggarwal, S., & Pandey, K. (2023). Early identification of PCOS with commonly known diseases: Obesity, Diabetes, High blood pressure and Heart disease using Machine Learning Techniques. Expert Systems with Applications, 119532.

[10] Gudodagi, R., Reddy, R. V. S., & Ahmed, M. R. (2022, April). Deep Learning Algorithm for Procedure and Network Inference for Genomic Data. In International Conference on Artificial Intelligence and Sustainable Engineering: Select Proceedings of AISE 2020, Volume 1 (pp. 493-503). Singapore: Springer Nature Singapore.

[11] Tanwani, N. (2020). Detecting PCOS using machine learning. Int J Modern Trends Eng Sci (IJMTES), 7(1), 1-20.

[12] Balogun, J. A., Idowu, P. A., & Babawale, O. T. DEVELOPMENT OF A PREDICTIVE MODEL FOR THE RISK OF INFERTILITY IN WOMEN USING SUPERVISED MACHINE LEARNING ALGORITHMS.

[13] Zhang, X., Liang, B., Zhang, J., Hao, X., Xu, X., Chang, H. M., ... & Tan, J. (2021). Raman spectroscopy of follicular fluid and plasma with machine-learning algorithms for polycystic ovary syndrome screening. Molecular and cellular endocrinology, 523, 111139.

[14] Silva, I. S., Ferreira, C. N., Costa, L. B. X., Ster, M. O., Carvalho, L. M. L., de C. Albuquerque, J., ... & Gomes, K. B. (2022). Polycystic ovary syndrome: clinical and laboratory variables related to new phenotypes using machine-learning models. Journal of Endocrinological Investigation, 1-9.

[15] Bharati, S., Podder, P., & Mondal, M. R. H. (2020, June). Diagnosis of polycystic ovary syndrome using machine learning algorithms. In 2020 IEEE Region 10 Symposium (TENSYMP) (pp. 1486-1489)

[16] Zigarelli, A., Jia, Z., & Lee, H. (2022). Machine-Aided Self-diagnostic Prediction Models for Polycystic Ovary Syndrome: Observational Study. JMIR Formative Research, 6(3), e29967.

[17] Maheswari, K., Baranidharan, T., Karthik, S., & Sumathi, T. (2021). Modelling of F3I based feature selection approach for PCOS classification and prediction. Journal of Ambient Intelligence and Humanized Computing, 12(1), 1349-1362.

[18] Rakshitha, K. P., & Naveen, N. C. (2022). Op-RMSprop (Optimized-Root Mean Square Propagation) Classification for Prediction of Polycystic Ovary Syndrome (PCOS) using Hybrid Machine Learning Technique. International Journal of Advanced Computer Science and Applications, 13(6).

[19] Nandipati, S. C., Ying, C. X., & Wah, K. K. (2020). Polycystic Ovarian Syndrome (PCOS) classification and feature selection by machine learning techniques. Appl Math Comput Intell, 9, 65-74.

[20] Na, Z., Guo, W., Song, J., Feng, D., Fang, Y., & Li, D. (2022). Identification of novel candidate biomarkers and immune infiltration in polycystic ovary syndrome. Journal of Ovarian Research, 15(1), 80.

sEMG Signal based Hand Gesture Recognition Using Machine Learning

Nithin Kurian
Department of Computer Science,
Cochin University of Science
And Technology
Cochin, Kerala, India
nithinkurian777@gmail.com

Dheeraj S Nair
Department of Computer Science,
Cochin University of Science
And Technology
Cochin, Kerala, India
dheekan1999@gmail.com

Shailesh S
Department of Computer Science,
Cochin University of Science
And Technology
Cochin, Kerala, India
shaileshsivan@cusat.ac.in

***Abstract*—Limb loss is a condition of loss of either upper limbs, lower limbs, or both. Even though there are several reasons, limb loss can be caused primarily due to vascular issues, trauma, or even due to cancer. It is occurring more frequently than expected. For many people suffering from limb loss, it is hard to rediscover their capabilities and limits. It also forms the reason for several issues like depression. A primary solution to tackle this hurdle is the use of prostheses. But, just some prostheses won't be of any good. It should be able to understand the movement and implement it perfectly, as the inability to do so might even lead to even fatal situations. In this paper, an idea for hand gesture classification for amputees suffering from upper limb loss, with the help of sEMG signal is being discussed. This sEMG signal data obtained was applied to several normal classifiers like SVC Classifier, KNN Classifier and Decision Tree Classifier, and ensemble learning classifiers like Random Forest Classifier, XG Boost classifier, and Voting Classifier and their performances were analyzed. The best performance was an accuracy of 88%, which was obtained for the Voting classifier, with SVC being the best performing model used for it**

***Index Terms*—sEMG signal, Random Forest Classifier, Support Vector Machine Classifier(SVC), KNN Classifier, XG Boost, Decision Tree Classifier, Voting Classifier, Grid Search, and Ensemble Learning.**

I. Introduction

Limb loss is a condition of loss of either upper limbs, lower limbs, or both. There are primarily three reasons for limb loss. Vascular diseases like diabetes, amputation caused by accidents and traumas, and amputation due to cancer [1], [2], [3]. Amputees usually are travelling through several physical and emotional issues day to day due to their inability to do their basic activities, not knowing their limit of capability, depression over them being a burden to others, financial hardships, etc. So it is an extremely important thing to find a solution for this situation which will help them to live a life like normal people to an extent. Prosthesis is the primary solution that can solve this issue. Artificial limbs can help them overcome their issues and hurdles. Though there are several works happening in this field, no one was able to get to the intended level. This is one of the primary reasons which motivated us to work in this field. The primary objective while designing an efficient prosthetic is that it should be able to accurately predict the motion or movement intended by the amputees and to correctly implement it. So there is a need of a medium that can easily help in predicting what kind of movement the person intends to do and it should also be easy and fast to interpret. That is where the case of Electro Myograpghy(EMG) signal comes into action. The surface electromygraphy (sEMG) signal is the electrical signal formed due to muscular activities. It has a direct correlation with muscle activity and exercise level [4]. Its principal energy is concentrated between the 0Hz and 500Hz frequency region, and its amplitude typically ranges from 0.01mV to 10mV. In order to use this for prosthesis, first it should be analyzed which machine learning model is the least resource(computation time, prediction, etc.) consuming and gives the fastest and most accurate result.

This paper deals with the implementation of a model for Hand gesture recognition based on the sEMG signals obtained from different subjects and analyses its performance in several machine learning models, including some ensemble learning techniques. Section 2 of the paper deals with the Literature review on the topics, where previous works done on the topic were analysed and better solutions were even considered for the work. Section 3 deals with the Proposed Design, which deals with implementation of the model were the selected dataset is pre-processed, feature extracted and then applied to the selected models in order to train them. Section 4 deals with the result of the implementation and its analysis. Section 5 is the final conclusion of the paper and the possible future expansions.

II. Literature Review

Several works has been happening in the this topic for years. It can be see in some works [6] [8] [9] [10] [13] [14] [15], that Deep learning methods are the most preferred among the researchers due to its better performance and accuracy. Among the Deep learning methods used, the most prominant one is the ANN. In a review paper [8], it has been pointed out that ANN is indeed showing better performance than other deep learning methods for employing in hand gesture classification and human computer interaction. HMM also yields satisfactory recognition results. It can be several other deep learning methods are also being utilised like in [14] a multi stream residual network(MResLSTM) is being used. [8]

suggests other methods like BPNN, LSTM, Bayes Network, etc.

In some research works [16], [10], it can be seen that Deep learning methods methods aren't the only means considered. Normal classifiers and ensemble learning techniques are also being used for research and are also showing promising results, though not the level of Deep learning methods. In a study [10], new model for describing has been put forward, which utilizes statistical properties of subband elements followed by the breaking down of seven levels of DWT(Discrete Wavelet Transform). It is a novel combination of DWT with the Bagging ensemble model. In a work [16], the ensemble classifiers trained and tested using feature vectors created from time-domain features from raw EMG signals and their performances were analysed.

In some works [6], [7], [9], [16], several feature extraction schemes or techniques were used to improve the performance. In a research work [6], CNN was employed for the purpose of feature extraction. In a study, [7], feature extraction is done with the help of the Advanced Energy Kernel-based Features(AEKF) Technique which utilises both time and frequency domain features of EMG signal. Comparing with the methods observed in the paper, the technique is showing significant improvement in terms of rate of pattern recognition, time for computation, etc. In a work [9], first a feature extraction was done on a dataset using reflected constraints. These features capture the frequency properties of the EMG signals.

In a research work [6], both unsupervised and supervised learning techniques were used for recognising hand gestures. Unsupervised learning algorithms like modified Fuzzy C-Means(FCM) and subtractive clustering are the techniques utilized to group gestures as clusters, each containing more than one gesture. A two-stage classification is done on the clusters, top stage classifier splits the data into three clusters, and sub-classifiers are employed to distinguish between the gestures in each cluster.

In a study [7], the feature-extracted dataset was applied to several normal classifiers like SVM, KNN, etc. and their performance was analyzed. In another research work [9], the feature extracted data has been applied performance Extreme Value Machine(EVM) and seven other classifiers to analyze their performance and EVM showed superior performance.

In a study [11], the author proposed an improved framework which identified the features that impacted pattern recognition. The work provides several spectral moment based and inter channel correlation based features which results in its improved performance.

In a work [12], a study on the effects of EMG data's temporal and spatial features were given to the classifier and its offline performance and correlations were analyzed. The impact of window overlap, number of electrode channels and analysis window length on the accuracy of classification and their correlations were thoroughly investigated.

In a research work [13], it was demonstrated that transfer learning is possible between different domains like electroencephalography and electromyography.

Even though Deep learning methods are showing better performance in the case of sEMG signal classification, the hard implementation of such methods is extremely difficult. Moreover, it can also be seen that studies done on ensemble learning techniques are also comparatively low. So, in this paper, it is intended to:-

1) Analyse the performance of different classifiers (other than deep learning methods) for sEMG classification.
2) Checking the performance of Ensemble learning techniques and finding which type is performing the best.
3) Finding which all models to be added in ensemble learning methods to improve their performance.

III. Proposed Design

In this work, analysis of different Machine learning models for sEMG-based Hand gesture recognition is intended to be done and also to find which method is most suited for it. For this, an EMG signal dataset available in Kaggle is selected. Then, pre-processing and feature extraction is done on the dataset to make it model-friendly and also to improve the performance of the model. The data is then split into train and test data. The processed dataset is then applied to several classifier models and their performance is analyzed for both train and test datasets. Based on the analysis, the model with better performance and ease of computation is selected.

There are six classifier models considered for this analysis. Three of them were normal classifiers, Support Vector Machine Classifier(SVC), KNN Classifier and Decision Tree Classifier. A Linear SVC's (Support Vector Classifier) goal is to split or categorise the input data into a ”best fit” hyperplane by fitting the data. The next step is to feed some features to the classifier after obtaining the hyperplane to determine what the ”predicted” class is. The KNN algorithm stores all of the information that is available and categorises a new data point based on similarity (euclidean distance). A Decision Tree Classifier, is a machine learning algorithm, which converts its dataset into a tree-like structure which helps to classify the dataset.

Along with this three ensemble learning classifier models were also used to analyze the performance of ensemble learning in sEMG signal classification. They are Random Forest Classifier from Bagging, XG Boost Classifier from boosting, and Voting Classifier. Random Forest classifier is a machine learning algorithm which incorporates the ensemble logic to classify data items. It makes use of multiple decision trees and combines the results altogether to generate the correct prediction. XG Boost is an example of boosting ensemble modeling, a method that aims to create a strong classifier out of many weak classifiers. By adding weak models in series, a model is constructed and the successive model is used to correct the errors of its predecessor. A Voting Classifier trains on an ensemble of numerous models and gives a class(output)

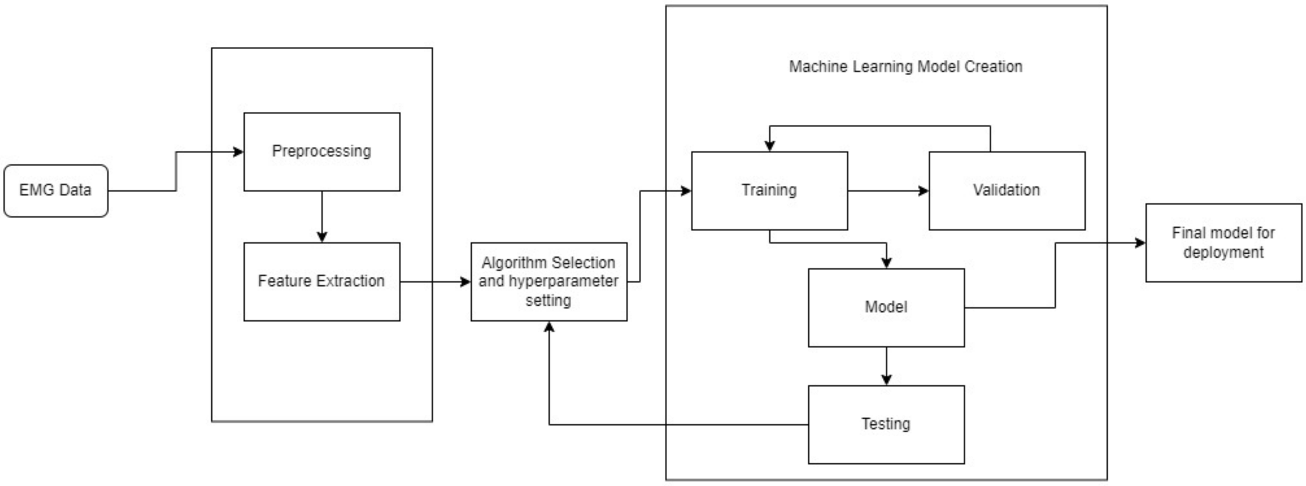

EMG Data
Preprocessing
Feature Extraction
Algorithm Selection and hyperparameter setting
Machine Learning Model Creation
Training
Validation
Model
Testing
Final model for deployment

Fig. 1. Overall Architecture

based on the maximum number of votes obtained. There are hard voting and soft voting, of which hard voting is used here.

Gridsearch [5] is a hyper-parameter selection method. It conducts an exhaustive search over specified parameter values for an estimator(model). Cross-validated grid-search is used to refine the estimator's parameters over a parameter grid. In this way, the hyper-parameter tuning becomes more efficient than the Trial and Error or Brute force method. The method is applied to all models in order to get their best performance.

A. Dataset

The dataset was obtained from Kaggle as a .csv file. There are 11 columns in the dataset [17].

The patterns were recorded by using a MYO Thalmic bracelet by the user at his/her forearm, which was then transmitted to the PC with a bluetooth receiver. Eight sensors evenly placed around the forearm are built inside the bracelet, and they simultaneously collect myographic signals, which is transferred to PC with the help of Bluetooth interface. The dataset shows 36 subjects' raw EMG data while they made a series of static hand gestures. Each subject performs two series consisting of six or seven gestures each. The duration of each gesture was 3 seconds, with a 3 second break in between each gesture.

Description of raw datafile. There are 10 columns in the datafile:
1) Time - in ms;
2-9) Channel - eight EMG channels of MYO Thalmic bracelet;
10) Class –the label of the hand gesture:
Class 0 - unmarked data,
Class 1 - hand at rest,
Class 2 - hand clenched in a fist,
Class 3 - wrist flexion,
Class 4 – wrist extension,
Class 5 – radial deviations,
Class 6 - ulnar deviations,
Class 7 - extended palm (the gesture was not performed by all subjects).

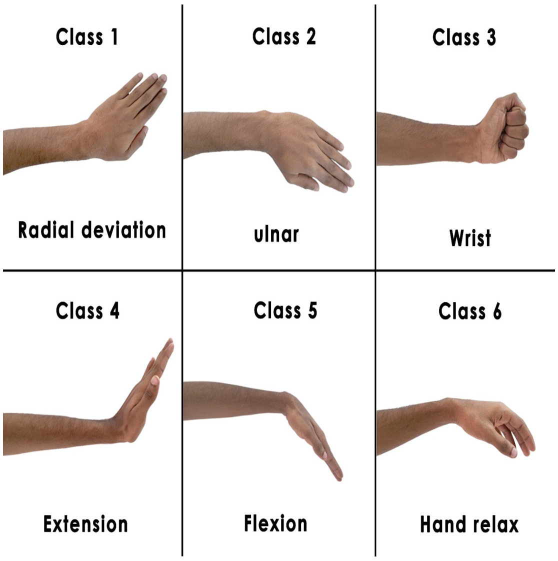


Fig. 2. Hand gestures

A "label" column has been added to the dataset, that points to the subject who has performed the gestures. There were 36 subject, who performed 7 gestures each twice.

$\mathbf{X}$ = Dataset
$\mathbf{X}_{C11}$= Output of subject 1 showing gesture Class 1

B. Data Pre-Processing

The first task was to check for null values and as there wasn't any we moved on to the next task, ie. taking value counts of each classes. It can be seen that, Class 7 data is extremely low compared to others as only two of 36 patients are only performing it. So that column was dropped. Even though there are several entries in Class 0, as it is unmarked data, no relevant information is also obtained from it. So that column is also dropped. By this step our pre-processing tasks are done. So as the pre-processing works are completed, next step is feature extraction.

C. Feature Extraction

Before feature extraction, a groupby function is used on the dataset based on the subject and target performed by the subject to group all readings at different instants of time at each sensor. Taking into consideration all instances directly into consideration will not be good for prediction. Now feature extraction done by taking aggregated sEMG features of data from each channel like minimum value, maximum value, rms value, simple square integral(SSI), absolute differential signal(ADS) and ptp value(range of values(max value - min value)).

$$Min(\mathbf{X}_{C11}) = Mininmum(\mathbf{X}_{C11}) \tag{1}$$

$$Max(\mathbf{X}_{C11}) = Maximum(\mathbf{X}_{C11}) \tag{2}$$

$$Rms(X_{C11}) = \sqrt{\frac{1}{n} \times \sum_{i} X_{C11_i}^2}$$

$$SSI(X_{C11}) = \sum_{i=1}^{N} X_{C11_i}^2$$

$$ADS(X_{C11}) = \sum_{i=1}^{N} |X_{C11_i}|$$

$$PtpValue(X_{C11}) = [Max(\mathbf{X}_{C11}) - Min(\mathbf{X}_{C11})] \tag{6}$$

Earlier the dataset consisted of 10 columns and approximately 2.5lakh entries. By doing this extraction process it was able to successfully convert the 2.5 lakh entries to 216 rows and 50 columns with six values for each sensor grouped for each subject and each action. Now the dataset is ready for training.

D. Training the Model

Now moving on to the major part of the work, that is the training and testing of the model. In order to conduct it, the dataset needs to be split into two parts train and test data. It is done based on the test size and random state specified.

In cross-validation (evaluation technique), the training data is split into subsets and training and validation takes place on different subsets. That technique is applied here. The training dataset is then applied to all six classifier models that were selected. First, the training dataset is applied on each of these models to train the models. After that, the test dataset is used to test and validate the model's performance based on the training. There are evaluation tools like accuracy, F1 score, etc. which can be used to analyze the performance, prediction error, and efficiency of the models.

After applying the test dataset and getting the results and evaluation metrics values, now it's time for the final stage of the work, ie. analysis of the result.

IV. Results and Analysis

The Evaluation metrics like precision, recall, accuracy, and F1 score values obtained for each model are shown in the following table and graph. If it had only metric we could have said it might have been biased due to the abundance of data of a particular class, etc. It is the reason why more than one metric was selected and also it helps to analyze were the error lies, etc.

Considering the case of a binary classification, which classifies data into positive class or negative class, the no. of Positive class elements predicted as the positive class itself is called True Positive(TP), the no. of Negative class elements predicted as the positive class itself is called False Positive(FP), the no. of Positive class elements predicted as the negative class itself is called False Negative(FN) and the no. of Negative class elements predicted as the negative class itself is called True Negative(TN). Taking this into consideration, the evaluation metrics precision, recall, accuracy, and F1 score can be found by,

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{9}$$

$$F1 - score = 2 \times \frac{Precision * Recall}{Precision + Recall} \tag{10}$$

In case of a multi-class classification problem, the metrics is calculated for each classes seperately and their average is taken as the final value of the metric. For precision, recall and accuracy the average of metric is taken. As for F1 score, there two methods F1macro or F1micro can be taken depending on the objective. Here, F1macro is selected as it is similar to taking the average. Along with that ROC curves are added to show the separability of the classes by all possible thresholds, or in other words, how well the model is classifying each class. The performance metrics are shown below.

Model	Precision	Recall	Accuracy	F1 Score	CV Score
Support Vector Classifier	0.873	0.854	86.4	0.857	0.814
KNN Classifier	0.823	0.812	81.8	0.803	0.837
Decision Tree Classifier	0.69	0.657	65.9	0.66	0.68
Random Forest Classifier	0.813	0.801	81.8	0.796	0.82
XG Boost Classifier	0.806	0.786	79.5	0.785	0.803
Voting Classifier	0.883	0.877	88.6	0.876	0.854

Comparison of Accuracy and F1 score

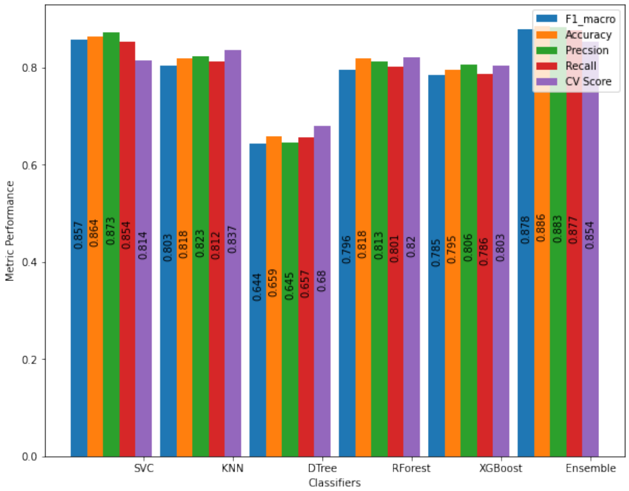


Fig. 3. Comparison Graph

It can be seen that the Voting Classifier has the best performance comparing to all other models in case of all metrics. It can also be seen that SVC, Random Forest and KNN better performance in case of sEMG based classification. By analysing each class for finding which models gives best prediction for that class(excluding Voting Classifier), it can be seen that Random Forest has better classification in case of Class 1, as is has no errors. SVM and KNN are good at Class 2, as they only 2 errors. SVM is best at Class 3, 3 errors. Random Forest and XG Boost are best at Class 4, has only 1 error. SVM is best at classifying Class 5. SVM, KNN and XG Boost are best for Class 6. Voting Classifier takes the best of all these models, which makes it the best performer among all models. So, ensemble learning techniques are giving a better performance than normal machine learning models and also it would best to take SVM as on of the models in ensemble learning as it is the best normal classifier in the case of sEMG signal classification.

Another observation seen here is that, the most number of errors coming are regarding Class 6 gesture, either as false predictions as the gesture is class 6 or as predicting the Class 6 gesture as some other gesture. The major reason for this is the closeness of extracted values of Class 6 with that of the features of other classes, especially Class 3 as the most no. of predictions are occurring between Class 3 and Class 6. As said earlier, the best performance among the models in identifying Class 6 gesture is shown by the Voting classifier, SVM, KNN and XG Boost. Among these models, XG Boost is best at classifying Class 3 and 6 perfectly as there is only one error prediction.

It can also be observed that among the ensemble learning techniques used here bagging ensemble learning techniques like Voting Classifier and Random Forest are giving better performance than boosting ensemble learning technique XG Boost. So, bagging ensemble learning techniques are best suited for sEMG signal classification. Confusion matrices of different models and ROC curves of different classes for SVM are given on the following page. It can be used to refer to the performance of the models in the dataset.

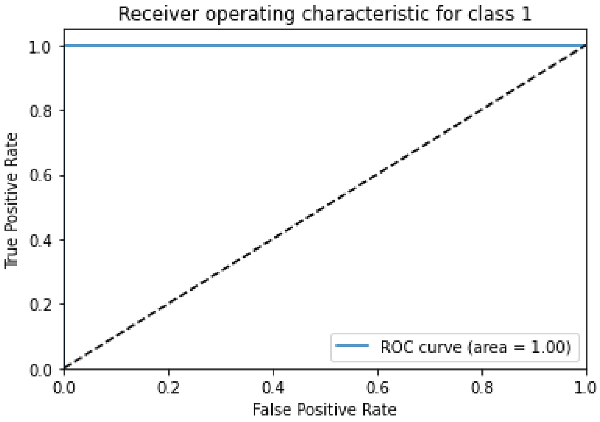


Fig. 4. ROC curve of Class 1 gesture

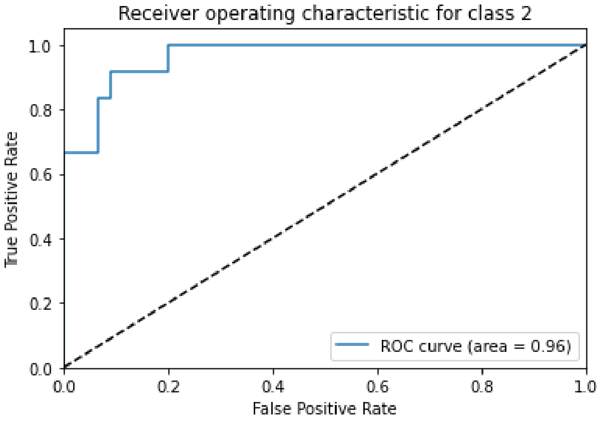


Fig. 5. ROC curve of Class 2 gesture

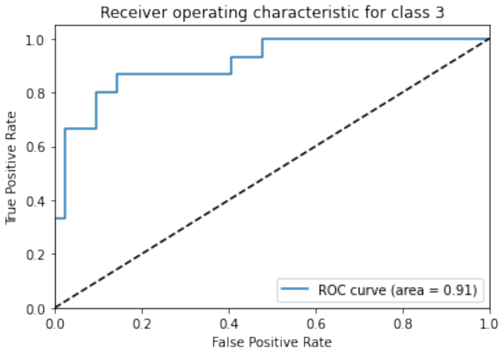


Fig. 6. ROC curve of Class 3 gesture

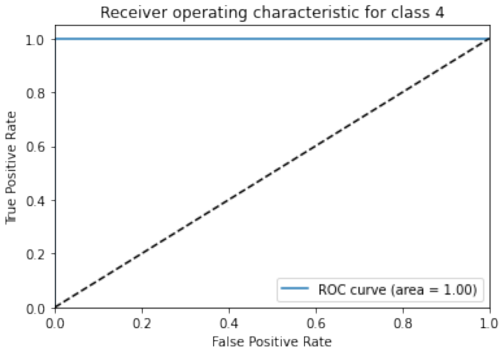


Fig. 7. ROC curve of Class 4 gesture

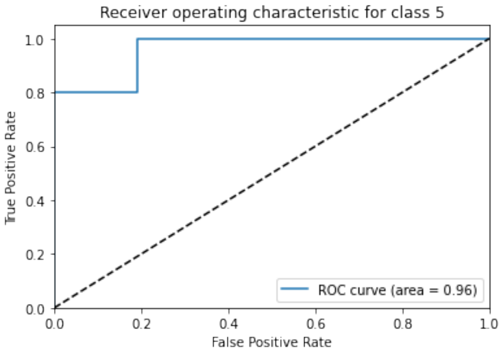


Fig. 9. ROC curve of Class 5 gesture

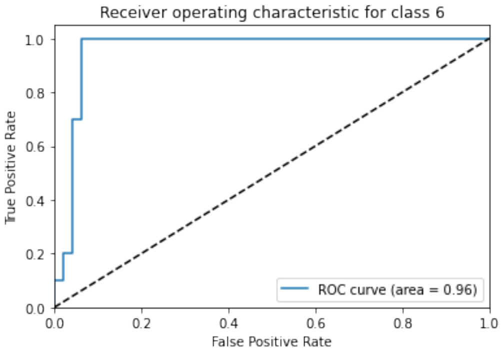


Fig. 10. ROC curve of Class 6 gesture

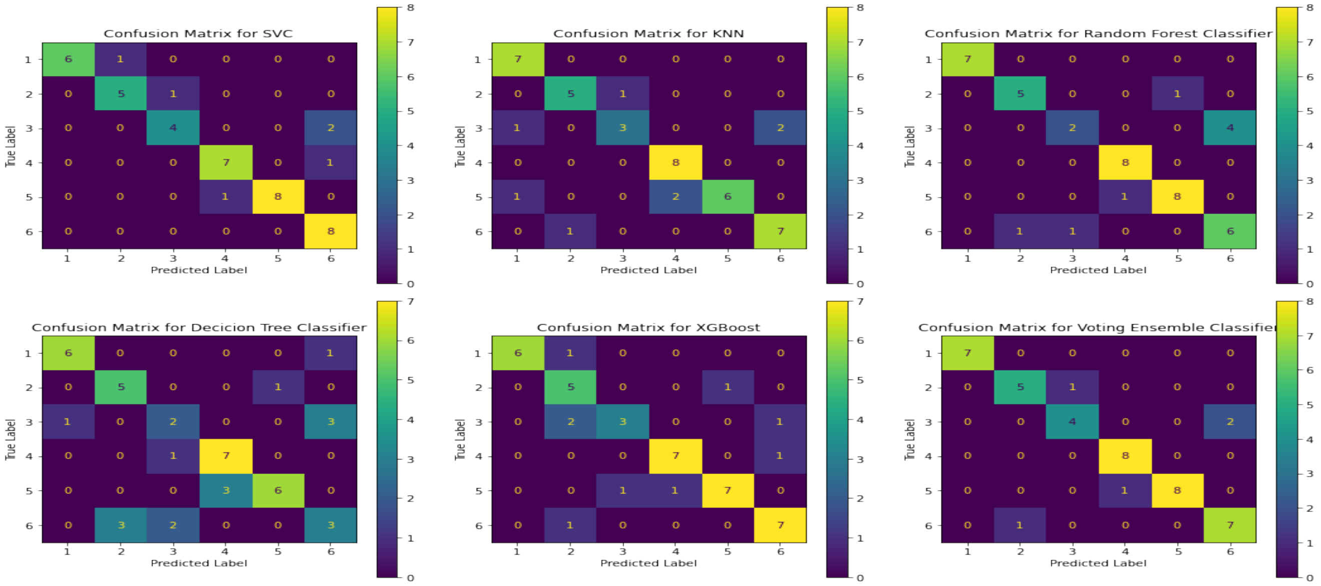


Fig. 8. Confusion Matrices for different classifiers

V. Conclusion

In this paper, an analysis for best suited model for sEMG signal based Hand gesture recognition. It has been found that ensemble learning techniques, especially bagging techniques is giving better performance than normal classifiers. Here, the selected dataset was then converted with the help of a feature extraction scheme where six aggregated EMG features (minimum value, maximum value, rms value, small square integral, absolute differential signal and ptp value) were taken. This helps in reducing both the computation time and and improving the performance of the model. Several normal(SVC, KNN, Decision tree) and ensemble learning(Random Forest, XG Boost and Voting) classifiers were selected and the feature extracted dataset was applied to and their performances were analysed. Based on this experiment it has been found that Voting Classifier gives better performance for EMG signal based Hand gesture recognition. Also that SVC is the best normal classifier than the others, which means it gives better performances if SVC was used in ensemble learning techniques. This also paves way for hardware implementation of intelligent prosthesis in future.

References

[1] Hanger Clinic, 'Limb Loss and Limb Difference: Facts, Statistics, Resources', 2021. [Online]. Available: https://hangerclinic.com/blog/prosthetics.

[2] Prathusha, Maduri., Hossein, Akhondi., 'Upper Limb Amputation', 2022. [Online]. Available: www.ncbi.nlm.nih.gov.

[3] Amputee Coalition, 'Limb Loss Statistics', 2023. [Online]. Available: https://www.amputee-coalition.org/.

[4] Mayo Clinic, 'Electromyography(EMG)', 2019. [Online]. Available: https://www.mayoclinic.org/.

[5] scikit-learn, 'Grid SearchCV', 2023. [Online]. Available: https://scikit-learn.org/.

[6] Guangyu, Jia, Hak-Keung, Lam, Shichao, Ma, Zhaohui, Yang, Yujia, Xu, and, Bo, Xiao, "Classification of Electromyographic Hand Gesture Signals Using Modified Fuzzy C-Means Clustering and Two-Step Machine Learning Approach," *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING*, VOL. 28, NO. 6, JUNE 2020.

[7] Pancholi, Siddharth, and, Joshi, M., Amit, "Advanced Energy Kernel-Based Feature Extraction Scheme for Improved EMG-PR-Based Prosthesis Control Against Force Variation", *IEEE TRANSACTIONS ON CYBERNETICS*, .

[8] Ibrahimy, Muhammed, and, Khalifa, Omran, Othman"EMG Signal Classification for Human Computer Interaction A Review",*European Journal of Scientific Research*,January 2009.

[9] Reza, Bagherian, Azhiri, Mohammad, Esmaeili, Mohsen, Jafarzadeh, Mehrdad, Nourani, "EMG Signal Classification Using Reflection Coefficients and Extreme Value Machine"*IEEE Biomedical Circuits and Systems Conference (BioCAS 2021)*, 2021.

[10] Abdulhamit, Subasi, Emine, Yaman, Yara, Somaily, Halah, A., Alynabawi, Fatemah, Alobaidi, Sumaiah, Altheibani, "Automated EMG Signal Classification for Diagnosis of Neuromuscular Disorders Using DWT and Bagging", *Complex Adaptive Systems Conference with Theme: Cyber Physical Systems and Deep Learning*, CAS 2018, 5 November – 7 November 2018, Chicago, Illinois, USA.

[11] Turlapety, C., Anish, Gokaraju, Balakrisna "Feature Analysis For Classification of Physical Actions using surface EMG data" *IEEE Sensors Journal, vol. 19, no. 24, pp. 12196-12204*, 2019.

[12] Menon, Radhika, Student Member, IEEE, Gaetano Di Caterina, Lakany, Heba, Member, IEEE, Lykourgos Petropoulakis, Conway, A., Bernard, Soraghan, J., John, Senior Member, IEEE, "Study on interaction between temporal and spatial information in classification of EMG signals for myoelectric prostheses "*IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 25, no. 10, pp. 1832-1842*, Oct 2017.

[13] Jordan, J., Bird, Jhonathan Kobylarz, Diego, R., Faria, Aniko, Ekart, and, Eduardo, P., Ribeiro, "Cross-Domain MLP and CNN Transfer Learning for Biological Signal Processing: EEG and EMG" *IEEE Access, vol. 8, pp. 54789-54801*, 2020.

[14] Zhiwen, Yang., Du, Jiang., Ying, Sun., Bo, Tao., Xiliang, Tong., Guozhang, Jiang., Manman, Xu., Juntong, Yun., Ying, Liu., Baojia, Chen., and Jianyi, Kong., "Dynamic Gesture Recognition Using Surface EMG Signals Based on Multi-Stream Residual Network",*Frontiers in Bioengineering and Biotechnology*, 2021.

[15] Md., Rezwanul, Ahsan., Muhammad, Ibn, Ibrahimy., Othman, O., Khalifa., "Electromygraphy (EMG) Signal based Hand Gesture Recognition using Artificial Neural Network (ANN)"*4th International Conference on Mechatronics (ICOM)*, 2011.

[16] Bhattacharjee, Debarati., Singh, Munesh., "Time-domain Feature and Ensemble Model based Classification"*Research Square*, 2021.

[17] Sojan, Prajapati, 'EMG data for gestures', 2021. [Online]. Available: https://www.kaggle.com/datasets/sojanprajapati/emg-signal-for-gesture-recognition.

Network Science and Climate Dynamics - A Mini Review

Abhijith S Babu
Department of Computer Science and Engineering
Indian Institute of Information Technology Kottayam
India
abhijithbabu2019@iiitkottayam.ac.in

Divya Sindhu Lekha
Department of Computer Science and Engineering
Indian Institute of Information Technology Kottayam
India
divyaslekha@iiitkottayam.ac.in

***Abstract*—Complex systems with multiple variables and their non-linear interactions are too complicated to be investigated directly. Analysis of these systems as complex networks gives us a better understanding of the systems. Global climate systems are complex with multiple variables and their non-linear interaction. They are represented and studied as a complex network which provides tools for analysing patterns in the system. Complex networks offer an interesting framework for representing the dynamics of the climate system. In this project, we discuss how a climate network can be modelled, which represents the interlinking of the climate data in different geographical regions. The analysis of the climate network gives us insights which can be applied in the study of climate events. Their application in predicting *El-Nino* events and the *Indian Monsoon* has been reviewed in this paper. From this perspective, we can argue that the complex network analysis approach can gainfully complement the numerical modelling.**

***Index Terms*—climate system, complex network, spatial network construction, predictive modeling**

I. Introduction

Climate indicates a pattern of weather in a specific geographic location. It is reliant on several variables like temperature, rainfall, air pressure, wind speed and humidity. This array of factors could exhibit changes which manifest as an indelible change in the climate. Climate change definitely shakes up our society to think in a more pragmatic way. Various studies reveal this correspondence between climate change and societal actions. Hence, studying the climate has become more important than ever before. There are various exacerbating consequences of climate change like human or animal migration, the spreading of social propaganda and various climatic disasters and the shifting of economic resources that go along with it. The causes for this climate change could be natural or man-made. It is to be observed here that these causes and consequences are invariably connected together to form a web [1]. The traditional use of hypothesis-driven statistical and data science methods may not be able to capture every cause and consequence to study climate change properly. The climate system looks like a suitable candidate for complex network analysis.

A complex network is a mathematical graph that possesses features which are not found in a simple graph. A graph is a mathematical object that is used to represent networks. The features of a complex network exhibit patterns which are not purely random. A complex network can be analyzed in various ways, by going deep into each node or by looking at the network as a whole. In a complex network, we can identify features such as lattices and random graphs which are helpful in capturing the dynamics of real-life scenarios [12]. In the case of a climate system, the causes and consequences of climate change are interconnected and are coupled into a casual web [1].

Generating functional networks from non-linear time series data aid in the study of complex and dynamic entities like climate systems, brain systems or others like economic, technological or social systems [11]. The complex network study of climate gives us a descriptive analysis as well as a model for predictions. The structural properties of the networks like the degree of nodes, centrality, betweenness and clustering can have a useful interpretation within the domain [17]. These structural properties are significantly better predictors than the traditional approaches [10].

The patterns in climate networks help in a better understanding of the complex process underlying various climatic events. The climate networks are used to describe the physical properties of a climate system and it is compared with the phenomena to get insights. Some of the prior works in these domains showcase data mining techniques in climate which are used to extract climate indices from historical data by clustering and correlating the clusters. Various numerical models were used in the prediction of climate for a long time, which depends on a large number of variables that made it chaotic. There was always a gap between the sub-seasonal weather forecast and long-term climate predictions. Near-term prediction is very challenging in the numerical modelling method. It failed to give sufficient warning time ahead of the climatic phenomenon in many cases.

II. Complex network construction

For a better understanding of the climate system, the climate network can be created in such a way that the geographical locations will serve as nodes while links between two nodes are determined by the correlation between variables of those nodes. The network thus formed can be analysed using various methods, which complement the numerical models explicitly. This is done so as to obtain additional information on the connectivity of two geographical locations by measuring the similarity in the dynamics of their physical quantities. Threshold values can be used to find the significant links among them. The network thus formed can be represented as an adjacency matrix [2], in which a link between two nodes (A_{ij}) is defined by equation 1, as shown below.

$$A_{ij} = \begin{cases} \text{non-zero, if there is a link from node } j \text{ to node } i \\ 0\text{, otherwise} \end{cases} \tag{1}$$

The adjacency matrix thus formed can be used for the calculation of network properties such as in and out degrees, clustering coefficients, betweenness centrality, and so on. One thing to note here is that the network doesn't have a fixed structure. The network variables are time variants, and so are the links. This makes the climate network a fully functional network, where the network properties are time-dependent.

One of the most direct methods to determine the connection strength is *Pearson correlation*. We can also use methods like the event synchronisation method, mutual formation method, e-recurrence method and so on. To calculate the *Pearson correlation*, a variable T is defined at different locations, where $T_i(t)$ is the time series of the variable at i^{th} node, then we can define the *time-delayed Pearson correlation* between i^{th} and j^{th} node as shown in equation 2 and equation 3 [3].

$$c^y_{i,j}(-\tau) = \frac{\langle T^y_i(t) T^y_j(t-\tau)\rangle - \langle T^y_i(t)\rangle \langle T^y_j(t-\tau)\rangle}{\sqrt{\langle (T^y_i(t) - \langle T^y_i(t)\rangle)^2\rangle} \sqrt{\langle (T^y_j(t-\tau) - \langle T^y_j(t-\tau)\rangle)^2\rangle}} \tag{2}$$

$$c^y_{i,j}(\tau) = \frac{\langle T^y_i(t-\tau) T^y_j(t)\rangle - \langle T^y_i(t-\tau)\rangle \langle T^y_j(t)\rangle}{\sqrt{\langle (T^y_i(t-\tau) - \langle T^y_i(t-\tau)\rangle)^2\rangle} \sqrt{\langle (T^y_j(t) - \langle T^y_j(t)\rangle)^2\rangle}} \tag{3}$$

where τ is the time lag.

Using the above equations, we can define the positive and negative link strengths between node i and j as shown below [3].

$$W^{+,y}_{i,j} = \frac{max(C^y_{i,j}) - mean(C^y_{i,j})}{std(C^y_{i,j})} \tag{4}$$

$$W^{-,y}_{i,j} = \frac{min(C^y_{i,j}) - mean(C^y_{i,j})}{std(C^y_{i,j})} \tag{5}$$

The correlation between two nodes gets tricky when there are multiple variables involved. This is an intuitive concept and the best solution to this is yet to be found. One approach used for this purpose was to create a feature space with a pairwise correlation of each pair of variables [4]. The edge weights determine the relationships between the nodes.

III. Case Study

Complex networks allowed the study of climate systems in a way traditional methods cannot. It helped in the prediction of climatic events by giving insights into a number of cases. In this section, we present two specific case studies (*El-Nino Southern Oscillation* and *Indian Monsoon*) in which the complex network approach has given promising results. An overview of the dataset formats and network construction techniques is given below for a better comprehension of the case studies.

A. Dataset Formats

The data required for creating climate networks is available in *NetCDF* format. *NetCDF* is a machine-independent data format that is used to create array-oriented scientific data [8]. The required data is available in different climate data repositories like the NCEP/NCAR reanalysis project [14]. This dataset was created from the sensor measurement of climate data around the globe, from the year 1948 onwards. It considers a wide range of surface and atmospheric climate descriptors. This dataset considers seven variables: sea surface temperature, sea level pressure, geo-potential height, precipitable water, relative humidity, horizontal wind speed and vertical wind speed.

B. Spatial Network construction techniques

The global climate system is inherent with several periodic behavioural patterns on spatial and temporal levels. The dynamic spatiotemporal patterns at different regions are interdependent with each other. A climate network of interacting regions can adequately capture the dynamic behaviour of the system along with the local and global topology of the system [15]. The nodes of the network refer to the grid points around the globe. The weights of the edges are calculated based on the statistical relationships between the pairs of time series of corresponding nodes. For the pair of nodes to have a link between them, the link strength between the nodes has to be determined, which quantifies the relations between the nodes.

The data, which is available in the *NetCDF* format can be read and manipulated using various software packages. *NetCDF4* is a python library used to manipulate the climate data available in the *NetCDF* format. The data from the dataset is normalized, so only the deviation from the mean value has to be considered. In that case, the Pearson correlation coefficient can be used as an effective method to find the link strength. The correlation coefficient of two series A and B can be calculated as shown in equation 6 [8]

$$r(A,B) = \frac{\sum_{i=1}^{t}(a_i - \overline{a})(b_i - \overline{b})}{\sqrt{\sum_{i=1}^{t}(a_i - \overline{a})^2 (b_i - \overline{b})^2}} \tag{6}$$

The correlation coefficient always takes the value between -1 and 1. Since we want a value that is either very low or very high, we take the absolute value of the correlation coefficient. Temporal lags are ignored in this process. The network will be pruned by removing the links with very low link strength. To find the threshold value for selecting a link, various methods are used. Some opted for a threshold value of $r > 0.5$ [9], while some used a fixed edge density to compare different networks [10]. Usually, the threshold is decided based on parametric and non-parametric significance tests.

For each variable, networks are thus formed by calculating the weight and pruning. Additionally, the network construction can be done using Python software package called *pyunicorn*, which is an open-source software package for data modelling and analysis of complex network theory. It implements methods from both complex network theory and non-linear time series analysis and unites these approaches in an adaptable way [11].

These networks are used for the descriptive analysis of climate. We can calculate the properties of networks such as the number of nodes and edges, clustering coefficient, characteristic path length, etc. Patterns can be extracted from the data by performing clustering. Thus, homogeneous regions in the network can be identified. This process is also called community detection in network sciences. Community detection is important in climate networks because it considers the network distance rather than pairwise distances.

Case I: El-Nino Southern Oscillation

El-Nino southern oscillation plays a huge role in the climate variability in the world. It can trigger extreme climatic disasters. The phenomenon is caused due to irregular variations of winds in the Pacific Ocean which affects the climate of tropical and subtropical regions. Trade winds usually create a current in the Pacific Ocean, which moves the hot ocean water towards the east Asian coast. This balances the climate in Asian and American objects. Sometimes, the trade winds slow down, which decreases the current and thus affects the balance of weather. This can cause drought in Southeast Asian countries and Australia and may be even in India. Instead, it gives heavy rainfall to the Peruvian coast which results in heavy flooding.

A complex network approach was used to develop a method for the early forecasting of *El-Nino* events with better accuracy [5]. This approach has provided tools to predict whether an *El-Nino* event will occur one year in advance and had a success rate of 80%. The *El-Nino* event can be quantified using the oceanic nino index or ONI. It is the difference between the oceanic surface temperature and its average temperature, on a rolling three-month average. The ONI value is measured using a network of buoys placed on each node point, that measures temperature wind and current in the equatorial region.

In the network analysis approach of *El-Nino* forecasting, the atmospheric temperature at grid points inside and outside the *El-Nino* basin is considered as nodes which are connected to each other using time evolution connections. The strength of the connection is represented by cross-correlation methods. The mean link strength of the network can be obtained by averaging all individual links in the network. This is a time variable function denoted by $S(t)$. When $S(t)$ is above a certain threshold value, if the ONI is below 0.5 degree [6], it predicts an *El-Nino* in the following year.

Case II: Indian Summer Monsoon

Indian Summer Monsoon (*ISM*) is the intense rainfall received by Indians during the summer season. It usually spans from June to October and it is the primary source of freshwater in India [23]. The exact date of the start of the monsoon varies each year. *Monsoon* is very important for the country as it affects billions of people in India for whom agriculture is the major source of living. *Monsoons* can also bring storms or heavy flooding in some parts of the nation. An early forecast of the *Monsoon* is thus very important. It is possible to predict the onset of *Monsoon* on the western coast of India, but not so practicable regarding the other parts of the country. A forecast of the *Monsoon* withdrawal which prompts many climatic mishaps is also unavailable.

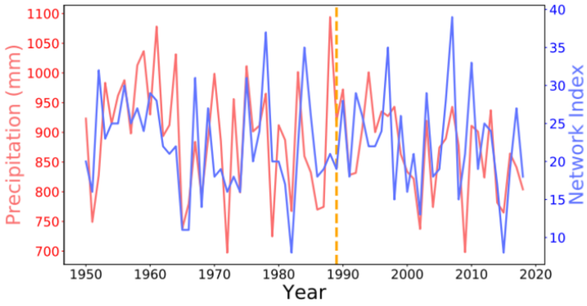


Fig. 1. The figure shows how the precipitation and the network index were correlated in [7]

Complex network analysis has been used to improve the forecasting capability of the onset and withdrawal of the *Indian Monsoon* [7], [19]–[21]. A complex network-based method has allowed predicting the onset of *Monsoon* 40 days in advance and the withdrawal of *Monsoon* 70 days in advance. Some of the complex network analysis of climate has predicted the *Annual Indian Rainfall* with less than 2% deviation.

In a study which forecasted the onset and withdrawal of *Monsoon* [7], the surface daily air temperature taken from the *NCEP/NCAR reanalysis* is used to construct a climate network for each year since 1948. Nodes are chosen from geographical locations homogeneously spread across the world. The link strength between two nodes is calculated based on the correlation equation, and those which fall in the top 5% positive strength are considered. The links were directed based on the value of the variables. In this study, the *degree* of a node, both *in-degree* and *outdegree*, was given the highest importance. The *in-degree* and *outdegree* of a node is a time-variant function, which is proved to have a robust and strong correlation with the rainfall index. See figure 1.

Furthermore, the complex relationship between different climate events was also predicted using functional network construction and analysis. For example, the patterns of extreme rainfall owing to the intricacies between the *Indian Summer Monsoon* and the *East Asian Summer Monsoon* events are investigated recently [18].

IV. Future Scope and Insights

The climate network methods, unlike the traditional methods, are capable of capturing complex relationships and incorporating the predictive modelling into a single framework. So, it has led to good insights and is believed to have even great potential. Non-linear relationships in climate data are not relevant in the context of network construction. An extensive study of their significance by comparing different correlation measures can be made in this regard.

The methods used to capture the multivariate relationships can be improved. The methods used so far are naive. It has to be quantitatively captured and integrated with the network to achieve a more realistic representation. More advanced computational methods can be devised to measure multivariate dependence. As time goes by, datasets continue to increase in size. Novel algorithms and efficient implementations will become a necessity. The scope of the network analysis has to be expanded to include more variables and allow additional spatial or temporal lags. Thus, network analysis can improve the study of complex systems to a larger extent.

V. Conclusion

Complex systems are very helpful in the study of natural dynamic systems. It reveals the interesting mechanisms underlying the phenomena. The recent success in machine learning and deep learning helps us to extract information from large time-varying data. This has great potential application in climate network analysis. The interactions between different parts of the coupled earth system are made possible through a climate network. The network-based analysis can articulate the connection of statistical information on recurring spatial co-variability patterns with the underlying mechanisms. Network science can thus find a way into the toolbox of climate scientists.

References

[1] P. Holme, J.C. Rocha, Networks of climate change: Connecting causes and consequences, Appl Netw Sci 8, 10 (2023).

[2] J. Ludescher, M. Martin, N. Boers, A. Bunde, C. Ciemer, J. Fan, S. Havlin, M. Kretschmer, J. Kurths, J. Runge, Network-based forecasting of climate phenomena, PNAS Vol. 118, No. 47 (2020).

[3] M. Jusup, P. Holme, K. Kanazawa, M. Takayasu, Z. Romi c, I.and Wang, Social physics. Physics Reports, 948, 1-148 (2022).

[4] N. Boers, J. Kurths, N. Marwan, Complex systems approaches for Earth system data analysis, J. Phys. Complex. 2 011001 (2021).

[5] J. Ludescher, A. Bunde, S. Havlin, H.J. Schellnhuber, Very early warning signal for El Nino in 2020 with a 4 in 5 likelihood, arXiv:1910.14642 [physics.ao-ph] (2019).

[6] A. Gozolchiani, S. Havlin, K. Yamasakir, The Emergence of El-Ni no as an Autonomous Component in the Climate Network, Phys. Rev. Lett. 107, 148501 (2018).

[7] J. Fan, J. Meng, J. Ludescher, Z. Li, E. Surovyatkina, X. Chen, J. Kurths, H.J. Schellnhuber, Network based Approach and climate Change Benefits for Forecasting the Amount of Indian Monsoon Rainfall, Journal of Climate, 35(3), 1009-1020 (2020).

[8] K. Steinhaeuser, N.V. Chawla, A.R. Ganguly, Complex Networks as a Unified Framework for Descriptive Analysis and Predictive Modeling in Climate, Statistical Analy Data Mining, 4: 497-511. https://doi.org/10.1002/sam.10100 (2011).

[9] A. Tsonis, K.L. Swanson, P.J. Roebber, What do networks have to do with climate?, Bulletin of the American Meteorological Society, vol. 87, no. 5, pp. 585–595, doi:10.1175/BAMS-87-5-585. (2006).

[10] J.F. Donges, Y. Zou, N. Marwan, J. Kurths., Complex networks in climate dynamics. European Physics Journal, Special Topics, 174:157–179 (2009).

[11] J.F. Donges, J. Heitzig, B. Beronov, M. Wiedermann, J. Runge, Q.-Y. Feng, L. Tupikina, V. Stolbova, R.V. Donner, N. Marwan, H.A. Dijkstra, and J. Kurths, Unified functional network and nonlinear time series analysis for complex systems science: The pyunicorn package, Chaos 25, 113101 (2015)

[12] .A.-L. Barabási, M.Pósfai, Network science, Cambridge University Press, Cambridge (2016).

[13] .M. Haas, B. Goswami, U. von Luxburg, U., Pitfalls of Climate Network Construction - A Statistical Perspective, Journal of Climate, pages 1—48 (2023).

[14] Kalnay et al.,The NCEP/NCAR 40-year reanalysis project, Bull. Amer. Meteor. Soc., 77, 437-470, (1996).

[15] F. Falasca, A. Bracco, A. Nenes, I. Fountalis, Dimensionality Reduction and Network Inference for Climate Data Using δ-MAPS: Application to the CESM Large Ensemble Sea Surface Temperature, Journal of Advances in Modeling Earth Systems, vol. 11, no. 6, pp. 1479–1515, 2019.

[16] A. Bracco, F. Falasca, A. Nenes, I. Fountalis, C. Dovrolis Advancing climate science with knowledge-discovery through data mining., npj Clim Atmos Sci 1, 20174 (2018).

[17] M. Newman, A. L. Barabasi, D. J. Watts, The structure and dynamics of networks, Princeton University Press, (2006).

[18] S. Gupta, Z. Su, N. Boers, J. Kurths, N. Marwan, F. Pappenberger, Interconnection between the Indian and the East Asian summer monsoon: Spatial synchronization patterns of extreme rainfall events., International Journal of Climatology, 43(2), 1034– 1049, (2023).

[19] N. Malik, B. Bookhagen, N. Marwan, J. Kurths, Analysis of spatial and temporal extreme monsoonal rainfall over South Asia using complex networks., Climate Dynamics, 39, 971–987, (2012).

[20] V. Stolbova, P. Martin, B. Bookhagen, N. Marwan, J. Kurths, Topology and seasonal evolution of the network of extreme precipitation over the Indian subcontinent and Sri Lanka., Nonlinear Processes in Geophysics, 21, 901–917, (2014).

[21] G. Di Capua, M. Kretschmer, R. V. Donner, B. Van Den Hurk, R. Vellore, R. Krishnan, D. Coumou, Tropical and mid-latitude teleconnections interacting with the Indian summer monsoon rainfall: a theory-guided causal effect network approach., Earth System Dynamics, 11, 17–34, (2020).

[22] R. H. Kripalani, S. Singh, Large scale aspects of India-China summer monsoon rainfall. Advances in Atmospheric Sciences, 10, 71–84, (1993).

[23] P. H. Hrudya, H.Varikoden, R. Vishnu, A review on the Indian summer monsoon rainfall, variability and its association with ENSO and IOD. Meteorol Atmos Phys 133, 1–14 (2021).

[24] F. Jingfang, J. Meng, J. Ludescher, X. Chen, Y. Ashkenazy, J. Kurths, S. Havlin, H. J. Schellnhuber, Statistical physics approaches to the complex Earth system, Physics Reports 896, 1–84, (2021).

Multi Block Transformer for Malayalam Language Modeling

Rohit T P
Department of Computer Science
SOE, Cochin University of Science and Technology
Kochi, India
tprohit9@gmail.com

Sasi Gopalan
Department of Mathematics
Cochin University of Science and Technology
Kochi, India
sgcusat@gmail.com

Varsha Shaheen
Department of Computer Science
SOE, Cochin University of Science and Technology
Kochi, India
varshashaheen2003@gmail.com

***Abstract*—In this research, we present a novel neural network architecture for natural language generation, specifically designed for Malayalam text. We have adapted the Transformer architecture[1][2] which is commonly used in language modeling and extended it to work in non-Latin languages. To evaluate the effectiveness of our model, we trained it on a large corpus of Malayalam text and fine-tuned the hyperparameters using a grid search. Our model achieved a significant improvement in generating coherent and grammatically correct Malayalam text compared to the state-of-the-art models. The model was able to generate text after just 4000 iterations and was able to effectively generalize the relation between symbols and alphabets of the language within 8000 training iterations. The transformer architecture used proved to be highly efficient in language modeling. Our work highlights the importance of developing new model architectures for text generation in complex and rich languages and opens up new avenues for future research in this area.**

***Index Terms*—Language modeling, Transformer architecture, Attention mechanism, Sequence modeling and transduction, Malayalam text generation, Text tokenization**

I. Introduction

The ability to generate text has become a crucial aspect of modern language processing, with applications in various fields such as machine translation, content generation, and chatbots. Despite significant progress in text generation in English, the problem remains challenging when applied to complex and rich languages such as Malayalam. Malayalam, being a South Indian language, has a large number of symbols and characters, making text generation a complex problem.

Existing solutions for text generation in Malayalam, such as recurrent neural networks (RNNs)[5][6] and encoder-decoder architectures[7][8], have faced limitations in terms of computational efficiency and parallelization. The sequential nature of RNNs, which generate a sequence of hidden states by computing a function of the previous hidden state and the current input, precludes parallelization within training examples and becomes critical at longer sequence lengths. The memory constraints limit batching across examples, further adding to the computational overhead.

This research aims to address these limitations by proposing a new model architecture for text generation in Malayalam, which relies entirely on an attention mechanism[1][2] to draw global dependencies between the input and output. The goal is to significantly improve computational efficiency and parallelization while achieving state-of-the-art results in terms of text generation quality.

II. Related Works

A. Attention Is All You Need

This paper introduced the Transformer architecture, which revolutionized the way NLP models process sequential data. The Transformer uses self-attention mechanisms to capture dependencies between words in a sentence, without relying on recurrent connections. However, the Transformer architecture has mainly been applied to tasks and datasets in English, with limited studies in other languages. In our research paper, we aim to fill this gap by exploring the applicability of the Transformer to the processing and generation of text in Malayalam, a South Indian language. Our work extends the Transformer architecture to handle Malayalam text and demonstrates its ability to generate coherent and grammatically correct sentences in the language.

B. Language Models are Unsupervised Multitask Learners

In this paper, the authors showed that large language models trained on a massive amount of text data can perform well on a variety of NLP tasks without any task-specific fine-tuning. While the approach of training large language models on large datasets has proven successful in various natural language processing tasks, it may lack effectiveness when it comes to Malayalam text generation. Malayalam, being a complex language with unique linguistic characteristics, may require fine-tuning of the language model specifically for this language. The complexity of the language and the diversity of expression may pose challenges for AI in generating coherent text for longer prompts. As a result, more research and fine-tuning may be required to address these limitations in the application of language models for Malayalam text generation.

III. The Architecture

A. Overview

The architecture uses a series of Transformer blocks, each consisting of a self-attention layer and a feedforward layer. The input to each block is first passed through the self-attention layer, where the model attends to different parts of the input sequence to compute a weighted sum. The output of the self-attention layer is then passed through a feedforward layer to obtain the final output of the block. The blocks are followed by a final output projection layer, which maps the output of the blocks to the desired number of tokens in the vocabulary. The architecture is implemented using PyTorch and can be optimized using gradient descent with a learning rate scheduler.

B. Blocks

Each block in the architecture is designed to handle the input sequence in a parallel and efficient manner. By stacking multiple blocks, the model is able to capture more complex dependencies in the input data and improve its overall performance. Each block consists of two sub-layers: a multi-head self-attention layer and a fully connected feed-forward layer. The blocks are optimized using a step learning rate scheduler, which helps to control the learning rate of the model during training and prevent overfitting. By controlling the learning rate, the model can converge to a more accurate solution and produce better results on the task it is designed for.

C. Layers

1. Causal Self-Attention Layer: The layer performs masked self-attention, which means it only attends to elements in the input sequence that are to the left of each element in the sequence. This is achieved by using a trilinear matrix as a mask to set attention scores for elements outside of the causal window to negative infinity. The masked self-attention is followed by a linear projection to obtain the final layer output.

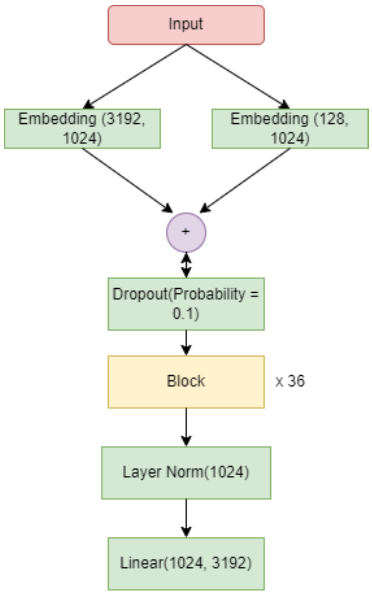


Fig. 1. Layer Diagram of the Transformer Model

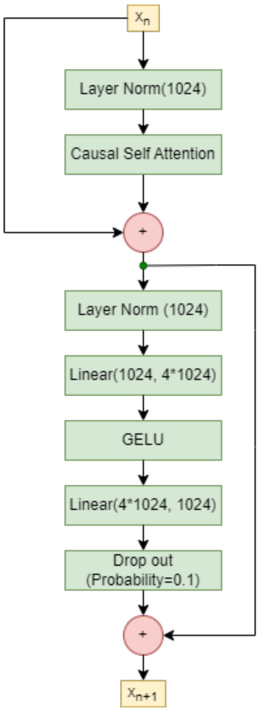


Fig. 2. Expanded Diagram of Each Individual Block

The input to the layer is an input sequence of shape $B \times T \times C$, where B is the batch size, T is the sequence length, and C is the number of hidden units (also referred to as the embedding dimensionality). The layer has two linear projections: the first is used to obtain the query, key, and value matrices for each head, and the second is used to obtain the final output after masked self-attention. The layer also contains two dropout layers for regularization.

The query, key, and value matrices for each head are obtained by applying the first linear projection to the input sequence. Let the output of this projection be Q, K, and V, respectively. The shape of Q, K, and V are $B \times T \times 3C$, and they are then reshaped to $B \times n_h \times T \times \frac{C}{n_h}$, where n_h is the number of heads.

The masked self-attention is performed by computing an attention score matrix A as follows:

$$A = \frac{QK^T}{\sqrt{d_k}} \tag{1}$$

where d_k is the dimension of the key vector for each head. Using a trilinear matrix, the attention score matrix is masked by setting elements outside the causal window to negative infinity. The attention scores are then normalized using the softmax function, resulting in the attention probability distribution:

$$P = \text{softmax}(A) \tag{2}$$

The final attention-weighted representation is obtained by computing the weighted sum of the value matrix V using the attention probability distribution P:

$$Y = P \cdot V \tag{3}$$

The final output of the layer is obtained by applying the second linear projection and applying dropout for regularization:

$$Z = \text{Dropout}(\text{Linear}(Y)) \tag{4}$$

2.Feed-forward layer: This is a two-layer fully connected neural network with a GELU activation function in between. The layer takes an input tensor and applies a linear transformation to it, followed by the GELU activation function, and then another linear transformation. The output of the feed-forward layer is then passed through dropout before being returned as the final result. The feed-forward layer is designed to provide a non-linearity to the input tensor and to increase the expressiveness of the model so as to increase the capacity of the model and to capture complex relationships between input and output.

$$\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{5}$$

where $\mathbf{W}$, $\mathbf{b}$, and $\sigma(\cdot)$ are the weights, biases, and activation functions respectively.

IV. Training

The model was trained on a corpus of Malayalam text data collected from Wikipedia. The data consists of mostly articles and use a formal Malayalam dialect in general. The model performed self-supervised on the text.

A. Data Preprocessing

The data collected from Wikipedia was cleaned and any nonprintable characters were removed. To protect privacy any traceable personal information like email addresses, phone numbers, etc were replaced with mask tokens. The data set was tokenized by mapping each character to a 16-bit integer. The mapping was done using the Unicode value of each character and shifting accordingly to make the range continuous. The character level tokenization here was preferred to reduce the number of tokens required and thereby reducing the size and complexity of the model.

B. Optimizer

We used AdamW optimizer with $\beta1 = 0.9, \beta2 = 0.95$, and learning rate = 3e-4. These specific values were derived by statistical analysis on multiple test runs and extrapolating.

C. Schedule

Our model was trained on a Tesla K80 GPU for 12 hrs with each training step duration of 100ms.

V. Result

The model was able to learn the relation between letters and symbols and also the proper use of spaces and punctuation. It was able to generate coherent completions for a given prompt as well as produce consistent outputs within its embedding size.

TABLE I
Training Progress

Iteration	***Train loss***	***Test loss***
0	8.4496	8.9434
500	5.5241	5.6930
1000	3.2999	4.8662
1500	2.3141	2.8410
2000	1.3355	1.8941
2500	1.3779	4.0960
3000	1.3517	1.4768
3500	1.3013	1.7277
4000	1.2914	1.2289
4500	1.1925	1.4992

A. Sample Outputs

With token limit set to 36 (for short / few word completions)

- Prompt: വിദ്യാർത്ഥി
  Generated: വിദ്യാർത്ഥികളും പ്രാധാന്യം കാണുന്നു.
- Prompt: പ്രതിഷേധം
  Generated: പ്രതിഷേധം നടത്തിയിരുന്നു.
- Prompt: പ്രാദേശിക
  Generated: പ്രാദേശിക കാലാവസ്ഥയിൽ കുറവായിരിക്കും
- Prompt: പൂക്കൾ
  Generated: പൂക്കൾ വളരെ കുറവായും കാണും

With the token limit set to 108 (for long/full sentence completions)

- Prompt: ഈ പദ്ധതി
  Generated: ഈ പദ്ധതി വിദ്യാഭ്യാസ സ്ഥാപനം നടത്തുകയും

അദ്ദേഹത്തിന്റെ പ്രധാന കേന്ദ്രങ്ങളെ അവതരിപ്പിക്കുകയും ചെയ്യുന്നു

- Prompt: പ്രവിശ്യയുടെ പ്രസിദ്ധമായ
  Generated: പ്രവിശ്യയുടെ പ്രസിദ്ധമായ പ്രസിദ്ധീകരണം എന്ന പുസ്തകത്തിൽ മികച്ച പല പ്രാധാന്യം കേന്ദ്രങ്ങളും കേന്ദ്ര
- Prompt: കേരളത്തിലെ
  Generated: കേരളത്തിലെ ജലസ്രോതസ്സുകളെ അടിസ്ഥാനമാക്കി ജലസംഭരണത്തിന്റെ ആത്മകഥയാണ് പാണ്ടം.

B. Comparison With State Of The Art

The current state-of-the-art in language modeling is GPT-3 by Open AI. This model was shown to be very effective in generalizing language modeling tasks. The major drawback of the GPT models is the use of subword-level tokenization. Even though this approach is adequate in modeling Latin and Latin-based languages like English it becomes limiting when the model tries to learn languages that use complex arrangements of letters and symbols. In languages like Malayalam, the meaning is expressed using the combination of both letters and symbols and sub-word level transformers fail to learn the relation between symbols and letters and how the arrangement can be changed to form different words. Our architecture overcomes this problem by using a character-level model. Our model was able to learn the inter-character relation and generalize it to generate as well as infer information from unseen words. This approach opens the model to learn more in-depth characteristics of the given language and generalize.

VI. Conclusion

In conclusion, the AI showed a remarkable capability in generating text in the Malayalam language, delivering grammatically correct outputs for short prompts with up to 36 tokens. The continuous decrease of both training and validation loss confirms the model's generalizability. However, it was observed that the AI faced difficulties in preserving context for longer text generation. Further exploration is needed to enhance the AI's coherence in generating longer sentences.

It is noteworthy that, despite the Transformer architecture's remarkable success in text generation for various languages, more efforts are required to optimize the model for the specific linguistic traits of Malayalam. Additionally, due to the intricacy of the language, the creation of large annotated datasets for training and model fine-tuning is crucial in enhancing performance.

This study provides a comprehensive understanding of the application of the Transformer architecture in generating text in the Malayalam language and highlights the significance of further research in this field. The results have the potential to be applied to a range of real-world applications, such as machine translation, text-to-speech synthesis, and text classification in Malayalam.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Gomez, Aidan N. Gomez Łukasz Kaiser, Illia Polosukhin(2017) Attention Is All You Need. In Advances in Neural Information Processing Systems (pp. 5998-6008).

[2] S. Saravanan and K. Sudha, "GPT-3 Powered System for Content Generation and Transformation," 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), Sonepat, India, 2022, pp. 514-519, doi: 10.1109/CCiCT56684.2022.00096.

[3] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI.

[4] R. Sunil, N. Manohar, V. Jayan and K. G. Sulochana, "Development of Malayalam Text Generator for translation from English," 2011 Annual IEEE India Conference, Hyderabad, India, 2011, pp. 1-6, doi: 10.1109/INDCON.2011.6139398.

[5] K. Souri, A., El Maazouzi, Z., Al Achhab, M., El Mohajir, B.E. (2018). Arabic Text Generation Using Recurrent Neural Networks. In: Tabii, Y., Lazaar, M., Al Achhab, M., Enneya, N. (eds) Big Data, Cloud and Applications. BDCA 2018. Communications in Computer and Information Science, vol 872. Springer, Cham. https://doi.org/10.1007/978-3-319-96292-.

[6] Milanova, I., Sarvanoska, K., Srbinoski, V., Gjoreski, H. (2019). Automatic Text Generation in Macedonian Using Recurrent Neural Networks. In: Gievska, S., Madjarov, G. (eds) ICT Innovations 2019. Big Data Processing and Mining. ICT Innovations 2019. Communications in Computer and Information Science, vol 1110. Springer, Cham. https://doi.org/10.1007/978-3-030-33110-8_1.

[7] X. Zhang, Y. Li, X. Peng, X. Qiao, H. Zhang and W. Lu, "Correlation Encoder-Decoder Model for Text Generation," 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 2022, pp. 1-7, doi: 10.1109/IJCNN55064.2022.9891880.

[8] C. Zhou, J. Shang, J. Zhang, Q. Li and D. Hu, "Topic-Attentive Encoder-Decoder with Pre-Trained Language Model for Keyphrase Generation," 2021 IEEE International Conference on Data Mining (ICDM), Auckland, New Zealand, 2021, pp. 1529-1534, doi: 10.1109/ICDM51629.2021.00200.

Natural Language Interface in Malayalam for SQL Databases

Anjali Augestin
Dept. of Information Technology
Government Engineering College Idukki
Painavu, Idukki
anjaliaugestin@gmail.com

Geethu Subramanyan C
Dept. of Information Technology
Government Engineering College Idukki
Painavu, Idukki
geethuc2002@gmail.com

Kavyadas
Dept. of Information Technology
Government Engineering College Idukki
Painavu, Idukki
kavyaalungal691@gmail.com

Ummu Habeeba P P
Dept. of Information Technology
Government Engineering College Idukki
Painavu, Idukki
ummuhabeeba467@gmail.com

Visakh R
Dept. of Information Technology
Government Engineering College Idukki
Painavu, Idukki
visakhrnarayanan@gmail.com

***Abstract*—Natural Language Processing (NLP) is the branch of Artificial Intelligence that gives machines the ability to read, understand and derive meaning from human natural language. Every NLP system consists of a Natural Language Interface (NLI) which is a user interface in which the user and system communicate through human natural language. The proposed system is an NLI for querying Structured Query Language (SQL) databases in the Malayalam language. In Kerala, the native language is Malayalam and most of the government offices, colleges, and e-governance application uses relational databases. Therefore, to manipulate data from such databases easily, non-technical people who are more confident in their natural language need a solution in which they can interact with the machine in their natural language and generate a valid SQL query. The scope of the proposed system is restricted to Data Definition Language (DDL) commands like CREATE, ALTER, and DROP, and Data Manipulation Language (DML) commands like INSERT, UPDATE, and SELECT. Due to the complexity of semantics in the Malayalam language, translating Malayalam queries into complex SQL constructs such as aggregate functions, joins and nested queries can be challenging. The proposed system finds application in schools, colleges, and government institutions where the data needs to be stored, retrieved, and updated efficiently. In the future, the system can be implemented with other complex queries and can be extended to NoSQL databases.**

***Index Terms*—Natural Language Processing, Artificial Intelligence, Sentiment Analysis, Text Extraction, Tokenization, Lemmatization, Natural Language Toolkit.**

I. Introduction

Data forms the heart of business logic in any organization. Data need to be stored, retrieved, and accessed efficiently for every process. At the same time, working with data stored in a database requires special kinds of technical skills like SQL. To a non-professional or a person from a non-technical background, it would be a nightmare to construct complex queries using SQL functions and keywords. Compared to other languages like European and Asian languages, the growth of Malayalam NLP systems is sluggish due to the semantic complexity in the Malayalam language. So it is essential to develop an NLI in Malayalam which enables people to interact with machines in their natural language. NLI and NLP systems play a pivotal role in realizing human-machine interaction.

The proposed system is a web interface for querying SQL databases. SQL is a special language used to retrieve data and modify data from the database. It consists of various complex functions, aggregate functions, ranking functions, etc. However, people who are not professionally qualified with SQL face failure in data fetching by using SQL queries. In this data-centric era, database technologies are having a dominant influence on the usage of technology.To retrieve accurate data from databases, users must need a good understanding of database query language. However, there is a problem with communication between non-technical users and database management systems. Moreover, designing a good database requires special training for maintaining the database which is time-consuming. This research work focuses on developing a Malayalam Language Interface that will convert natural language queries into formal SQL queries. Developing a computational system that enables people to interact with a database in their natural language - the language in which they are more confident would be convenient and time-saving. The proposed system accepts user queries in the Malayalam language and is processed in a linguistic component, after which it is converted into a valid SQL query and is executed in the database component. The result is fetched from the database and is displayed to the user in the default view.

II. Literature Survey

NLP deals with the nuances of human languages through a machine's reading comprehension. It enables computers to understand human language and derive meaning from text or speech. There are a few notable research works in the recent literature in this direction.

Duneesha Suloshini et al. [1] developed a NLI in Sinhala for producing valid SQL queries from Sinhala natural language queries. The model accepts the queries in Sinhala, then generates the corresponding SQL query, and after execution, the result is displayed back in the same language. It is a prototype and initially works for only one table. The proposed system is domain-specific and depends on the Sinhala lexicon, which contains only a limited number of words. The whole process is divided into a linguistic component and a database component. The query entered by the user initially goes into the linguistic component. Then the sentence is divided into tokens, and then the tokens are lemmatized. Then each token is tagged using the Sinhala lexicon. Here the tokens will be tagged as table name, column name, command, or conditional operator, and an intermediate query is generated with the tagged words. This intermediate logical query will be provided to the database component, and all items identified will be combined to form a SQL query, which will then be executed.

Minhazul Arefin et al. [2] developed an NLI for SQL databases in which the system accepts queries in the English Language. For the conversion, they utilized lowercase conversion, removing escaped words, tokenization, PoS tagging [23], the Jaro-Winkler [18] matching algorithm, and the Naive Bayes [16] algorithm. The key idea of this work is to use Natural Language Tool Kit [15] (NLTK) and machine learning to convert English queries to valid SQL queries. In the text pre-processing phase, the natural language query is converted into lowercase, then tokenized. After that escape words are removed then the tokens are tagged using Parts-of-Speech (PoS) tagging. Then word similarity is identified using a synonym list for noun tags created using WordNet [19]. They implemented these steps using NLTK, which provides various libraries for text pre-processing and a lot of datasets. In the next phase attributes, tables, operations, and conditions are extracted and a query is generated. The model is designed independently of the database's domain, language, and model, which increases performance accuracy. Also, the system works better for the conversion of natural language queries into SQL queries. The proposed methodology of the system is shown in fig 1. The entered English query is pre-processed. The structured data from the pre-processing stage is analyzed to extract attributes, tables, operations, and conditions. For attribute and operation extraction Jaro-Winkler string matching algorithm is used, which looks for similar words in a database and the query. Multinomial Naive Bayes classifiers are used to find the table and provide the table name with high probability. Conditions are extracted using decision trees.

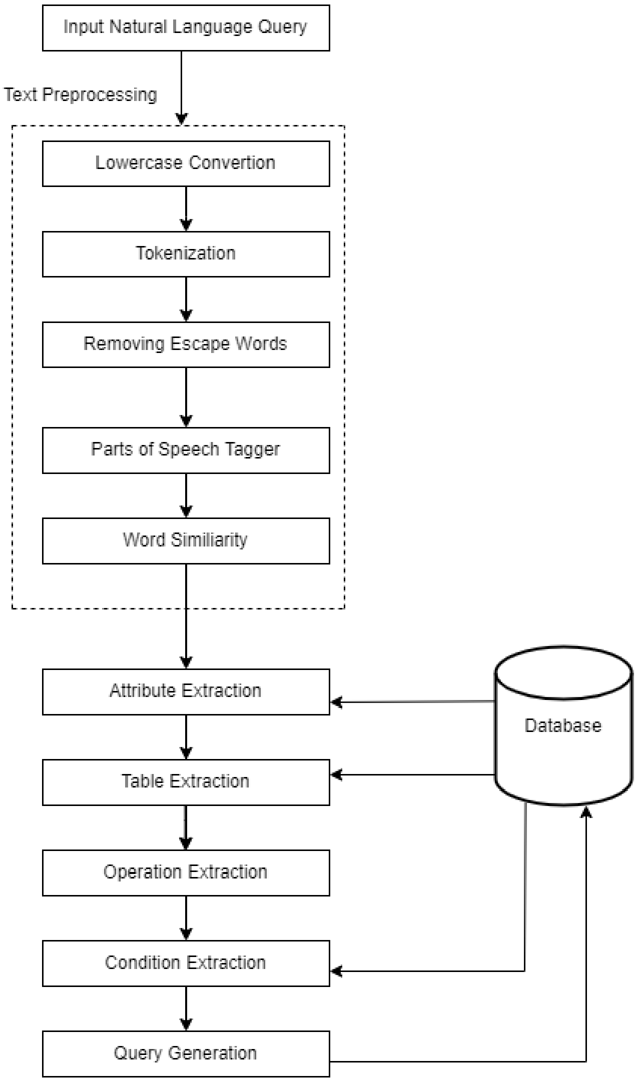


Fig. 1. Block diagram of Natural language to SQL Conversion using Machine Learning

Sadullah Karimi et al. [3] presented a model called Afghan Language Query and Control Interface (ALQCI) based on the NLIDB Approach. For interacting with the database, the Afghan language is used. This system is a solution for Persian-speaking people to interact with the database without any struggle. This system has better user-friendly User Interfaces (UI). The user input a query in the Persian language and this technology will generate a valid SQL query. To turn a natural language question into a SQL query, the word mapping approach is utilized. The SQL is built through semantic analysis. The generated SQL query is further used to fetch the required data from the database. The proposed system consists of three main phases: lexical analysis, syntax analysis, and semantic analysis. The input query is initially passed into the lexical analysis phase. In this phase case folding, removing punctuations, and tokenization are performed. Then the sentence is syntactically analyzed by removing stop words, parts of speech tagging, stemming, and lemmatization. The analyzed text is semantically analyzed using a dictionary mapper which represents the meaning of the words. From the structured data, a valid SQL query is generated.

George Obaido et al. [4] developed TalkSQL, a voice-based tool that generates SQL queries from natural languages and provides feedback to the user. The proposed Natural Language Interface to Databases (NLIBD) system framework evolved to include Create, Read, Update, Delete (CRUD) operations which encourages users to create, retrieve, modify, and delete data from the database. The system can also perform error checking and it provides response for simple CRUD operations. The system determines SQL operations and provides feedback using regular expressions. TalkSQL finds applications in Question Answering Systems, Learning Aids, Intelligent Tutoring Systems, Assistive Technology Systems, and Improved SQL Comprehension. TalkSQL was able to recognize simple queries that do not comprise balanced parentheses.

Zeinab Borhanifard et al. [5] proposed a Natural Language Understanding (NLU) model with a specific named entity recognizer for shopping using a Bidirectional Encoder Representations from Transformers [22] (BERT) transformer. BERT is a machine learning platform for NLP techniques in which the model is initially trained using wiki data and then finely pre-tuned using relevant training data sets. Due to the unavailability of published data sets for Persian online shopping, they implemented two methods for generating training data: fully-simulated and semi-simulated methods, which helped to address the lack of data . In the fully-simulated method, they created a dataset based on the hybrid of rule-based and template-based generation. In the semi-simulated method, the dataset is created using some original data and simulated data, and the language generation part is done by a human which improves the quality of the data. Experiments show that a combination of the semi-simulated and fully-simulated datasets can improve the results and accuracy. Intially, they trained the proposed NLU model for simulated datasets and then for a combination of simulated and semi-simulated datasets. Then the model is finely tuned using PasBert. They trained the model with the simulated data set and observed that expanding the number of simulated conversations does not certainly improve the precision. They added some dialogues from semi-simulated datasets and trained the model with it. Training the model with the merged dataset increased the results by 5.4 percent in the F1 measure.

Wanbo Li et al. [6] developed an improved version of the sign language recognition model, combining Convolutional Neural Network [20] (CNN) and Long Short-Term Memory [17] (LSTM) neural network. Sign language recognition and research based on this enables people to understand sign language, and also allows deaf people to recognize what is said by others. The process which enables the understanding of sign language is called sign language recognition and translation, while the process which makes disabled people understand the natural language is called sign language generation. The development of both sign language recognition and generation is significant for people with hearing impairment. Compared to other models the proposed model is also designed for sign language generation. This model uses a PyQt - designed Graphical User Interface (GUI). This system has four main features and they are user authentication, sign language recognition, sign language generation, and remote communication services. Users can input their account number and the password for the people that has been registered an account, also login into the user interface of the system, and if the user chooses sign language recognition, then the system will automatically activates and open the camera to record the movements of the hand. For model training the system uses American sign language and Arabic numeral gestures. The results are displayed in English letters or will be a number of 0-9. The system also enables the user to select sign language generation, by the recognition of either voice or text of the user. If the user gives voice input, then the voice is converted to corresponding English, and then the recognized sign language pictures are converted into a video for displaying to the user. The final results shows that the sign language recognition rate is 95.52% compared with other algorithms. The accuracy of sign language generation i.e., American sign language and Arabic numerals is 90.3%.

Anis Cherid et al. [7] created an NLU system by listing various rules manually in a knowledge base. The current system is very expensive as it make use of natural language to execute office application functions. They propose to build an NLU system by analyzing the text contained in the office application user manual with the help of NLP. They developed different variety of simple rules using the text which is given in the user manual, and they are specifically built to NLP. In future, we can develop system to support the automatic creation of variety of complex rules by analysing the commercially available user manuals. The proposed system facilitates essential operations to execute the application functions executed on an office application prototype.

Hanane Bais and Mustapha Machkour [8] developed an Arabic language interface for XML data (ANLQ). The model receives queries in Arabic and then transforms the Arabic Natural Language Question into a Database Language Query (DBLQ) using a specially developed linguistic operation. The technology is intended for databases in a certain subject and is based on the Arabic language. The entire procedure is separated into language and database knowledge components. The user's inquiry is first processed by the linguistic components. The ANLQ is processed in the linguistic component utilizing morphological, syntactic, and semantic processes. The logical representation is converted into a database logic query in the Database knowledge component. The database logic query is generated in three phases. Each phase focuses on a different portion of the DBLQ. The system treats the section that matches the characteristics to create the selection clause in the first one. The FROM clause is determined in the second stage by utilizing the table names section. Finally, it receives the search criteria from the IXLQ in order to have the condition clause. The DBLQ is formed by combining the outcomes of these processes. Following each step, we check to see if the items returned by a logical query are still in the catalog. Instead, it employs a custom table that stores tables and attribute synonyms. Using this table, a natural query can be written without utilizing the precise table and attribute name(s).

G. Arora [9] presented iNLTK which is an open-source NLP library that includes pre-trained language models that cover Textual Similarity, Data Augmentation, Word Embeddings, Tokenization, Sentence Embeddings, and Text Creation in nearly thirteen Indian languages. We may employ pre-trained models from iNLTK for text categorization on data, and this work significantly exceeds prior years' results. The work also shown that combining pre-trained models and data augmentation from iNLTK, we can obtain more than 95% of the performance of earlier efforts while using less than 10% of the training data. iNLTK is presently widely utilised by the community, with over 40,000 downloads, almost 600 stars, and 100 forks on GitHub.

Ursin Brunner and Kurt Stockinger [10] designed a system called ValueNet, which was the first system to convert a natural language query to a proper SQL query. The central concept of this paper is to use the information in the base data rather than just the metadata information. Here, a neural-based model is used to build a SQL query. By using the execution accuracy metric, the system is estimated. The system is evaluated using the spider dataset [21]. The main steps used in this approach are Value extraction, Value candidate generation, Value candidate validation, and Value candidate encoding.The experimental assessment shows that ValueNet lite and ValueNet achieve state-of-the-art results for translating Natural Language (NL) to SQL of 67% and 62%, respectively.

Rashi Kumar and Vineet Sahula [11] created an intelligent approach to NLP for Indian languages. The goal is to find context within a written text. This system employs the LSTM technique. For testing purposes, a parallel catalog of Sanskrit-Hindi has been created. The information needed to build the parallel corpus was collected from the Madhya Pradesh Department of Public Information. The lack of parallel corpus for the language Hindi and Sanskrit and comprehending the structure of the language was challenging. They have updated and supplemented the architecture in this study by including a token at the beginning of a phrase identifying the target language. The design employs a multilayered LSTM to delineate input information succession to a fixed dimensionality vector, followed by the employment of a multi-layered LSTM to unravel the objective grouping from the vector.

Madhuri Kuthadi et al. [12] developed a language user interface for generating SQL queries from natural language. The model accepts the queries in natural language, then generates the corresponding SQL query and after execution, the result is displayed back in the same language. The key idea of the paper is regarding the creation of an interface to the database component. The pattern-Matching system, Semantic Grammar Systems, and Intermediate Representation Languages are the main techniques that are used to develop the NLIDB.

Yawar Abass Mir et al. [13] used NLP to create a Common Database Interface for Relational Databases. The model will take inquiries written in simple English. The main idea is to turn a statement into a logical query language, which is then translated into a standard database query language, such as SQL. The NLP-based Common Database Interface for Relational Databases first converts a simple English language inquiry into an intermediate suitable query. The intermediate logical query describes the user's query in terms of high-quality world conceptions that are independent to the database structure. The logical query is then converted into an expression in the database's query language and checked against it. The system's overall strategy is to build an employee database that will store employee information. The system identifies the operation, such as select, update, delete, or create, using a standard database interface. By collecting table and column information from input English texts, tokens should be appropriately mapped to database keywords. The SQL query is built by translating the input query to the values in the database tables.

Alaka Das et al. [14] developed an NLIDB for generating SQL queries from a natural language. The model accepts the queries in natural language, then generates the corresponding SQL query and after execution, the result is displayed back in the same language. The key idea behind the paper is to use NLTK for tokenization and dependency parsing. The system is domain-independent and database independent.

III. Proposed System

The proposed system is a web interface for querying databases in Malayalam that uses NLP techniques and machine learning algorithms for the processing of natural language queries to convert them into valid SQL queries. The system consists of four functional modules. The first one is a web interface that enables users to enter the query in Malayalam. The SQL query corresponding to the Malayalam query and the results are displayed in the interface. The second module is the linguistic component. The input query is passed into the linguistic component where the text is pre-processed and converted into structured data. The data is tokenized, lemmatized, syntactically, and semantically analyzed with the NLTK framework. NLTK [15] is a python framework that provides various libraries for data processing steps in NLP and provides a lot of datasets. The processed data is passed to the query generator module which generates the valid SQL query from the data from the previous step. Finally, the query executor module performs the SQL query in the database, fetches the results from the database, and displays the results to the users.

The system is an NLI in Malayalam for querying SQL databases. The systems involve the following method/modules to be designed:

A. Web Interface

Web interface that facilitates interaction between non-technical users and other modules. The user interface contains:

- A text area for entering Malayalam natural language questions
- A reset button to re-enter the question
- A button that allows the user to construct a SQL query
- Text box which shows the produced SQL query
- A button for generating query results
- Button to clear all the field

B. Linguistic Component

The unstructured query entered by the user needs to be structured so that the algorithm can easily analyze the text. The linguistic component converts the user text into machine-understandable form. The text is tokenized, lemmatized, syntactically analyzed, and mapped to the lexicon. The important entities identified are classified into tables, attributes, commands, etc.

C. Query Generator

An intermediate logical query is generated from the input. The important entities are identified and classified. It is the intermediate representation of the valid SQL query which is fed to the query executor. Intermediate representation enables easy and more efficient mapping to the SQL query. Further, it transforms the intermediate query into an SQL query.

D. Query Executor

The query executor module executes the SQL query and fetches results from the database. The result is displayed back to the user through the UI. The overview of the system with all the modules is shown in fig 2.

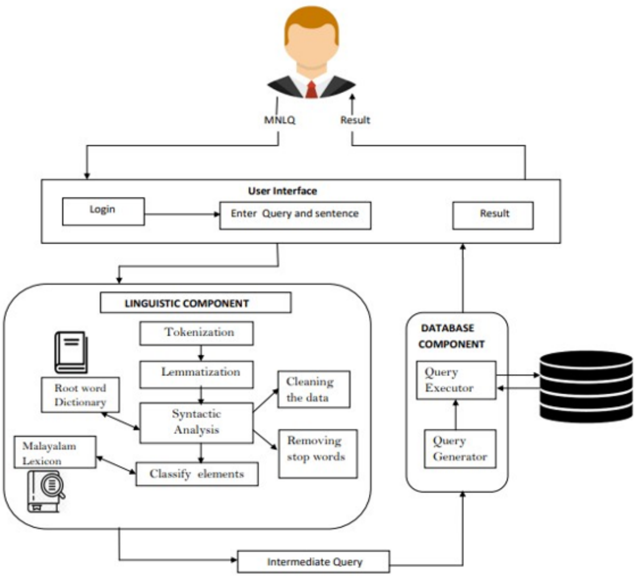
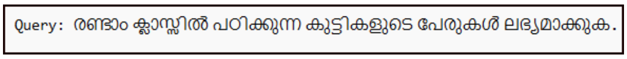


Fig. 2. Block diagram illustrating functional modules of the system

IV. Experimental Setup

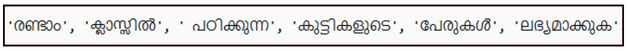
The system utilizes NLTK [15] framework for syntactic and semantic analysis of the input query . For POS tagging [23], the system uses a tagged malyalam corpus from [24]. This malayalam corpus dataset is in the form of a dictionary comprising of word and its corresponding tag. The dataset consist of 287588 words and 36 unique tags.

A sample input query to the system and the expected output are given below.



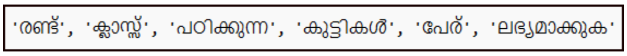
```
Query: രണ്ടാം ക്ലാസ്സിൽ പഠിക്കുന്ന കുട്ടികളുടെ പേരുകൾ ലഭ്യമാക്കുക.
```

After tokenization the result will be:



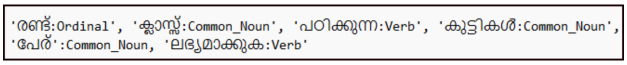
```
'രണ്ടാം', 'ക്ലാസ്സിൽ', ' പഠിക്കുന്ന', 'കുട്ടികളുടെ', 'പേരുകൾ', 'ലഭ്യമാക്കുക'
```

The tokenized words are then mapped to their base words:



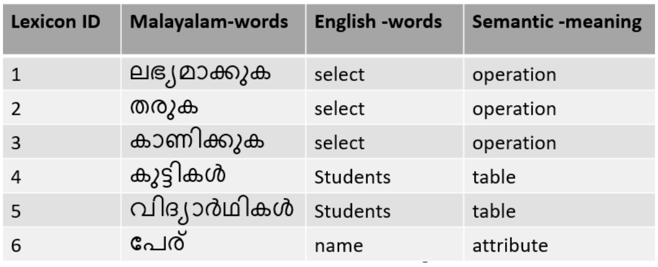
```
'രണ്ട്', 'ക്ലാസ്സ്', 'പഠിക്കുന്ന', 'കുട്ടികൾ', 'പേര്', 'ലഭ്യമാക്കുക'
```

The stream of tokens is then fed to a tagger. The tagger outputs a dictionary that resembles the following:



```
'രണ്ട്:Ordinal', 'ക്ലാസ്സ്:Common_Noun', 'പഠിക്കുന്ന:Verb', 'കുട്ടികൾ:Common_Noun',
'പേര്':Common_Noun, 'ലഭ്യമാക്കുക:Verb'
```

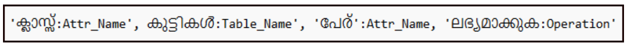
The database is scanned and all the metadata are written into a lexicon file. The lexicon captures all possible regional variants of Malayalam words related to the domain. The structure of a lexicon is shown in Fig 3. The operation, table name, attribute name(s) and conditional operator (if any) in the input query are identified by a lookup operation into the lexicon.



Lexicon ID	Malayalam-words	English -words	Semantic -meaning
1	ലഭ്യമാക്കുക	select	operation
2	തരുക	select	operation
3	കാണിക്കുക	select	operation
4	കുട്ടികൾ	Students	table
5	വിദ്യാർഥികൾ	Students	table
6	പേര്	name	attribute

Fig. 3. Malayalam Lexicon

The lexicon lookup operation identifies the operation, table name, and attribute name(s) in the input query as follows:



```
'ക്ലാസ്സ്:Attr_Name', കുട്ടികൾ:Table_Name', 'പേര്':Attr_Name, 'ലഭ്യമാക്കുക:Operation'
```

’SELECT’ is identified as the operation. After the identification of the operation, the query is examined to identify the conditional part. In this example, the system identifies ’student’ as the table name and ’name’ as the attribute. Then, an intermediate query is formed by utilizing the properties of a valid SQL query.

Expected output: The above query will generate the following valid SQL query:
SELECT Name FROM students WHERE class=’two’;

The expected output of the input query is shown in Table I.

TABLE I
OUTPUT OF THE INPUT QUERY

Name
Anjali
Anu
Binoy
Charles

V. Scope and Challenges

The scope is restricted to DDL commands like CREATE, ALTER, DROP, and DML commands like SELECT, INSERT, UPDATE, and DELETE and focuses our domain on student databases in educational institutions. Due to the complexity of semantics in the Malayalam language, translating Malayalam queries into complex SQL constructs such as aggregate functions, joins and nested queries can be challenging. So we are only dealing with the above-mentioned commands. Furthermore, the system's operation is dependent on a restricted Malayalam lexicon.

VI. Conclusion

This research work aims at implementing a natural language interface in Malayalam for querying SQL databases. The input query is processed using various NLP techniques. The system enables the interaction of ordinary people with the databases. Most of the offices and institutions in Kerala state still rely on relational databases for storing data. Keralalites, who are more confident in their natural language can thus easily retrieve and modify data from databases. The proposed system provides a natural language interface that accepts Malayalam queries as input and provides a convenient and user-friendly platform for the users. The proposed system can be extended in the future by incorporating voice-based interfaces.

References

[1] Duneesha Suloshini Peduru Hewa and Cassim Farook, *"A Sinhala natural language interface for querying databases using natural language processing,"* 21st International Conference on Advances in ICT for Emerging Regions (ICTer 2021): 213 - 218.,02nd - 03rd December 2021, SriLanka.

[2] Minhazul Arefin, Kazi Mojammel Hossen and Mohammed Nasir Uddin, *"Natural language query to SQL conversion using machine learning approach,"* 2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI), 18-19 December, Dhaka.

[3] Sadullah Karimi, Annajiat Alim Rasel, Matin Saad Abdullah, *"Natural language query and control interface for database using afghan language,"* 2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), 2022.

[4] George Obaido, Abejide Ade-Ibijola, Abejide Ade-Ibijola *"TalkSQL: A tool for the synthesis of SQL queries from verbal specifications,"* 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC) 2020.

[5] Zeinab Borhanifard, Hossein Basafa, Seyedeh Zahra Razavi, *"Persian language understanding in task-oriented dialogue system for online shopping,"* 11th International Conference on Information and Knowledge Technology (IKT) December 22-23, 2020; Shahid Beheshti University - Tehran, Iran.

[6] Wanbo Li, Hang Pu, Ruijuan Wang, *"Sign language recognition based on computer sign language recognition based on computer vision,"* IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), June28-30,2021, Dalian,China.

[7] Anis Cherid, Edi Winarko, Mujiono Sadikin, Afiyati Reno, *"Building natural language understanding system from user manual to execute office application functions,"* 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI),28 October 2021, Jakarta - Indonesia.

[8] Hanane Bais and Mustapha Machkour, *"Arabic language interface for XML databases,"* 2019 4th World Conference on Complex Systems (WCCS), 2019, pp. 1-5, doi: 10.1109/ICoCS.2019.8930803.

[9] G. Arora, *"iNLTK: Natural language toolkit for indic languages,"* in Proceedings of Second Workshop for NLP Open Source Software(NLP-OSS). Online: Association for Computational liguistics, Nov 2020, pp 66-71.

[10] Ursin Brunner and Kurt Stockinger, *"ValueNet: A natural language-to-SQL system that learns from database information,"* 2021 IEEE 37th International Conference on Data Engineering (ICDE).

[11] Rashi Kumar and Vineet Sahula, *"Intelligent approaches for natural language processing for indic languages,"* 2021 IEEE International Symposium on Smart Electronic Systems (iSES).

[12] Madhuri Kuthadi, *"Natural language interface to databases,"* International Journal For Technological Research In Engineering Volume 5, Issue 2, October-2017.

[13] Yawar Abass Mir, Zahid Zahoor Koul and Syed Irfan *"Common database interface with NLP,"* IJCSMC, Vol. 6, Issue. 6, June 2017, pg.195 – 199.

[14] Alaka Das and Rakesh Chandra Balabantaray, *"MyNLIDB: A natural language interface to database,"* 2019 International Conference on Information Technology (ICIT), 2019.

[15] Bird, Steven, Edward Loper and Ewan Klein (2009). *"Natural Language Processing with Python,"* O'Reilly Media Inc.

[16] Lewis, D. (1998) *"Naive Bayes at forty: the independence assumption in information retrieval,"*, Machine Learning: ECML-98, Proceedings of the 10th European Conference on Machine Learning, Chemnitz, Germany (pp. 4–15).

[17] Hochreiter, S. & Schmidhuber, J"urgen, 1997. *"Long short-term memory,"* Neural computation, 9(8), pp.1735–1780.

[18] Winkler, William. (1990). *"String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage,"*, Proceedings of the Section on Survey Research Methods.

[19] George A. Miller (1995),*"WordNet: A Lexical Database for English,"* Communications of the ACM Vol. 38, No. 11: 39-41.

[20] M.V. Valueva, N.N. Nagornov, P.A. Lyakhov, G.V. Valuev, N.I. Chervyakov, *"Application of the residue number system to reduce hardware costs of the convolutional neural network implementation,"* Mathematics and Computers in Simulation, Volume 177, 2020, Pages 232-243, ISSN 0378-4754.

[21] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. *"Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task.,"* Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova,*"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,"* 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[23] Eric Brill. 1992. *"A simple rule-based part of speech tagger,"* Proceedings of the third conference on Applied natural language processing (ANLC '92). Association for Computational Linguistics, Stroudsburg, PA, USA, 152-155.

[24] Ajees A P and Sumam Mary Idicula, *"A PoS Tagger for Malayalam using conditional random forest,"* International journal of applied engineering research, ISSN 0973-4562 volume 13, number 3 (2018) Spl.

Strategies of power allocation for QoS parameters in NOMA for 5G wireless communications

Lekshmi Nair M
Department of ECE, SCMS School of Engineering and Technology
lekshmi8887@gmail.com

Neelakantan P. C
Muthoot Institute of Technology and science
npcnitc@gmail.com

***Abstract*— In the new era of wireless communication, the number of users is significantly increasing on a daily basis. Because of the fast-growing traffic, network capacity has to be increased drastically. The conventional multiple access techniques like OMA will not be sufficient alone to satisfy the demands of the new standards. Non-Orthogonal Multiple Access (NOMA) is a hopeful candidate for future generation mobile communication, where multiuser channel capacity can be increased by sharing same time -frequency resources. In this paper BER analysis of NOMA in AWGN channel is studied with dynamic power allocation. BER analysis shows that allocation of power contributes a significant role in the performance of NOMA.**

Keywords— ***BER, OMA, NOMA, superposition coding, successive interference cancellation, AWGN channels.***

I. Introduction

The fundamental objective of cellular system design is increased channel capacity with sufficient standard of quality of service. The number of users who need access to spectrum is drastically increasing on a daily basis. In such a case, sharing is needed to increase the capacity since spectrum is a limited resource. The available bandwidth is to be used at the same time by multiple users. In doing so system must make sure that there should not be any degradation to the quality of service to users. Multiple access techniques allow a large number of users to achieve the spectrum allocation in most efficient manner [1],[2]. Each generation of cellular technology is witnessing various multiple access techniques, which contribute a major share in escalating the capacity and efficiency of the system.

The key idea of multiple access techniques is to allow multiple users to share the resources. Several multiple access techniques are introduced in wireless communications. The important features of multiple access schemes are a) Maximise the spectral efficiency b) Handle multiple users with zero mutual interference c) Better handover between cells etc. Each generation of mobile communication adopted its own method of multiple access techniques. Let us have a look into the details of such schemes used in each generation.

Previous generation of cellular networks have adopted one or more of following multiple access methods for sharing the resources among users. Different categories of multiple access techniques are Frequency division multiple access (FDMA), Time division multiple access (TDMA), Spread Spectrum multiple access (SSMA), Orthogonal frequency division multiple access (OFDMA), Space division Multiple Access Techniques (SDMA) etc. This section gives a brief idea about various multiple access techniques used.

Frequency Division Multiple Access (FDMA) [3] is the earliest multiple access technology and has contributed a major share in determining capacity of first-generation cellular systems. The second generation of cellular generation uses time division multiple access (TDMA) [4] as spectrum sharing method. In the case of spread spectrum multiple access (SSMA), which was introduced in 3G, the bandwidth of transmission is several orders higher than minimum required bandwidth. 4G mobile communication mainly used Orthogonal Frequency Division Multiplexing (OFDM) as the multiple access technique [6].

Present generation wireless networks (till 4G) provide resource to users based on orthogonal multiple access principal. When number of users are increasing drastically, orthogonality approach may not meet the emerging requirements including high connectivity, increased spectral efficiency and massive device connectivity [7]. Non-orthogonal multiple access (NOMA) principle emerges as a possible solution of multiple access technique for 5G with increased spectral efficiency and decreased mutual interference between users. Along with NOMA different multiple access schemes are also implemented in 5G. But this paper focuses NOMA as multiple access of 5G systems. This work contributes on BER analysis of NOMA in AWGN channels for different power levels allocated to users. The relationship between power allotted to each user and BER is studied.

The structure of the paper is organized as follows: Section II discuss existing works of NOMA. Section III discuss advantages of NOMA. In Section IV, motivations and contributions of this work is discussed. In Section V an overview of NOMA is analysed. Section VI deals with results and discussions. Section VII provides the conclusion along with future work and summarises this paper.

II. EXISTING RESEARCH ON NOMA

NOMA is viewed as the most promising multiple access technology for 5G wireless networks [8],[9] because of its increased spectral efficiency. Principle of NOMA relies on multiplexing the information signals for each user into the power domain. Users are allotted power based on their channel gain. Major share of power is provided to users with less channel gain and vice versa. In [11], authors proved that NOMA can achieve higher spectral efficiency because of its capability to exploit channel diversity more efficiently using SIC. Practical considerations of NOMA is surveyed in [12]. Beam formation and application in NOMA is studied in [13]. Power domain NOMA is the most experimented multiple access technology of 5G. But various other forms like Code Domain Multiple Access [14],[15] also has a share in the multiple access technology of 5G systems. [16] shows that NOMA is better than OMA in terms of sum rate because in the case of OMA, there is split in the resource because of orthogonal resource allocation. But for NOMA, complete spectrum allocation is allotted to the users. In [17], ergodic capacity maximization is investigated for Rayleigh fading channel. In [18], MIMO -NOMA framework is evaluated for uplink and downlink transmission. Comparison between the capacity regions of OMA and NOMA is done in [19]. It shows NOMA outperforms OMA in most regions. In [20] the performance analysis of NOMA under various channel is estimated. Bit error rate of NOMA is analysed in [21].

NOMA schemes can be generally classified as two types: power-domain NOMA (PD-NOMA) and code-domain NOMA(CD-NOMA). In PD-NOMA, [8], each user is assigned a unique power level and multiple users share the same time frequency -code resources, using its own allotted power. Channel gain decides the portion of power dedicated for each user. At the transmitter side, signals are superimposed by a process known as superposition coding. At the receiver side, the user signals are decoded by the technique of successive interference cancellation. Code domain NOMA [14],[15] uses a codebook, spreading sequences, and scrambling sequences etc. to allocate resources non orthogonally to users. In this paper NOMA generally represents PD-NOMA on the downlink scenario. NOMA and OMA can be implemented simultaneously in 5G systems. NOMA is suitable only at places where users have distinct channel gains.

III. HIGHLIGHTS OF NOMA

Comparing with OMA, the key highlights of NOMA include the following: -

- Enhanced spectral efficiency and throughput: In NOMA, different users are served simultaneously using the same resources. So compared to the previous generations, spectral efficiency will be very much enhanced. Similarly, throughput performance of NOMA is higher compared to OMA.

- Increased user fairness: NOMA supports comparatively higher level of user fairness using suitable power allocation criteria.

- Decreased transmission latency and signalling cost: In Orthogonal multiple access techniques, the process takes place initialized with a scheduling request which results in large latency. Since this scheduling request can be avoided in NOMA, it can offer low transmission latency and less signalling cost.

- Massive connectivity: Since more users can be connected using NOMA, massive connectivity can be assured here.

- Relaxed channel feedback: In NOMA, CSI feedback is used only for power allocation. So, the accurate CSI value is not needed. If the channel characteristics doesn't change rapidly, it won't affect the system performance.

IV. MOTIVATIONS AND CONTRIBUTIONS

NOMA is an effective technology which increases the spectrum efficiency. Normally the users with poor channel conditions are highly affected in the case of previous generations of wireless technology. In NOMA high consideration is given to users with poor channel condition. But in the receiver side, complexity is increased. Our motivation is to find the suitable power allocation factors for users to give the best BER value. The contributions of our works are as follows:

BER analysis of NOMA is done for AWGN channels with different values of power allocation to users. The graph is shown for various range of SNR. It shows that the response of users changes when power allocation factors change. BER of user with good channel condition is slightly affected because of successive interference cancellation. When equal power allocation is done, NOMA can't be viewed as the best option of multiple access for 5G communications.

V. NON-ORTHOGONAL MULTIPLE ACCESS TECHNIQUE – AN OVERVIEW

A simple NOMA system (Fig 1) consists of a single BS and two users. At the transmitter side, the principal of superposition coding allows for simultaneous processing of input from several users. Each user is equipped with a single antenna. Each user is awarded with a power coefficient depending upon fairness of system and QoS expectation of consumers. The signals are mixed together through a process called superposition coding and is fed to the channel for transmission.

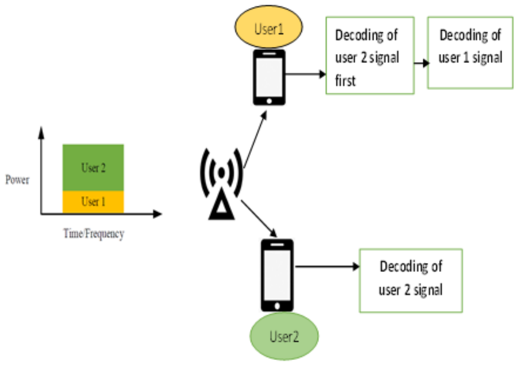

User1
Decoding of user 2 signal first
Decoding of user 1 signal
Power
User 2
User 1
Time/Frequency
Decoding of user 2 signal
User2

Fig. 1. A 2 user NOMA where superposition coding is implemented at transmitter side and SIC is implemented at receiver side.

NOMA uses principle of SIC at the receiver side. SIC is an iterative process where data is decoded in the order of decreasing strength. The data of user with greatest power is initially decoded and it has the least interference contaminated signal. Then strongest user re-encodes and re-modulates its signal and is subtracted from composite signal. Then next strongest signal starts decoding its signal. And finally, the weakest user decodes its information without much interference from other users. SIC offers improved interference management and hence network capacity can be improved. Concept of superposition coding is as follows:

Suppose x_1 and x_2 are the signals to be transmitted from BS to users 1 and 2, respectively. User 1 is the far user which is expected to have worse channel conditions and user 2 is the near user possessing good channel conditions. More power is allocated to user 1. The resultant superposition coded signals are represented as

$$s = \alpha_1 x_1 + \alpha_2 x_2. \qquad (1)$$

where α_1 and α_2 are the power allocation factor for user 1 and 2. Sum of α_1 and α_2 should be unity. [$\alpha_1+\alpha_2= 1$]

The transmitted signal is given by

$$y_i = h_i s + N \qquad (2)$$

Where h_i is the channel gain between BS and user. N represents noise in the channel.

Consider the scenario where 2 users, are transmitting data x_1 and x_2 respectively. Assume each user has five bits of data to send. Let x_1 = 00101 and x_2 = 11101. The user signals are first digitally modulated (BPSK) before transmission. In superposition coding principle, the signals are assigned with different power levels and then superimposed with each other. The general rule is that $\alpha_1+\alpha_2$ should be equal to one. The power factors are scaled to make them represent amplitude. After scaling, the signals are added and superposition coding signal is obtained (Fig 2).

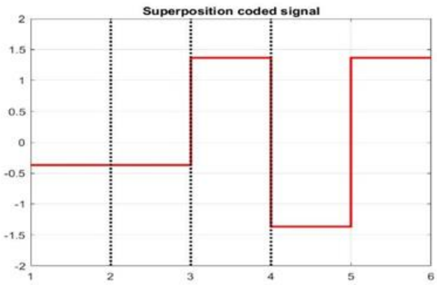


Fig .2. Resultant superposition coded signals.

VI. RESULTS AND DISCUSSIONS

We have considered AWGN channels for various SNR where power allocation is different for 2 users and in the last case equal power allocation is provided for users. (Fig .4d). BER Analysis of NOMA is done for all the cases. Response of user changes when power allocation factor changes. Let's first consider the cases where power allocation is different, i.e. we are giving more power to the user with worse channel condition and less power to the user with good channel condition. BER of user with good channel condition is slightly affected because of successive interference cancellation. Here first we allow fixed power allocation strategy for different values of SNR. By fixed allocation, we always set α_1=0.75 (for far user) and α_2=0.25 (for near user). The result is shown for two ranges of SNR (Fig.3(a),3(b)). When fixed allocation strategy is followed, no computation is required. And also, no knowledge of channel state information is required. Next, we allotted different power factors for each set of users and the BER analysis is done (Fig 4). When equal power allocation is provided, BER of users is decreased drastically. So, NOMA is suitable only for user pairs with distinct channel gains. The results shows that power allocation plays a major role in the BER analysis of NOMA system.

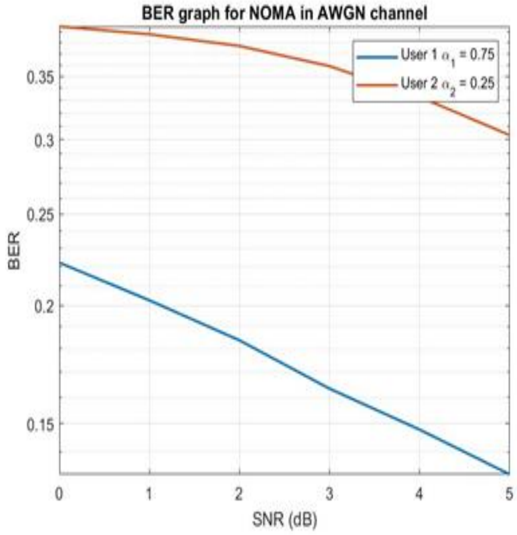


Fig 3. (a). A two user NOMA scheme in the range of 0-5 dB.

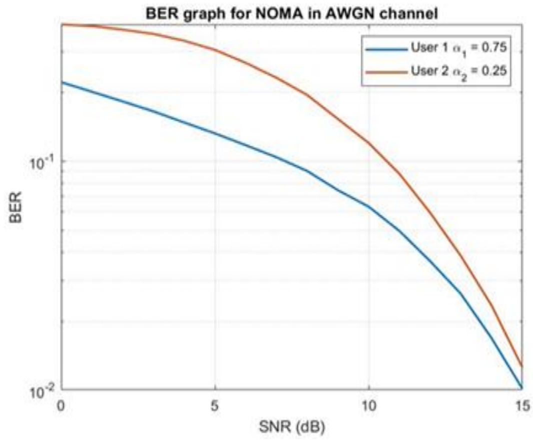


Fig.3. (b). A two user NOMA scheme in the range of 0-15 dB.

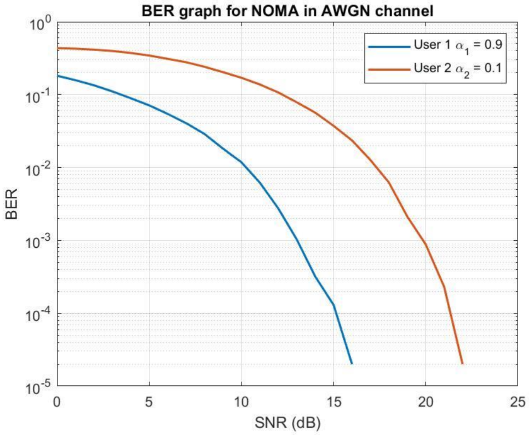


Fig 4.(a). Users 1 and 2 are allotted with powers 0.9 and 0.1 respectively.

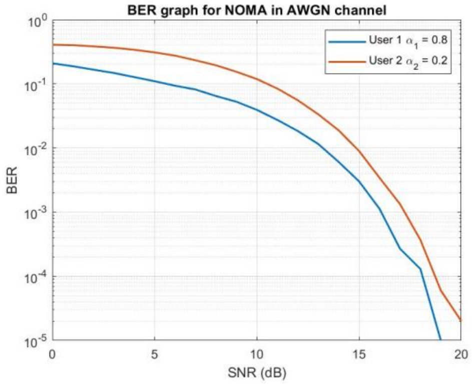


Fig.4.(b) .Users 1 and 2 are allotted with powers 0.8 and 0.2 respectively.

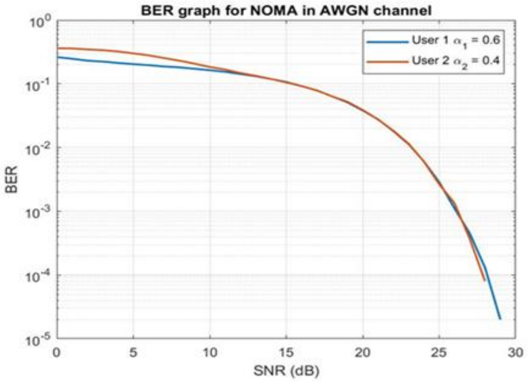


Fig.4(c) Users 1 and 2 are allotted with powers 0.6 and 0.4 respectively

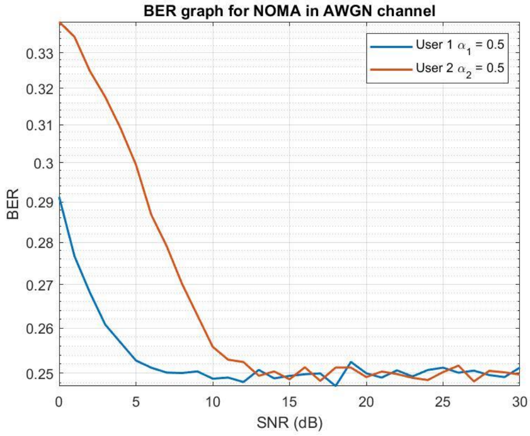


Fig.4. (d). Users 1 and 2 are allotted with equal powers 0.5 and 0.5 respectively.

VII. CONCLUSION AND FUTURE WORK

This work aimed for investigating two user NOMA model and its influence on increasing system capacity. The BER rate of NOMA in AWGN channel for various SNR have been plotted. The results prove that selection of power plays a vital role in performance of NOMA. When the two users are allotted with same power level, the performance of system is drastically compromised. NOMA is suitable only for user pairs with distinct channel gains. When the channel condition changes, dynamic selection of power helps to improve the system performance. As a future work, the performance of NOMA system aided with clustering and multiple antenna techniques can be considered.

REFERENCES

[1] C Shannon. A mathematical theory of communication[J]. Bell Syst Tech J.27, 1948, 379-432.

[2] C Shannon. Communication in the presence of noise[J]. Proc IRE.31, 1949, 10-21.

[3] Liu, G.L. FDMA system performance with synchronization errors[C].Military Communications Conference, 1996. MILCOM '96, Conference Proceedings, IEEE. 1996. McLean, VA.

[4] Raith, K. and J. Uddenfeldt. Capacity of digital cellular TDMA systems[J]. Vehicular Technology, IEEE Transactions on, 1991. 40(2): p. 323-332.

[5] Dong, G.J., G.K. Il and K. Dongwoo. Capacity analysis of spectrally overlaid multiband CDMA mobile networks[J]. Vehicular Technology, IEEE Transactions on, 1998. 47(3): p. 798-807.

[6] Xinyu, Z. and L. Baochun. Network-Coding-Aware Dynamic Subcarrier Assignment in OFDMA-Based Wireless Networks[J]. Vehicular Technology, IEEE Transactions on, 2011. 60(9): p. 4609-4619.

[7] Y. Yifei and Z. Longming. Application scenarios and enabling technologies of 5G[J]. Communications, China, 2014, 11(11): 69.

[8] Y. Saito, Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li, and K. Higuchi, “Non-Orthogonal Multiple Access (NOMA) for Cellular Future Radio Access,” in 2013 IEEE 77th Vehicular Technology Conference (VTC Spring), June 2013, pp. 1–5.

[9] Z. Ding, Z. Yang, P. Fan, and H. V. Poor, “On the Performance of Non-Orthogonal Multiple Access in 5G Systems with Randomly Deployed Users,” IEEE Signal Processing Letters, vol. 21, no. 12, pp. 1501–1505, Dec 2014.

[10] L. Dai, B. Wang, Y. Yuan, S. Han, C. l. I, and Z. Wang, “Non-Orthogonal Multiple Access for 5G: Solutions, Challenges, Opportunities, and Future Research Trends,” IEEE Communications Magazine, vol. 53, no. 9, pp. 74–81, September 2015.

[11] Z. Ding, Y. Liu, J. Choi, Q. Sun, M. Elkashlan, C. L. I, and H. V. Poor, “Application of Non-Orthogonal Multiple Access in LTE and 5G Networks,” IEEE Communications Magazine, vol. 55, no. 2, pp. 185– 191, February 2017.

[12] A. Benjebbour, Y. Saito, Y. Kishiyama, A. Li, A. Harada, and T. Nakamura, “Concept and practical considerations of non-orthogonal multiple access (NOMA) for future radio access,” in Proc. IEEE Intelligent Signal Processing and Communications Systems (IEEE ISPACS’13), Nov. 2013, pp. 770–774.

[13] K. Higuchi and Y. Kishiyama, “Non-orthogonal access with random beamforming and intra-beam SIC for cellular MIMO downlink,” in Proc. IEEE Vehicular Technology Conference (IEEE VTC’13 Fall), Sep. 2013, pp. 1–5.

[14] R. Hoshyar, F. P. Wathan, and R. Tafazolli, “Novel low-density signature for synchronous CDMA systems over AWGN channel,” IEEE Trans. Signal Process., vol. 56, no. 4, pp. 1616–1626, Apr. 2008.

[15] D. Guo and C.-C. Wang, “Multiuser detection of sparsely spread CDMA,” IEEE J. Sel. Areas Commun., vol. 26, no. 3, pp. 421–431, Apr. 2008

[16] D. Tse and P. Viswanath, Fundamentals ofWireless Communication. Cam-bridge, U.K.:Cambridge Univ. Press, 2005, ch. 6.

[17] Q. Sun, S. Han, C.-L. I, and Z. Pan, ``On the ergodic capacity of MIMO NOMA systems,'' IEEE Wireless Commun. Lett., vol. 4, no. 4, pp. 405_408, Aug. 2015.

[18] Z. Ding, R. Schober, and H. V. Poor, ``A general MIMO framework for NOMA downlink and uplink transmission based on signal alignment,'' IEEE Trans. Wireless Commun., to be published in 2016, to be published in 2016, doi: 10.1109/TWC.2016.2542066

[19] P. Xu, Z. Ding, X. Dai, and H. V. Poor, ``A new evaluation criterion for non-orthogonal multiple access in 5G software dened networks,'' IEEE Access, vol. 3, pp. 16331639, 2015.

[20] Sadia, Haleema, Muhammad Zeeshan, and Shahzad Amin Sheikh. "Performance analysis of downlink power domain NOMA under fading channels." 2018 ELEKTRO. IEEE, 2018.

[21] Aldababsa, Mahmoud, et al. "Bit error rate for NOMA network." IEEE Communications Letters 24.6 (2020): 1188-1191.