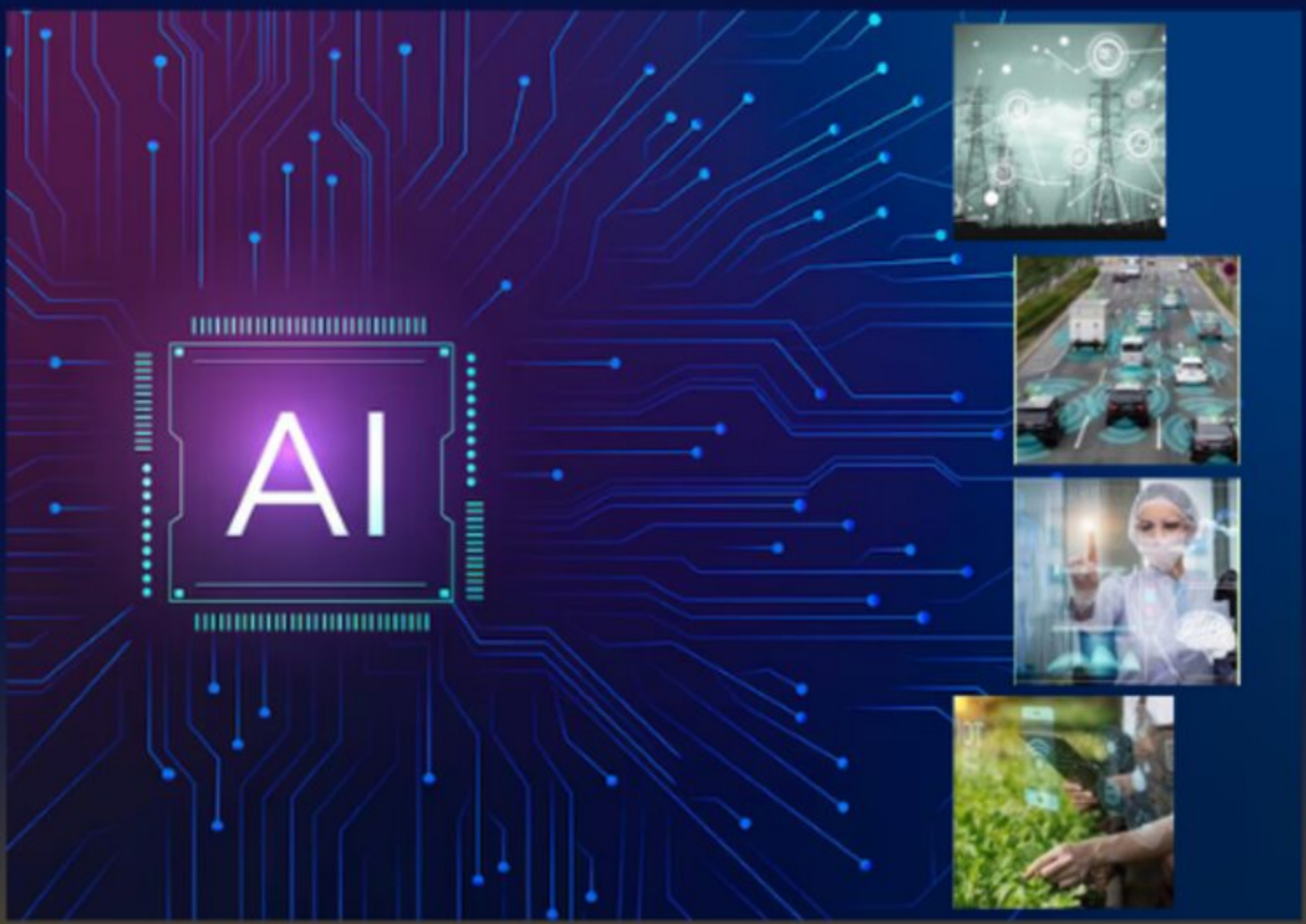
People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
M'hamed BOUGARA University of Boumerdes
Faculty of Sciences

DEPARTMENT OF COMPUTER SCIENCE
&
LIMOSE

Proceedings of the First Workshop on

APPLICATIONS OF ARTIFICIAL INTELLIGENCE (AAI'24)



Boumerdes, 16–17th April 2024

About AAI-24

In today's rapidly evolving world, we face pressing challenges in critical areas such as healthcare, agriculture, energy, urban development, finance, e-commerce, and information security. To address these complex issues, the Workshop on Applications of Artificial Intelligence was organized to create a platform for innovation. This event brought together researchers and PhD students to present cutting-edge AI solutions that specifically address societal challenges. By fostering interdisciplinary dialogue, our aim was to promote AI applications that not only offer immediate solutions but also hold the potential to enhance the overall quality of life.

The workshop took place on April 16–17, 2024, at the University of M'hamed Bougara Boumerdes in Algeria. Participants were invited to contribute to societal advancement by submitting proposals across a range of topics, including but not limited to AI applications in:

Healthcare and well-being

Agriculture and food security

Power and energy transition

Urban planning, transport, and logistics

Finance, e-commerce, and e-payment

Information security and networking

Education and research

We hope that the insights and collaborations emerging from this workshop will inspire new approaches and meaningful contributions in AI for the betterment of society.

Organizing committee

Pr. Ali Berrichi, Head of the department of computer science, UMBB, Algeria

Pr. Menouar Boulif, Director of LIMOSE, UMBB, Algeria

Pr. Mohamed Amine Riahla, Director of ENSTA, Algeria

Pr. Djamel Gaceb, head of a research team in LIMOSE

Dr. Abdellah Rezoug, UMBB, Algeria

Dr. Youcef Yahiatene, UMBB, Algeria

Dr. Djamel Belkasmi, UMBB, Algeria

Dr. Hocine Mokrani, UMBB, Algeria

Dr. Razika Lounas, UMBB, Algeria

Dr. Rachid Djerbi, UMBB, Algeria

Dr. Rabah Imache, UMBB, Algeria

Dr. Med Tahar Bennai, UMBB, Algeria

Dr. Fayçal Touazi, UMBB, Algeria

Dr. Amel Boustil, UMBB, Algeria

Technical program committee

Pr. Djamel Gaceb, UMBB, Algeria

Dr. Djamel Belkasmi, UMBB, Algeria

Dr. Rachid Djerbi, UMBB, Algeria

Dr. Hocine Mokrani, UMBB, Algeria

Dr. Drifa Hadjidj, UMBB, Algeria

Dr. Razika Lounas, UMBB, Algeria

Dr. Mohamed-Taher Bennai, UMBB, Algeria

Dr. Amel Boustil, UMBB, Algeria

Dr. Saida Ishak Boushaki , UMBB, Algeria

Dr. Kheyreddine Djouzi, UMBB, Algeria

Dr. Samya Hamadouche, UMBB, Algeria

Dr. Abdelhak Mesbah, UMBB, Algeria

Dr. Abdelhak Saouli, UMBB, Algeria

Dr. Ali Belgacem, UMBB, Algeria

Dr. Ibtihel Baddari, UMBB, Algeria

Dr. Nabila Rahmoune, UMBB, Algeria

Dr. Besma Alouane, UMBB, Algeria

Dr. Tayeb Benzenati, UMBB, Algeria

Dr. Abdellah Rezoug, UMBB, Algeria

Dr. Fairouz Chahbour, UMBB, Algeria

Keynote speakers

Pr. Chafika Benzaid

Senior Research Fellow at University of Oulu, Finland.

Plenary title: AI for Beyond 5G Networks: A Cyber-Security Defense or Offense Enabler?

Pr. Menouar Boulif

Professor at University of Boumerdes, Algeria.

Plenary title: Artificial Intelligence to manage constraint handling in evolutionary optimization

Table of content

Overview: The Principle Behind Developing Leading NLP Tools

Ali Belgacem[1]

[1] ***Department of Computer Science, University of Boumerdes, LIMOSE laboratory***
a.belgacem@univ-boumerdes.dz

Abstract

Natural Language Processing (NLP) tools have become indispensable assets in various fields, from chatbots for customer service to sentiment analysis in social media, from machine translation systems to information retrieval engines. Their importance lies in their ability to extract meaningful insights from unstructured textual data, facilitate decision-making processes, automate tasks, and enhance user experiences. NLP is a key technology that drives advances across machine learning, artificial intelligence, and linguistics. This workshop explores the fundamental principle underlying the development of popular NLP tools. By the end of the workshop, attendees will have a comprehensive understanding of the principle behind widely used NLP tools. The workshop provides invaluable insights for researchers seeking to improve the performance of an NLP tool or invent new language models rooted in these principles.

Keywords: Natural Language Processing, Artificial intelligence, Popular NLP tools

1 Introduction

In the rapidly evolving field of natural language processing (NLP), many tools and libraries have emerged to address the complexities of understanding and processing human language. These tools play a pivotal role in many applications such as sentiment analysis, machine translation, named entity recognition, text summarization, and others [1]. Numerous popular NLP tools have gained widespread recognition and adoption within the community, ranging from versatile libraries such as NLTK and SpaCy to cutting-edge platforms like Google Cloud NLP API and IBM Watson. Each tool brings its own unique set of features and capabilities. Understanding these tools is essential for anyone venturing into the world of Natural Language Processing (NLP), as they serve as the foundational building blocks for developing sophisticated language processing applications.

Indeed, the ascent of natural language processing (NLP) marks a profound paradigm shift in artificial intelligence, altering how robots grasp, decipher, and produce human language. This transformative trend has introduced large language models (LLMs) like Generative Pre-trained Transformers (GPT), utilizing deep learning methods to generate text that blurs the line between machine-generated and human-authored content. NLP, rapidly evolving fueled by vast datasets and computational progress, wields a pervasive impact across various applications, spanning from virtual assistants and chatbots to sentiment analysis and machine translation [2].

LLMs serve as potent assets in NLP, embodying artificial intelligence systems capable of processing and generating human-like language. Trained on vast text datasets, these models excel in tasks such as natural language processing, language translation, and text generation [3, 4]. Their adeptness in generating coherent and relevant responses renders them invaluable in various applications, from chatbots to content creation, while their adaptability extends to virtual assistants, customer service, and voice-enabled technologies.

On the other hand, the GPT series stands out among LLMs, distinguished by its utilization of transformer architecture and its diverse models, with GPT-3 featuring over 175 billion parameters. GPT-3 demonstrates exceptional performance in NLP tasks such as language translation and question answering, broadening its applications to various formats and genres, including news articles, poetry, and code [5]. Conversely, chatbots simulate human conversation primarily online, employing NLP algorithms and AI techniques to interpret and respond to user input [6]. These versatile tools address frequently asked questions (FAQs), offer customer support, and even facilitate tasks like ordering food or making reservations. Additionally, chatbots designed as virtual pets or chat-based games provide companionship or entertainment.

One of the leading AI research organizations in this field is OpenAI, spearheading the NLP revolution with a commitment to developing safe and beneficial AI solutions. Renowned for its groundbreaking research across NLP, robotics, and reinforcement learning, OpenAI's innovations have far-reaching societal impacts [7]. NLP tools' integration into various fields, from chatbots to language translation, underscores their relevance to both researchers and industry professionals. Rigorous research has evidenced these tools' state-of-the-art performance, while evaluations of their backend components facilitate performance enhancements and issue resolutions, ensuring optimal functionality [3].

During this workshop, participants will discover the basic principles of popular NLP tools and gain insight into their main practical applications. The rest of the paper is structured as follows: Section 2 discusses Popular NLP (Natural Language Processing) tools. Section 3 unveils the Architecture of Leading NLP Tools. Section 4 explores Potential Application Areas. Finally, Section 5 presents the conclusion.

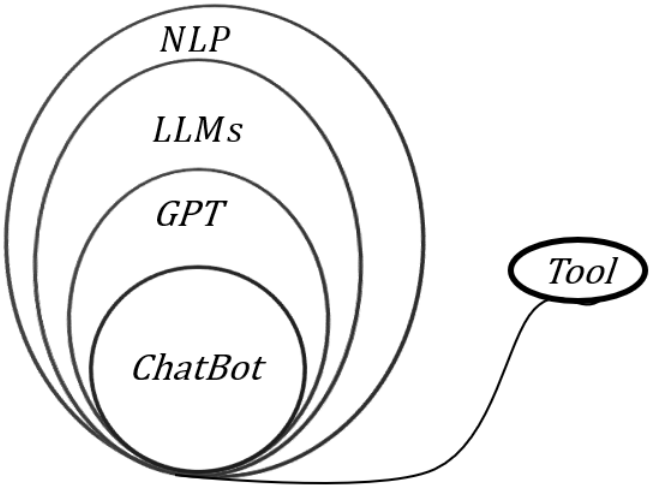


Figure 1: Relevant Technologies for NLP tools

2 Popular NLP (Natural Language Processing) tools

NLP tools, harnessing the capabilities of natural language processing and artificial intelligence, stands alongside several existing software and platforms that operate on similar functional principles. Among these, OpenAI's GPT series, featuring models like GPT-2 and GPT-3, stands out for its prowess in generating coherent and contextually relevant text based on provided prompts, akin to the functionality of NLP tools. Similarly, Google's BERT (Bidirectional Encoder Representations from Transformers) focuses on bidirectional contextual understanding of text, although primarily utilized for tasks such as text classification and named entity recognition, it can also be adapted to generate conversational responses akin to those produced by NLP tools.

Additionally, platforms like Microsoft's DialoGPT, tailored specifically for generating human-like responses in conversational contexts, and Facebook's BlenderBot, integrating techniques from large-scale language models, reinforcement learning, and retrieval-based methods to produce contextually relevant responses, are notable counterparts to NLP tools. Moreover, open-source solutions like Rasa, offering tools for natural language understanding, dialogue management, and response generation, enable developers to craft custom chatbots aligned with specific domains and applications. Similarly, Dialogflow, a Google-owned platform, provides robust capabilities for building conversational interfaces, including chatbots and voice assistants, with features like natural language understanding, integration with messaging platforms, and tools for designing conversational flows and responses. While these platforms and models share common ground with NLP tools in leveraging natural language processing techniques and large-scale pre-trained models for generating human-like responses, their unique features, strengths, and limitations are shaped by their underlying architecture and focus on specific use cases.

3 Unveiling the Architecture of Leading NLP Tools

Creating a machine learning system involves careful attention to three basic components: tasks, algorithms, and models[8]. In addition, datasets play a crucial role in developing machine learning models. This framework logically corresponds to the learning system of NLP tools, and thus it is not surprising that it adheres to this format (Figure 2). In the literature, many researchers studying NLP tools have implicitly referred to aspects of this structure, as described below:

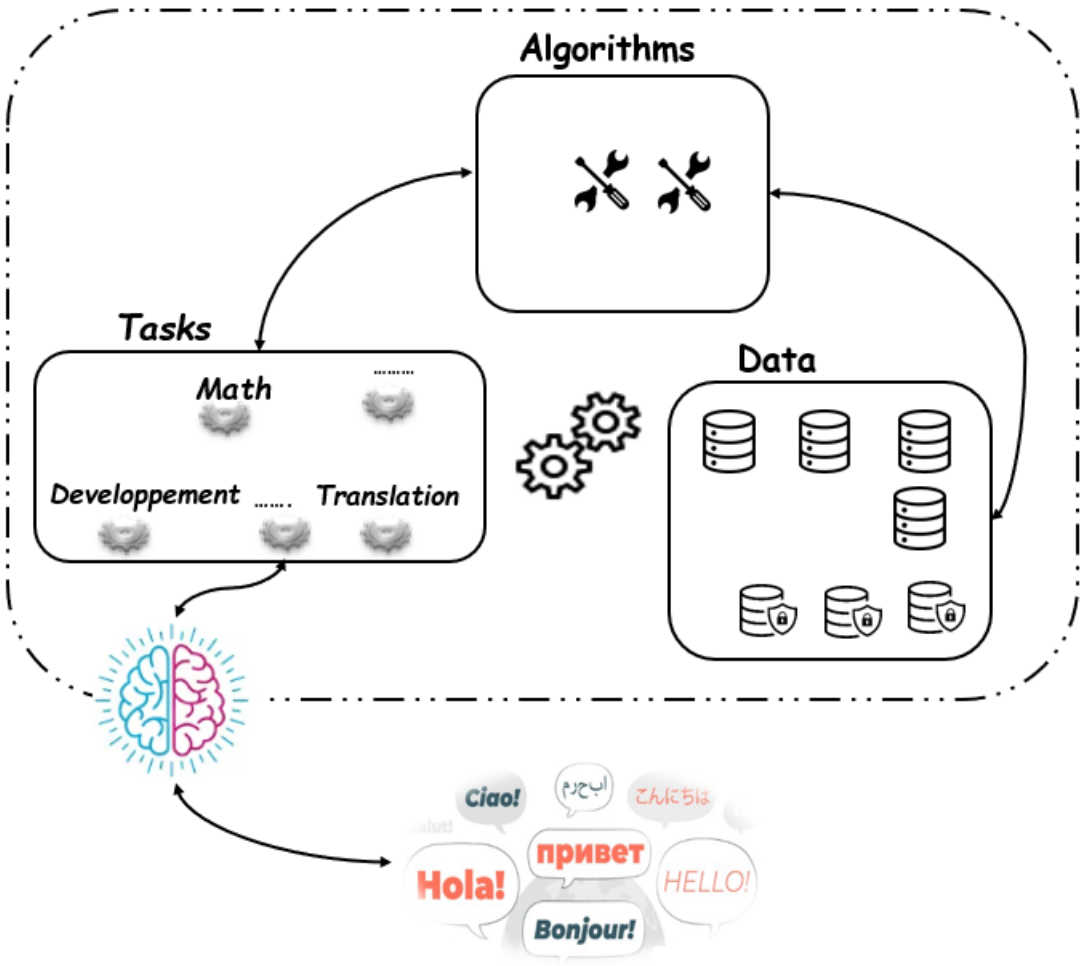


Figure 2: NLP tools machine learning system building

3.1 Tasks

When we discuss a "task" in the context of machine learning, we are referring to a specific problem that a machine learning algorithm aims to solve. In practice, a learning model can be tailored for either a singular task or for multiple tasks concurrently. NLP tools encompass a wide array of tasks, each serving distinct purposes [3, 9, 4, 10]. These tools demonstrate their versatility across various domains through a diverse range of functionalities. They excel in text generation, generating coherent and contextually relevant responses crucial for applications such as chatbots, language translation, and content creation. Additionally, NLP tools facilitate seamless language translation, effectively bridging communication barriers and enabling interaction across linguistic divides. They also play a pivotal role in sentiment analysis by accurately discerning the emotional tone of text, which aids in gauging opinions or attitudes. Moreover, NLP tools adeptly categorize text into distinct topics or classifications, serving purposes such as content moderation, spam detection, and document organization. With the capability to provide answers based on contextual understanding or knowledge bases, these tools enhance functionalities like virtual assistants and customer support through question answering. Furthermore, they streamline information

processing by condensing lengthy text into concise summaries, thus facilitating content curation and information retrieval tasks. Leveraging NLP models to simulate conversations between users and AI agents enables the creation of chatbots, virtual assistants, and interactive applications. Notably, GPT-4, a large-scale multimodal model, showcases recent advancements in the field by uniquely processing both image and text inputs to generate text outputs.

3.2 Algorithms

NLP tools' algorithms are rooted in the GPT architecture, augmented by various techniques to enhance performance and capabilities, tailored to specific applications or use cases for which the model is trained or deployed. These algorithms form the backbone of NLP systems, enabling them to process and generate human-like language with increasing accuracy and efficiency. Leveraging the foundational principles of the GPT architecture, NLP algorithms undergo continual refinement and adaptation to address evolving challenges and requirements across diverse domains. From preprocessing to post-processing, data augmentation to transfer learning, these algorithms play a pivotal role in the seamless integration of NLP tools into various real-world applications, ranging from chatbots and virtual assistants to language translation and sentiment analysis.

Key algorithms employed by NLP tools encompass a range of preprocessing, post-processing, data augmentation, transfer learning, task-adaptive pretraining, and language modeling techniques. Preprocessing algorithms handle tasks like text cleaning, lowercasing, tokenization, stopword removal, and text normalization, ensuring the input data is properly formatted and structured for further analysis. Post-processing steps refine results through tasks such as part-of-speech tagging and named entity recognition, enhancing the quality and relevance of outputs for downstream applications. Data augmentation algorithms generate additional training data by applying transformations, effectively reducing overfitting and enhancing the robustness of the model. Transfer learning fine-tunes pre-trained models for specific tasks, adapting them to new data and domains to improve performance and generalization capabilities. Additionally, task-adaptive pretraining and language modeling algorithms play a crucial role in capturing intricate language patterns and structures, enabling NLP tools to accurately predict word sequences and generate contextually relevant responses.

3.3 Modeles

The OpenAI API offers a flexible solution capable of handling diverse tasks spanning natural language processing, code interpretation, and image manipulation. With models of varying complexity tailored to different use cases, users can leverage the API for tasks such as content generation, semantic search, and classification [11]. This versatility extends to the customization of models, enabling users to fine-tune them according to their specific requirements. Whether it's generating content, conducting semantic searches, or performing classification tasks, the OpenAI API provides a suite of models equipped to handle diverse challenges and applications.

Among the models offered by the OpenAI API, GPT-4 stands out as the latest and most powerful iteration, building upon the GPT-3.5 architecture to enhance natural language understanding and generation capabilities. NLP tools, on the other hand, leverages the GPT-3.5-Turbo model, specifically optimized for conversational formats. Users have the flexibility to choose the most suitable model for their needs, whether they require a robust language model like GPT-4 or one tailored for specific tasks. Commonly utilized models include GPT-3.5, capable of understanding and generating natural language or code, DALL·E for image generation based on natural language prompts, Whisper for audio-to-text conversion, and Moderation for detecting sensitive content within text inputs. Additionally, models like GPT-3 and Codex offer comprehensive language understanding and code generation capabilities, further expanding the range of tasks that can be accomplished using the OpenAI API [10, 7].

3.4 NLP tools data

NLP tools underwent extensive training as an AI language model using vast datasets comprised of diverse textual data from a multitude of sources and domains, ranging from books and web pages to scientific papers and social media content. Although the datasets employed for training various iterations of NLP tools may vary in size, quality, and focus, their overarching aim remains consistent: to furnish the model with a comprehensive and representative sample of language usage across different contexts [9]. Before being fed into the machine learning model, these datasets typically undergo preprocessing to eliminate noise, irrelevant information, or bias, thereby optimizing the model's language modeling

task, which revolves around predicting the subsequent word or sequence of words based on the preceding context. Notable datasets utilized in training different versions of NLP tools encompass collections such as booksCorpus, common crawl, english wikipedia, openWebtext, and webtext [11].

These datasets can be classified into several categories based on their characteristics and applications. Dialogue datasets, for instance, comprise conversations, interactions, or dialogues between two or more speakers, offering valuable insights into various aspects of natural language processing research. Examples include the cornell movie dialogs corpus, persona-chat, dailydialog, ultiWOZ, and convAI2. Large-scale datasets, essential for training Large Language Models (LLMs) like NLP tools, encompass vast amounts of data collected from diverse sources such as sensor data, social media, and transactional records. Notable examples of such datasets include omageNet, common crawl, openstreetMap, million song dataset, yelp dataset, and twitter dataset. Multi-domain datasets span multiple domains or topics, enabling machine learning models trained on them to generalize effectively across diverse domains. Amazon reviews multilingual, yahoo! comprehensive answers, AG news corpus, pubmed, wikiBio, and MS COCO are prominent examples in this category. Additionally, specific datasets tailored to particular domains, like the math word problems extracted from the DRAW-1K dataset analyzed in reference [12], serve specialized research or application needs within specific domains.

4 Potential Application areas

NLP tools has been extensively investigated across various domains, showcasing its potential to make significant contributions in diverse areas:

In the realm of data science processing, [13] delves into how NLP tools is reshaping the landscape of data science by streamlining critical tasks such as data cleaning, preprocessing, model training, and result interpretation. The study highlights compelling evidence of NLP tools' prowess in supporting data scientists, enhancing the efficiency, effectiveness, and accuracy of their work. By leveraging NLP tools, researchers can unlock valuable insights from complex datasets, paving the way for transformative advancements in the field of data science.

Turning attention to robotics, [14] conducts an in-depth exploration of OpenAI's NLP tools' potential applications in robotics. The study sheds light on the effectiveness and limitations of NLP tools in efficiently tackling various tasks, demonstrating its natural language processing capabilities that enable seamless interaction through natural language instructions. These insights underscore the promising role of NLP tools in enhancing the functionality and user experience of robotic systems, opening avenues for innovation and advancement in the field of robotics.

5 Conclusion

This workshop provided a comprehensive exploration of NLP tools, delving into their back-end architecture and core components. Through an analysis of tasks, datasets, and models, we gained valuable insights into the technical complexities of NLP tools. With continuous evolution and improvement, facilitated by regular updates, NLP tools remain at the forefront of technological innovation in the fields of artificial intelligence and machine learning. As we move forward, NLP tools are poised to revolutionize the way we interact with technology, opening up new possibilities and reshaping different areas across industries.

References

[1] Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, et al. The language interpretability tool: Extensible, interactive visualizations and analysis for nlp models. *arXiv preprint arXiv:2008.05122*, 2020.

[2] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021.

[3] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[4] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*, 2020.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[6] Sameera A Abdul-Kader and John C Woods. Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, 6(7), 2015.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[8] Sunila Gollapudi. *Practical machine learning*. Packt Publishing Ltd, 2016.

[9] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

[10] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021.

[11] On line. Gpt-4 is openai's most advanced system, producing safer and more useful responses. https://openai.com/product/gpt-4. Accessed on April 14, 2023.

[12] Teo Susnjak. Chatgpt: The end of online exam integrity? *arXiv preprint arXiv:2212.09292*, 2022.

[13] Hossein Hassani and Emmanuel Sirmal Silva. The role of chatgpt in data science: How ai-assisted conversational interfaces are revolutionizing the field. *Big Data and Cognitive Computing*, 7(2):62, 2023.

[14] Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. Chatgpt for robotics: Design principles and model abilities. 2023.

Design of an intelligent irrigation system

Reda Delli [1], Yassine Ait ali yahia [2], Imane Zian [3], and Neila Benseghir [4]

[1] ***ENSA(école nationale supérieure agronomique) ,*** *reda.delli@edu.ensa.dz*
[2] ***ESI (école supérieure d'informatique),*** *y.aitaliyahia@gmail.com*
[3] ***ENSA(école nationale supérieure agronomique),*** *imane.zian@edu.ensa.dz*
[4] ***ENSA(école nationale supérieure agronomique),*** *neila.benseghir@edu.ensa.dz*

Abstract

Effective irrigation management is crucial due to water scarcity especially for large scale crops, growing agricultural demands, climate change, soil salinity, and suboptimal methods. Despite the agricultural sector's major water consumption (70% of total usage), its inefficiencies necessitate practical solutions and rational water management. The adoption of smart irrigation, powered by AI and ML techniques like Random Forests and Recurrent Neural Networks, marks a significant transformation in agriculture, aiming to reproduce human expertise. This study aims to integrate AI models into irrigation, focusing on the evaluation of random forest and recurrent neural networks. The goal is to enhance the accuracy of irrigation forecasts and decisions while minimizing human intervention.

Keywords: Intelligent irrigation, artificial intelligence (AI), machine learning (ML), deep learning (DL), random forest (RF), recurrent neural networks (RNN), optimal timing, water quantity.

1 Introduction

Irrigation is crucial for crop growth, necessitating an understanding of the soil-water-plant relationship, as well as the factors influencing the efficiency of irrigation [4]. Despite drawbacks of traditional methods, such as excessive water consumption, smart agriculture emerges as a solution to address challenges like growing demand and labor decline [2], there is a need for intelligent systems to efficiently utilize available water throughout the growing season [1]. In response to water shortages due to climate change, our work builds on Djafour Sara's 2022 research, proposing an intelligent watering model using AI technologies. Specifically, we employ a Random Forest model for wheat cultivation and a simple recurrent neural network for forecasting water quantity, this innovative approach employs two distinct AI models: the first model predicts irrigation decisions, while the second model forecasts the required quantity of water. We use historical data related to irrigation decisions as well as parameters related to soil, climate, and plants to train and evaluate the random forest model, making it capable of handling new data.

2 Materials and methods

2.1 Implementation of random forest for wheat cultivation

Our random forest model comprises multiple decision trees, each independently predicting irrigation presence or absence. However, simultaneously working on all crops is impractical. Therefore, we conduct independent training and evaluation for each crop, presenting a detailed approach for wheat cultivation and a concise overview for other crops.

We utilized available Kaggle data with 501 entries, featuring information on nine crops. Each data entry represents a unique observation and includes 5 features: crop type, crop days, soil moisture, temperature, humidity, and a binary 'Irrigation' variable indicating application (class 1) or absence (class 0). Specifically for wheat, there were 26 cases of irrigation and 52 cases without, (this is an excerpt; the data contains many more entries). Figure 1 below presents the datasets used for training the Random Forest (RF) model.

	Type de culture	Jours	Humidité du sol	température	Humidité de l`air	Irrigation
0	Blé	10	400	30	15	0
1	Blé	7	200	30	32	0
2	Blé	9	300	21	28	0
3	Blé	3	500	40	22	0
4	Blé	2	700	23	34	0
...	...	...	...	...	...	...
496	caféier	93	675	25	19	1
497	caféier	95	210	23	17	0
498	caféier	97	398	25	18	0
499	caféier	99	678	24	18	1
500	caféier	101	201	21	14	0

501 rows × 6 columns

Figure 1: The dataset of the RF model

To implement the Random Forest model on Collab, we import and clean data from. The dataset is split into training (67%) and test sets (33%) using Scikit-Learn's train_test_split, it was an experimental choice with this choice, we were able to build a good model, which yielded better results. The "Random Forest Classifier" algorithm is applied with parameter tuning for a high-performing model. Table 1 below shows the selection of the most suitable model parameters for optimizing wheat cultivation practices, the choice of these parameters was made experimentally, these parameters allowed us to achieve good results.

Number	Parameter	Value
1	N_estimators	15
2	Max_depth	9
3	Max_features	None
4	Min_samples_leaf	9
5	Max_leaf_nodes	30
6	**Min_samples_split**	**10**
7	Random state	123
8	Bootstrap	True
9	Criterion	gini
10	ccp_alpha	0

Table 1: Selection of model parameters for wheat cultivation

We trained the model on 67% of the dataset to learn patterns and relationships. Evaluation on the test set includes key metrics like the classification report and confusion matrix to assess performance.

In data science, saving a trained model for future use is crucial. We use the 'joblib' library to save model weights and configuration, naming the file 'RandomForestClassifier' to indicate its foundation in scikit-learn.

2.2 Implementation of a simple univariate recurrent neural network model

Recurrent neural network is a neural network specifically designed for processing sequential data while preserving a memory of past events.

We obtained a dataset with 9,999 entries from the same website, containing information on watering time in seconds and hours. Each entry includes the associated water volume, which serves as the model's prediction variable. Figure 2 below presents the datasets utilized for recurrent neural network model which includes (time in seconds and hours plus the water volume).

	élement	temps(s)	volume d`eau	temps(h)
0	0	1.451632e+09	36.86	14.0
1	1	1.451646e+09	38.13	12.0
2	2	1.451660e+09	21.22	16.0
3	3	1.451639e+09	15.26	10.0
4	4	1.451632e+09	9.47	8.0
...	...	...	...	...
9994	9994	1.624540e+09	203.39	14.0
9995	9995	1.624532e+09	0.88	12.0
9996	9996	1.624547e+09	0.86	16.0
9997	9997	1.624525e+09	0.04	10.0
9998	9998	1.624518e+09	0.00	8.0

9999 rows × 4 columns

Figure 2: The datasets of the recurrent neural network model

In our recurrent neural network model with the dataset, we remove irrelevant columns, normalize for consistent scales, and create 10-sample sequences (X for time, Y for volume). Training uses 90% of the data, reserving 10% for testing (experimental choice).

The recurrent neural network architecture consists of three layers: a hidden layer with 10 neurons, a dropout layer, and a dense layer with a single neuron for regression tasks. The 'tanh' activation function, along with bias, adapts the network to input data. The 'adam' optimization algorithm minimizes error using mean absolute error for accurate real value approximation. Table 2 below represents the recurrent neural network parameters used in this study.

	Parameter	Value
1	Samples	10
2	Steps	1
3	Threshold	0.9
4	Activation function	tanh
5	Bias	true
6	Number of neurons	10
7	**Layer dropout**	**0.2**
8	Number of layers	3
9	the loss function	adam
10	Epochs	100
11	Lot (batch size)	32
12	Shuffle	false

Table 2: The recurrent neural network parameters

The model undergoes 100 epochs, allocating 20% of the training data for validation. Post-training, we use the model to make predictions and compare them with actual values. This evaluation is crucial for assessing the model's performance and accuracy in predicting target values. We have chosen the 'model.save()' method to store both the model weights and its configuration in an HDF5 file since our model is based on the Keras library.

3 Results and discussion

3.1 Random forest

For the five variables, the training set comprises 52 entries, while the test set consists of 26 entries. These results are determined by the random state parameter, a crucial factor for understanding the size of our training and test data when building and evaluating machine learning models.

The model demonstrates high precision (100% for class 0, 89% for class 1) and notable recall rates (94% for class 0, 100% for class 1). With an overall accuracy of 96% for wheat cultivation and an average accuracy rate of 98.33% across all crops, the model's robust and efficient performance is evident. Table 3 below represents the model success rate in 10 crops in %.

Culture	Accuracy (%)
Wheat	96
Corn	100
Peanut	100
Garden flower	94

Rice	**100**
Potatoes	**95**
Legumes	**100**
Sugarcane	**100**
Coffee	**100**
Average	**98.33**

Table 3: Model success rate for each crop

Table 4 summarizes the average metrics for all crops across each class, showcasing the model's strong performance in terms of accuracy, recall, F1 score, and support.

- **Precision:** The proportion of positive observations correctly classified among all the positive observations predicted by the model. Precision (P) =TP/TP+FP
- **Recall**: This measures the proportion of positive observations correctly identified among all the actual positive observations in the dataset. Recall (R) =TP/TP+FN
- **F1-score:** The F1 score is a combined measure of precision and recall, expressed as the harmonic mean of the two. F1-score =2*P*R/(P+R)
- **Accuracy:** it measures the total proportion of correct predictions, whether they are positive or negative. Accuracy =TP+TN/TP+FP+FN+TN

Class / Metrics	0	1
Precision (%)	98,33333	98,625
Recall (%)	99,33333	95,88889
F1-score (%)	98,77778	97,11111

Table 4: Average metrics for all crops

To understand how the model makes decisions based on the given variables, we present below in figure 3 a visualization of a single decision tree from the model.

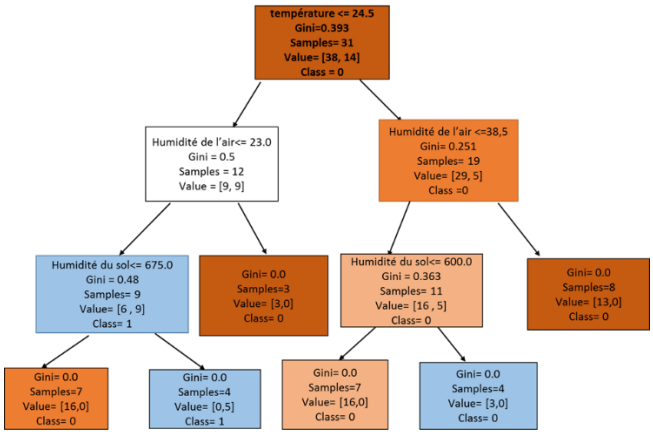


Figure 3: A decision tree of the random forest model

3.2 Recurrent neural network

Minimizing the error towards zero serves as an indicator of good recurrent neural network performance, implying that the model aligns well with reality. The selection of 100 epochs is associated with a significant reduction in the error towards 0.0152 and the mean absolute error towards 0. 080.

The goal is to minimize the loss function to reduce prediction error. The validation loss, used to assess the model's performance, is slightly lower than the training loss by a margin of 0.02, suggesting a slight underfitting. This indicates that the model has not fully adapted to the training data, implying a gap between its performance on training and validation data. Figure 4 below is a graph representing the evolution of the loss function for the recurrent neural network model.

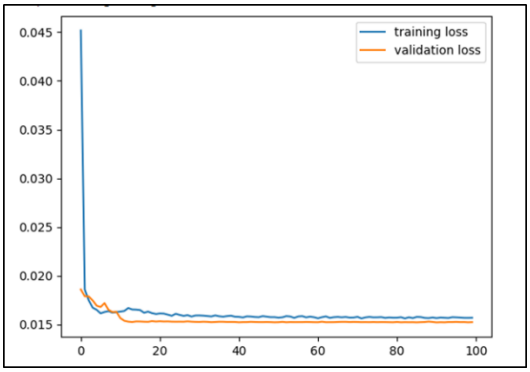


Figure 4: Evolution of the loss function for the recurrent neural network model

We have a graph showing a close alignment between predicted and true values. Notably, predicted values plateau, indicating the model's best effort in predicting maximum values.

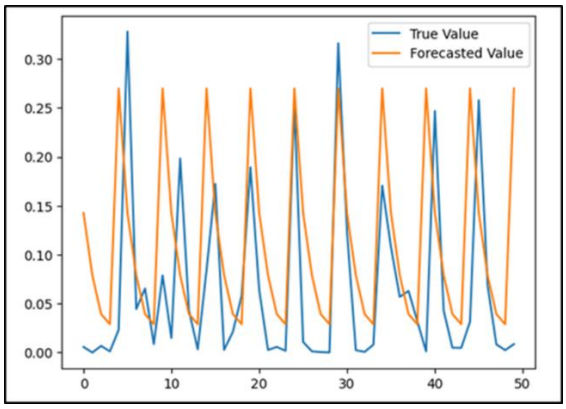


Figure 5: Comparison between true values and predicted values for the RNN model

4 Conclusion

The integration of AI in agriculture becomes increasingly crucial amidst the challenges posed by climate change and water scarcity. By automating irrigation decisions, efficiency is significantly improved, thereby addressing issues of food security and promoting environmental sustainability. Our research underscores the effectiveness of employing both random forest and recurrent neural network models. The random forest model, utilizing its own dataset including various parameters, achieves an impressive overall performance rate of 98.33%, particularly excelling in responding adeptly to watering decisions. On the other hand, the recurrent neural network demonstrates notable performance by minimizing prediction errors to 0.0152 and 0.0804 for mean absolute error, utilizing a distinct dataset used to predict water volumes accurately. These promising findings indicate the successful optimization of irrigation timing alongside ensuring adequate water supply.

References

[1] Rao, R. N., & Sridhar, B. (2018, January). IoT based smart crop-field monitoring and automation irrigation system. In 2018 2nd International Conference on Inventive Systems and Control (ICISC) (pp. 478-483). IEEE.

[2] Santos, L., Santos, F. N., Oliveira, P. M., & Shinde, P. (2020). Deep learning applications in agriculture: A short review. In Robot 2019: Fourth Iberian Robotics Conference: Advances in Robotics, Volume 1 (pp. 139-151). Springer International Publishing.

[3] Gavriloff, J. (2021, Mai 24). Fichier CSV : définition, création et import dans Excel. Récupéré sur logiciel HubSpot: https://blog.hubspot.fr/marketing/fichier-csv.

[4] Scherer, T. F., Seelig, B., & Franzen, D. (1996). Soil, water and plant characteristics important to irrigation.

[5] DJAFOUR, S. (2022). L'utilisation de l'intelligence artificielle dans la gestion des Irrigations (final study project dissertation).

From Communities' Detection in Social Networks to Recommendation Systems

Rachid DJERBI[1], M.Tahar BENNAI[1], Rabah IMACHE[1], ITOUA NGALOMI Bil Théodore[1], Seyba Yacouba SISSOKO[1], and Mourad AMAD[2]

[1] *Department of Computer Science, University of Boumerdes, r.djerbi@univ-boumerdes.dz*
[2] *University Akli Mohand Oulhadj of Bouira, Algeria*

Abstract

Social networks are a space of communication, sharing, and information but also an economic environment full of marketing and advertising services. For this purpose, they are used to recommend products, services, friends, ..., to different subscribers. We are talking about Recommendation Systems (RS). These systems often make a well-targeted diffusion to reach the right population at the right time. The goal of this paper is to find, for an input user, a set of items to recommend to him and, similarly, find a set of users to whom to recommend an input item. For this, firstly, we propose how to find these two sets for a given input using the "Communities' Detection" mechanism, of which we have different approaches and algorithms in the literature; we choose one among them and extend it. Secondly, we designed a new model of RS based on communities detection. The choice in this paper is focused on the LFM model (Large Families Model), given its ease, extensibility, and flexibility.

Keywords: Social Networks, Communities' Detection, Recommendation systems.

1 Introduction

Social networks have become a significant means of communication, information sharing, and online services. As users increasingly rely on these platforms, there is a growing need for recommendation systems (RS) to assist them in discovering relevant content, products, and connections. RS have found applications in various domains, such as e-commerce, digital libraries, and social networking, helping to increase productivity, success, and profitability.

However, RS still suffers from some limitations. Recommendations based on past user behavior can lead to false or undesirable recommendations, as user preferences and interests may change over time or not align with the items they have previously interacted with.

To address these challenges and develop a more sophisticated RS that can better understand and cater to user preferences, we propose a new approach based on our previous works on communities' detection.

This paper aims to present a novel recommendation systems approach based on communities' detection in social networks. The underlying hypothesis is that users who share similar interests and preferences are likely to form communities, and by identifying these communities, more accurate and personalized recommendations can be provided.

Specifically, the paper has two main objectives: (1) to find, for a given input user, a set of items to recommend, and (2) to find, for a given input item, a set of users to whom the item should be recommended. To achieve these objectives and introduce a new model of recommendation system, we used a community detection mechanism based on the LFM (Large Families Model) algorithm [3].

The remainder of this paper is organized as follows: Section 2 provides an overview of recommender systems and their applications. Section 3 discusses the concept of community detection in social networks, focusing on the LFM algorithm. Section 4 presents the proposed recommendation system model and its associated algorithms. Finally, Section 5 concludes the paper and outlines future research directions.

2 Recommender systems

Recommender systems (RS) [11][21][12] are a specific form of information filtering that aims to provide users with personalized suggestions for new items, such as products, services, or content. These systems

learn user preferences from their profiles and past behaviors and then use this knowledge to propose new possibilities that the user is likely to find interesting or relevant.

RS have found applications in a wide range of domains [9], including e-commerce [1], digital libraries [2], the security sector [18], collaborative filtering systems [10][19], demographic-based systems [7][16][17], and eLearning [8]. By reducing the time users spend searching for exciting items and suggesting things they may have overlooked, RS can significantly improve the user experience and increase engagement.

3 Communities' detection

The analysis of complex networks, both in the virtual and real world, often involves the detection of communities [5] – groups of nodes that are more densely connected than the rest of the network. This community detection task has gained increasing interest in recent years, with researchers exploring both overlapping [6][20] and non-overlapping [4] community structures.

This paper focuses on one of the recent and efficient community detection algorithms: the Large Families Model (LFM) algorithm we proposed in [7]. This algorithm and an extension will be used as the foundation for the proposed recommendation system approach.

3.1 Large Families Model (LFM)

The key principle of the LFM algorithm is to identify the maximum number of common neighbors between pairs of nodes, which are then considered as the initial "families" or communities. The algorithm then allows other nodes to join these initial communities, optimizing the modularity [14][15][13] of the final community distribution.

The notion of communities is particularly relevant in this work, as it is hypothesized that "two individuals, having chosen the same friends, have more similarity in terms of taste, interests, preferences, and abilities." This insight forms the basis for our proposed recommendation system model.

The LFM (Large Families Model) algorithm proposed in [3] follows a multi-step process to detect communities within the network. The initial step involves decomposing the graph into Maximum Related Components (MRCs). For each MRC, the algorithm proceeds as follows:
1. Randomly select pairs of nodes considered the "parent" nodes.
2. Calculate the number of common "children" nodes for each parent pair.
3. Identify the parent pairs with the maximum number of common children, and designate these as the initial communities or "large families".
4. For each remaining "out" node, determine the most appropriate community for it to join based on the number of connections to the existing community members.

After this initial community detection, the algorithm iteratively adjusts the value of the "CadjMax" parameter, representing the minimum number of common children required for a parent pair to be considered an initial community. The community distribution corresponding to the highest Newman's modularity (Q) [19][15][13] is selected as the final community structure.

This iterative process allows the LFM algorithm to adaptively identify the optimal community boundaries, ensuring that the final community distribution best captures the underlying network structure. In this paper, we describe the extension of the LFM algorithm to support an "egocentric" community detection approach called "Ego-LFM" for the recommendation system application.

This modification allows the algorithm to identify the community structure centered around a specific user or item, an essential requirement for the proposed recommendation system models.

3.2 Ego-LFM

While the standard community detection approaches, such as the LFM algorithm, provide a comprehensive view of the network structure, there are cases where the focus should be on the community centered around a specific node. This "egocentric" community detection can be particularly useful for recommendation systems, where the goal is to identify the most relevant items or users for a given input. In the evolution of the LFM algorithm that supports this notion of egocentric communities, any input (user/item) needs to be initially categorized into one of the following three categories: parent, children, or out-node. We note the input user with the symbol "IU". The following algorithm details only the first case where the input category is parent. The other two cases (IU as children or out-node) can be considered as perspectives for our future works.
Input:

• U: Set of users
• M: Adjacency matrix
• IU: Target user (or item) for which the egocentric community is to be detected
Output:
• Ego-centered community of IU, referred to as “witnesses(IU)” Algorithm:
1. Identify the “spouse” nodes of the input user IU - these are the parent nodes that share the maximum number of common “children” nodes with IU.
2. Add all common nodes (children) in the community.
3. Join any remaining “out” nodes to the existing community based on their connections to the community members.
4. Return the final community, which represents the “witnesses(IU)”—the users (or items) that are most similar to and relevant to the input user (or item) IU.
By focusing on the egocentric community of a specific user or item, the Ego-LFM algorithm can provide a more targeted and personalized set of recommendations instead of considering the entire network structure.
We call ”witnesses” (of recommendation) of a specific user IU (item II), as illustrated in Figure 1, all users (items) members of the community(ies) of IU (II).

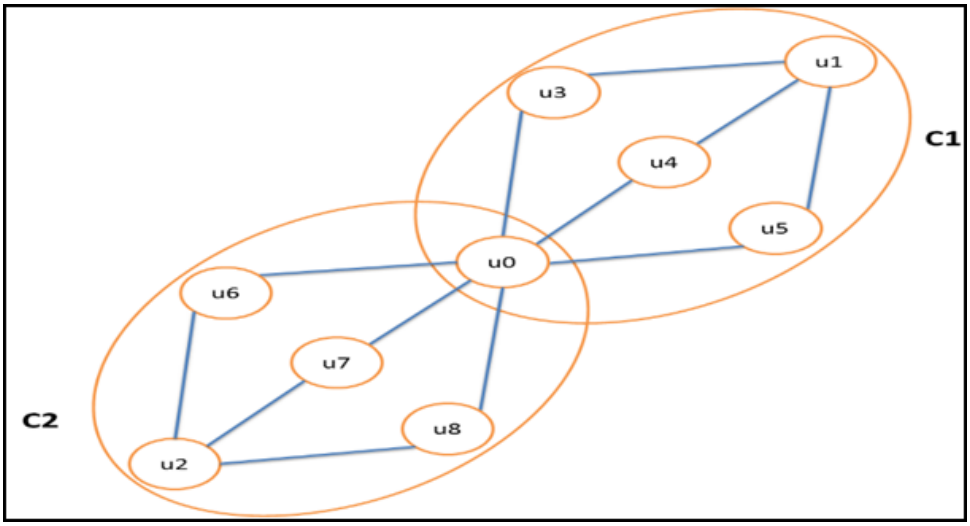


Figure 1: Ego-LFM, two communities for u0

In Figure 1, the set of witnesses(u0)=u1,u2,u3,u4,u5,u6,u7,u8.

4 Our proposition

The core of the proposed recommendation system approach is to leverage the detected communities of a target user (or item) as the basis for generating personalized recommendations. The underlying principle is that users with similar profiles, interests, and preferences will likely form distinct communities within the social network. By identifying the community (or “witnesses”) associated with a given user or item, the recommendation system can then analyze the collective preferences and past actions of that community to suggest relevant items (or users).

The proposed system first applies the Ego-LFM algorithm to detect the egocentric community surrounding the target user (or item). This allows them to identify the users who share similar characteristics and interests with the input user (or those who have interacted with the input item).

The recommendation system then utilizes critical information - the ratings or actions performed by the community members on the various items (or by the users on the input item). These ratings, typically represented as an integer score between 0 and 5 (e.g., number of stars), capture the users’ level of satisfaction or preference for the different items.

The general workflow of the recommendation system approach is as follows:

1. Apply the Ego-LFM algorithm to the target user (or item) to identify the egocentric community, referred to as the “witnesses” of that user (or item).

2. Analyze the past actions (e.g., ratings) performed by the identified community members on the various items (or users).

3. Based on the community's preferences and interests, recommend a set of items (or users) to the target user (or for the target item). To implement this approach, the authors utilize two types of adjacency matrices:

1. User-User Adjacency Matrix: This matrix captures the binary relationships between users. Two users are considered connected if they have provided the "same" rating for at least one common item.

2. Item-Item Adjacency Matrix: This matrix represents the binary relationships between items. Two items are considered connected if they have received the "same" rating from at least one common user.

The choice of which adjacency matrix to use depends on the specific recommendation scenario, as described in the following two algorithms:

4.1 Algorithm RS-LFM-IU (User as Input)

As shown in Figure 2, this algorithm takes a specific user as input and generates a set of recommended items for him. The key steps are:

1. Apply the Ego-LFM algorithm to the input user to identify the egocentric community ("witnesses").
2. Analyze the past ratings provided by the community members on various items.
3. Based on the community's preferences, recommend items to the input user.

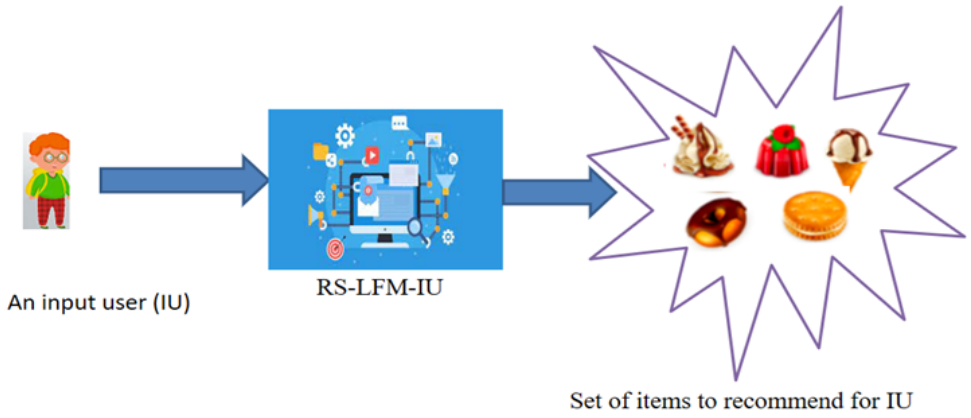


Figure 2: Approach1: RS-LFM-IU

The algorithm that implements RS-LFM-IU is given as follows (Algorithm2):

4.2 Algorithm RS-LFM-II (Item as Input)

The approach described in Figure 4 takes a specific item as input and generates a set of users to whom it should be recommended. We will detail the algorithm that implements RS-LFM-II in our future work. By leveraging the community-based approach and the rich rating data, the authors' recommendation system aims to provide users with more accurate and personalized suggestions, going beyond traditional methods that may rely solely on individual user profiles or broader collaborative filtering techniques

The algorithm that implements RS-LFM-II is given as follows (Algorithm3):

5 Conclusion

In this paper, we proposed a novel community-based approach for recommendation systems. This approach's key advantage is that it focuses the computational efforts on the users belonging to the identified communities rather than considering the entire user profile dataset as in traditional methods.

We have chosen the LFM (Large Families Model) algorithm for communities' detection to enable this community-based recommendation by extending it to support the notion of "egocentric" communities

Algorithm 2 RS-LFM-UI: Recommendation algorithm based on Ego-LFM with user as input

Input:
Set of items
Set of users with relation (Adjacency matrix)
Set of ratings
User IU

Output:
Set of items to recommend to IU

begin
  $k \leftarrow \max(ratings)$ ▷ the maximum note of ratings
  $U \leftarrow witnesses(IU)$ ▷ apply the Ego-LFM algorithm on IU to have its community named witnesses
  $I \leftarrow \{\}$ ▷ set of items to recommend to IU
  for u in U **do**
    $I \leftarrow I \cup uratings(k, u)$ ▷ $uratings(k, u)$: set of items loved (with a note k) by the user u
  end for
  return I ▷ set of items liked by the witnesses of IU
end=0

Figure 3: Algorithm2: RS-LFM-IU implementation

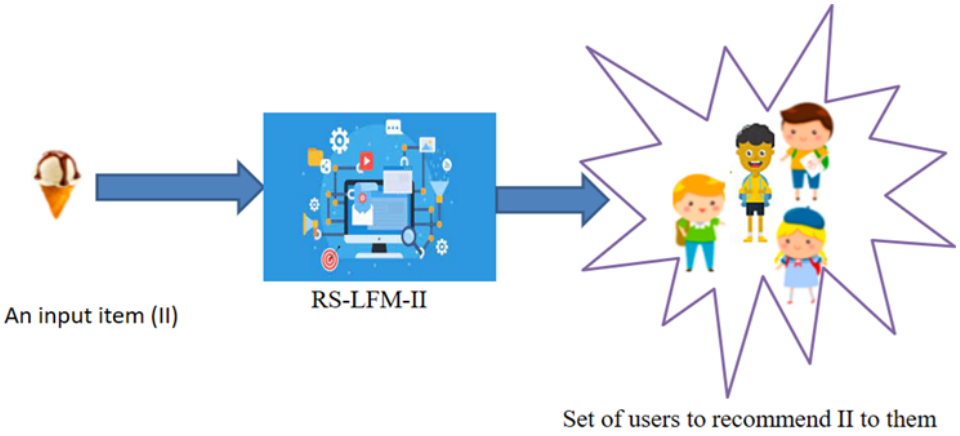


Figure 4: Approach2: RS-LFM-II

centered around a specific user or item. This Ego-LFM algorithm allows for finding the relevant community (or "witnesses") for a given user or item and then leveraging the preferences and interests of these community members to provide personalized recommendations.

Therefore, several future research directions can be taken to enhance the performance and capabilities of their recommendation system, among them:

1. Implementation: The extended version of this work will consider the implementation, application, and experimentation of the proposed approach using datasets like MovieLens and the evaluation metrics.

2. Temporal Dynamics: As users' tastes, interests, and preferences can change over time, the authors propose incorporating the notion of "testimony duration" into the recommendation process. This would involve checking the recency of a community member's interactions with an item before using their preferences to recommend it.

3. Weighted Ratings: Rather than just considering the rating value (e.g., number of stars) provided by a community member, the authors suggest incorporating the frequency of their interactions with similar items. This "weight" of a community member's feedback could help improve the quality of recommendations by focusing on the most engaged and relevant users.

4. Prioritized Recommendations: Instead of presenting all the recommended items, the authors propose sorting the items based on a specific priority metric (e.g., cumulative ratings from the community) and only displaying the top K recommendations to the user.

Algorithm 3 RS-LFM-II: Recommendation algorithm based on Ego-LFM but item as input

Input:
Set of users;
Set of items with relation (Adjacency matrix);
Set of ratings;
Item II;
Output:
Set of users to recommend them item II;

```
1: begin
2:     k ← max (ratings);                                  ▷ the maximum note of ratings.
3:     I ← witnesses (II); ▷ we apply the Ego-LFM algorithm on II to have its community
   named witnesses
4:     U ← {};                                             ▷ set of users to recommend their II
5:     for all i in I do
6:         U ← U ∪ iratings (k, i);  ▷ iratings (k, i) : set of users loved (with a note k) the
   item i.
7:     end for
8:     return U;                                           ▷ Set of users to recommend them item II.
9: end
```

Figure 5: Algorithm3: RS-LFM-II implementation

5. Cold Start Solutions: To address the challenge of providing recommendations for new users or recently added items with limited historical data, the authors plan to explore techniques to leverage the community-based approach to overcome the cold start problem.

References

[1] S Chiang. Combining content-based and collaborative article recommendation in literature digital libraries. master's thesis. *Information Management Department, National Sun Yat-sen University. Taiwan, September. Available via ethesys(DL) at http://etd. lib. nsysu. edu. tw/ETD-db/ETD-search/view_etd*, 2002.

[2] P Connolly and D Reidy. Introduction in the digital library: challenges and solutions for the new millennium. In *Proceedings of an International Conference, Bologna, Italy, Boston Spa, UK: IFLA*, 2000.

[3] Rachid Djerbi, Mourad Amad, and Rabah Imache. A new model for communities' detection in dynamic social networks inspired from human families. *International Journal of Internet Technology and Secured Transactions*, 10(1-2):24–60, 2020.

[4] Rachid Djerbi, Allel Hadjali, Mourad Amad, Rabah Imache, and Mohamed T Bennai. Social context-based non-overlapping communities' detection model in social networks. In *International Conference on Advanced Intelligent Systems for Sustainable Development*, pages 948–958. Springer, 2020.

[5] Rachid Djerbi, Rabah Imache, and Mourad Amad. Communities' detection in social networks: State of the art and perspectives. In *2018 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6. IEEE, 2018.

[6] Yasamin Ghahremani and Babak Amiri. Time series overlapping clustering based on link community detection. *IEEE Access*, 12:41102–41124, 2024.

[7] P Johansson. Design and development of recomendador dialogue systems. licentiate thesis no. 1079. linköping university. *Linköping Studies in Science and Technology*, 2004.

[8] Pradnya V Kulkarni, Sunil Rai, and Rohini Kale. Recommender system in elearning: a survey. In *Proceeding of International Conference on Computational Science and Applications: ICCSA 2019*, pages 119–126. Springer, 2020.

[9] Lydia Kyei-Blankson, Jared Keengwe, and Esther Ntuli. *Designing Equitable and Accessible Online Learning Environments*. IGI Global, 2024.

[10] Jia Li. Using distinct information channels for a hybrid web recommender system. 2004.

[11] Shiwei Li, Huifeng Guo, Xing Tang, Ruiming Tang, Lu Hou, Ruixuan Li, and Rui Zhang. Embedding compression in recommender systems: A survey. *ACM Computing Surveys*, 56(5):1–21, 2024.

[12] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. *Physics reports*, 519(1):1–49, 2012.

[13] Mark EJ Newman. Analysis of weighted networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 70(5):056131, 2004.

[14] Mark EJ Newman. Finding and evaluating community structurein networks. *Physical review E*, 69(26113):1–16, 2004.

[15] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[16] Tomas Olsson. *Bootstrapping and decentralizing recommender systems*. PhD thesis, Uppsala University, 2003.

[17] Elaine Rich. User modeling via stereotypes. *Cognitive science*, 3(4):329–354, 1979.

[18] Felipe Rodriquez. Burning the village to roast the pig. censorship of online media. In *OSCE Workshop on Freedom of the Media and the Internet*, 2002.

[19] Y Shih. Extending traditional collaborative filtering with attributes extraction to recommend new products. *Master's Thesis. Department of Business Administration. National Sun Yat-sen University, 17th may. Taiwan. Available via ethesys (DL) at link http://thesis. lib. ncu. edu. tw/ETD-db/ETD-search/view_etd*, 2004.

[20] Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *Acm computing surveys (csur)*, 45(4):1–35, 2013.

[21] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. Recommender systems in the era of large language models (llms). *arXiv preprint arXiv:2307.02046*, 2023.

Brain Lesion Detection on MRI Images Using Deep Learning: A Review of Vision Transformer-Based Methods

Laribi Nor-elhouda [1], Gaceb Djamel[1], and Rezoug Abdellah[1]

[1] ***LIMOSE laboratory, University M'hamed Bougara, Independence Avenue, 35000 Boumerdes, Algeria,***

Abstract

Brain Magnetic Resonance Imaging (MRI) plays a pivotal role in modern neuroscience and clinical diagnosis of brain lesions. However, the inherent complexity of brain MRI arises from the utilization of multiple techniques, each capturing distinct aspects of brain structure and function. Existing deep learning models often struggle to effectively capture the context of the lesion and the relationships between different parts of the brain in the image, leading to limitations in performance. This paper introduces an overall review explain the inspiration application of self- attention mechanism in transformer from natural language processing to medical imaging especially Brain MRI and how transformer have been used to handle the limitation of existing models in capturing long-range dependencies in the image and the transformer computation costs by integration transformer in different hybrid ways. **Keywords:** Brain lesion detection , Magnetic Resonance Imaging ,deep learning hybrid models , CNN , Transformer , self- attention mechanism.

1 Introduction

Brain lesions are damaged or abnormal tissue in the brain, resulting from injury, disease, infection, tumor, stroke, glioma, Multiple sclerosis, meningioma, pituitary adenoma, trauma, aneurysm, arteriovenous malformation, etc. They affect different brain areas and cause varying symptoms. Accurate diagnosis is crucial for prompt treatment. MRI scans use strong magnetic fields and radio waves to produce detailed images of organs and tissues. By manipulating the timing and properties of these radio waves in different sequences (T1, T2, FLAIR, etc.), we can highlight different tissue properties for Brain Lesions and distinguish between various types of brain lesions. [**?**].

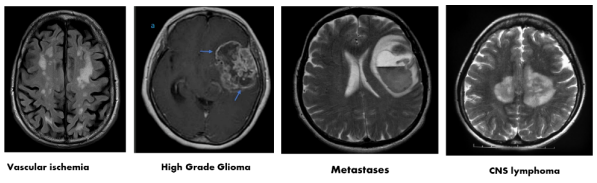


Figure 1: Different types of brain lesion using MRI imaging.

Recently, Deep learning is powerful tool allowing computers to automatically detect and classify abnormalities in MRI scans leading to earlier diagnoses and better patient outcomes. While traditional convolutional neural network (CNN) architectures, like DensenNet, ResNet and Unet, rely on sliding filters to extract local features, they often struggle to capture long-range dependencies and context within the image. This limitation can hinder performance on tasks where understanding the content of the image is crucial. Hybrid deep learning architectures inspired by the success of visual attention mechanisms, transformers, address this by incorporating mechanisms that promote context understanding and features detection, allowing the network to better leverage contextual information during feature extraction. These approaches combine the strengths of Transformer architectures with deep learning models to improve performance and enable efficient cost computation.

Transformers, originally was developed for Natural Language Processing (NLP), and it has become an innovative architecture for analyzing sequential data by capturing long-range dependencies. Transformers have been adapted for various tasks beyond text-based applications which include sentiment analysis

[28], named entity recognition [15], machine translation [16], text summarization [6], in the other hand computer vision application as Vision Transformer (ViT) for image classification [23], DETR (DEtection TRansformer) for Object Detection and Localization [27], Vision Transformer for Semantic Segmentation (ViT-Seg) for Semantic Segmentation [14], Image GPT (iGPT) for Image Generation [12].

This paper presents a literature review which concerns the use of vision transformers (originally inspired from NLP) in the field of computer vision. We describe in particular the most interesting work using these models, with hybridizations with other deep learning models such as CNN for automatic detection of brain lesions on MRI images and other general medical imaging.

2 Transformers from Natural Language Processing to Computer Vision

In the field of Natural Language Processing (NLP), Transformers revolutionized text processing by capturing long-range dependencies in sequential data. Unlike traditional Recurrent Neural Networks (RNNs) that process information sequentially, Transformers excel at capturing long-range dependencies between elements in a sequence by understanding the relationship between tokens (words) at the beginning and end of a sentence. Transformers achieve this through a mechanism called "self-attention" as their name suggests (Attention Is All You Need [23]) which allows the model to attend to all parts of the input sequence simultaneously. First, the elements of the sequence present as words in a sentence. Self-attention doesn't process them one by one like traditional models. Instead, it considers all the elements (words) simultaneously and calculates a score for each element (word) based on its relevance to every other element (word) in the sentence (attention scores). After that, the Feed Forward Network (FFN) operates on the output from self-attention to delve deeper into each element itself, not just the connections between them. The FFN refines this understanding further by allowing the attention module output captures the relationships.

In computer vision, and especially in medical imaging, instead of words and their meaning, self-attention focuses on different parts (elements) of the image (which present the sequence in NLP). Self-attention calculates a score for each part based on its similarity (relationship) to all other parts in the sequence. The goal remains the same which is to understand how each part of the image relates to the whole. In the field of medical imaging based on the use of self-attention mechanism, we identify two categories of transformer models: 2D Transformers and 3D Transformers. The 2D transformer use patches (obtained after cutting of the images into small blocks). These patches become the "elements" that self-attention analyzes. The 2D transformer models such as SwinUnet [11] DS-TransUnet [13] are used for segmentation tasks and achieve, respectively, an average dice score of 79.13% on Synapse multi-organ dataset and of 89.71% on ACDC dataset, where ScribFormer [31] Cascaded MERIT [7] was inspired by the success of both TransUnet and SwinUnet for segmentation tasks of Medical MRI imaging and achieve higher Dice Score (DS = 0.88% and 92.32% respectively) on ACDC dataset. For classification task, LCDEiT [26] is the most relevant model achieve an F-score of 93.43% on Brats 2021 Dataset and 97.20% on Figshare Dataset. All these models use patch-wise self-attention which are computationally efficient, making it easier to apply Transformers to large-scale image Datasets without excessive computational costs. In contrast, 3D Transformer models as Medical transformer [22], SF^{2}Former [10] Longformer [9] MAT [30] SwinMM [1] ViT-Bi-LSTM [8] employ a self-attention mechanism to capture global and local contextual information of 3D volumes where each voxel in the 3D volume attends to other voxels within the same volume by considering the relationships between neighboring voxels as VT-Unet [20], Longformer [9] or dividing the 3D volume smaller 2D slices, and the self-attention mechanism is applied at the 2D slice level as Medical transformer [22] and SF^{2}Former [45]. Each slice attends to other slices within the volume, capturing both local and global contextual information. They achieve mean Area Under the Curve mAUC of 0.8347 ± 0.0072 for classification task on ABIDE Dataset [19] and Dice Score (DS = 0.8733 ± 0.0086) for brain tumor segmentation task on IXI Dataset [24]. However, the projection of 3D objects onto a 2D plane may result in information loss and challenges in representing objects with complex geometries or occlusions. To improve more the computational costs in self- attention, the Swin Transformer addresses this issue by operating on smaller patches instead of individual tokens, reducing the overall computational complexity. This patch-based processing allows for parallelization and reduces the number of operations required for self-attention. By dividing the image into smaller patches, the computational complexity becomes linear with respect to the number of patches rather than the overall image size.

3 Transformers applied to Brain lesion detection on MRI images

The recent research, applied to brain lesion detection on MRI images, shows that several hybrid approaches based on the combination of CNN and Transformer models have been developed, such as the models TransUNet (segmentation of different type of medical images) [17], TRansBTs [17] and BiTr Unet [2] (for brain tumor segmentation) that has reached a Dice Score (DS = 91.79%). In these models, a CNN is used in feature-extractor mode to extract local semantic information and generate a feature map, which is then divided into patches. The sequences are further processed by a Transformer encoder for further refinement.

Another hybridization is based on Swin transformer models, such as Unter [21], SWIN Unter [29] and NnFormer [3], the image data is initially processed by the swin-Transformer, which identifies and highlights the most significant parts of the input. These tokens, representing the image features, are then fed into the CNN decoder to decode and reconstruct the final output image. They achieve by this architecture a DS rate of 71.1% using the model Unter [21] and of 64.35% using the model SWIN Unter [29], on MICCAI 2015 dataset.

In other hand, the CNN and Transformer models are employed simultaneously as both encoder and decoder components. This approach can be observed in models like ST-Unet [5] and achieve a Dice Score (DS = 78.86 %)on Synapse multi-organ dataset. For recognition tasks multiple CNN blocks and Transformer layers are stacked together in MobileViT (achieved an accuracy of 96.7% in classifying Alzheimer's disease) [4] and LightMHS (lightweight model proposed for the segmentation of 3D hippocampus) [25] . These models stack multiple CNN blocks and Transformer layers sequentially.

Additionally, models can be designed to handle multimodal data, MMGL (Multimodal graph learning) and HybridCTrm are two such examples [32] [18] . These models leverage a combination of CNNs and Transformers to extract meaningful features from each data type, and then combine these features for improved performance on downstream tasks. Notably, HybridCTrm achieves an overall mean score of 83.47% for multimodal brain image segmentation. This model is tested on two benchmark datasets and compared with a CNN model HyperDenseNet.

4 Discussion

This paper presents a comprehensive literature review of recent approaches utilizing transformers for medical MRI imaging tasks. The review explores the inspiration behind the attention mechanism and its transition from natural language processing (NLP) to medical imaging applications (particularly, brain lesion detection) using attention mechanism on different patch-wise and voxel-wise manners.

In the field of Brain MRI analysis, hybrid models combining CNN networks and transformers have become the dominant approach. The advancements in brain lesion detection have led to the development of powerful based transformer models such BiTr Unet and the Swin Transformer- based models Unter [21] SWIN Unter [29] NnFormer [3] ST-Unet [3] and SwinMM [1] by integrating a CNN as a feature extractor with a transformer encoder, achieves state-of-the-art performance on the MICCAI datasets.

Alternatively, the processing can be reversed in a stage-wise refinement approach. Here, a ViT takes the lead, performing high-level analysis to identify the overall structure and relationships within the image. The ViT's output is then passed on to another model which can be CNN, Adaptative Graph etwork (GAN), LSTM..., to refine the results by incorporating its strength in local feature extraction mode. This combined approach ensures both a comprehensive understanding of the image's global context and a detailed analysis of its finer details.

However, Self- attention-based mechanisms applied in Traditional Transformers have a quadratic computational complexity, resulting in increased computational requirements as the input dimension grows. The Swin Transformer based models Unter [21] SWIN Unter [29] NnFormer [3], ST-Unet [5] and SwinMM [1] introduce a more efficient computational complexity compared to traditional Transformers, making it a favorable choice for practical applications for brain lesion diagnosis and stands out among hybrid transformer architectures by effectively capturing local and global features through shifted window self-attention mechanisms.

5 Conclusion

In conclusion, hybrid deep learning techniques have emerged as a powerful tool for brain lesion detection and localization, demonstrating promising results in improving diagnosis, treatment planning, and monitoring of different types of brain lesions. By integrating strengths from CNNs and Transformers, these models achieve more accurate and robust detection while optimizing computational efficiency. However, challenges remain. Future research should focus on acquiring larger and more diverse datasets, enhancing interoperability of hybrid models, and further reducing computational costs. Addressing these challenges will unlock the full potential of hybrid deep learning, ultimately leading to improved patient care and outcomes.

References

[1] https://api.semanticscholar.org/CorpusID:265609342. Accessed: 2024-11-9.

[2] https://api.semanticscholar.org/CorpusID:256503807. Accessed: 2024-11-9.

[3] https://abide.readthedocs.io/en/latest/. Accessed: 2024-11-9.

[4] https://api.semanticscholar.org/CorpusID:231847326. Accessed: 2024-11-9.

[5] IXI dataset. http://brain-development.org/ixi-dataset/. Accessed: 2024-11-9.

[6] Taymaz Akan, Sait Alp, and Mohammad A N Bhuiyanb. Vision transformers and bi-LSTM for alzheimer's disease diagnosis from 3D MRI. January 2024.

[7] Cillian Berragan, Alex Singleton, Alessia Calafiore, and Jeremy Morley. Transformer based named entity recognition for place name extraction from unstructured text. *Geogr. Inf. Syst.*, 37(4):747–766, April 2023.

[8] H Cao, Y Wang, J Chen, D Jiang, X Zhang, Q Tian, and M Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, pages 205–218. Springer, Cham; Nature Switzerland, 2022.

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. May 2020.

[10] A Dosovitskiy, L Beyer, A Kolesnikov, D Weissenborn, X Zhai, T Unterthiner, and N Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. International Conference on Learning Representations.* 2021.

[11] T A Elasy and M J Levinsky. Effect of low-dose continuous estrogen and progesterone therapy with calcium and vitamin D on bone in elderly women. *Ann. Intern. Med.*, 132(3):244; author reply 245, February 2000.

[12] Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger Roth, and Daguang Xu. UNETR: Transformers for 3D medical image segmentation. March 2021.

[13] K Itschev. Ultrastructural features of the terminal blood vessel system under various physiological and pathological influences. *Z. Mikrosk. Anat. Forsch.*, 83(3):305–313, 1971.

[14] Qiran Jia and Hai Shu. BiTr-Unet: A CNN-transformer combined network for MRI brain tumor segmentation. September 2021.

[15] Eunji Jun, Seungwoo Jeong, Da-Woon Heo, and Heung-Il Suk. Medical transformer: Universal encoder for 3-D brain MRI analysis. *IEEE Trans. Neural Netw. Learn. Syst.*, PP:1–11, September 2023.

[16] Eunji Jun, Seungwoo Jeong, Da-Woon Heo, and Heung-Il Suk. Medical transformer: Universal encoder for 3-D brain MRI analysis. *IEEE Trans. Neural Netw. Learn. Syst.*, PP:1–11, September 2023.

[17] Rafsanjany Kushol, Collin C Luk, Avyarthana Dey, Michael Benatar, Hannah Briemberg, Annie Dionne, Nicolas Dupré, Richard Frayne, Angela Genge, Summer Gibson, Simon J Graham, Lawrence Korngut, Peter Seres, Robert C Welsh, Alan H Wilman, Lorne Zinman, Sanjay Kalra, and Yee-Hong Yang. SF2Former: Amyotrophic lateral sclerosis identification from multi-center MRI data using spatial and frequency fusion transformer. *Comput. Med. Imaging Graph.*, 108(102279):102279, September 2023.

[18] Y Li, J Tang, L Li, X Wang, W Ding, X Li, and Wu. MobileViT-based classification of alzheimer's disease. In *2023 IEEE 6th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, pages 443–448. 2023.

[19] Zihan Li, Yuan Zheng, Dandan Shan, Shuzhou Yang, Qingde Li, Beizhan Wang, Yuanting Zhang, Qingqi Hong, and Dinggang Shen. ScribFormer: Transformer makes CNN work better for scribble-based medical image segmentation. *IEEE Trans. Med. Imaging*, 43(6):2254–2265, June 2024.

[20] Ailiang Lin, Bingzhi Chen, Jiayu Xu, Zheng Zhang, Guangming Lu, and David Zhang. DS-TransUNet: Dual swin transformer U-Net for medical image segmentation. *IEEE Trans. Instrum. Meas.*, 71:1–15, 2022.

[21] C Liu and H Kiryu. 3D medical axial transformer: A lightweight transformer model for 3D brain tumor segmentation. In *Medical Imaging with Deep Learning*, pages 799–813. 2024.

[22] Tinghuai Ma, Qian Pan, Huan Rong, Yurong Qian, Yuan Tian, and Najla Al-Nabhan. T-BERTSum: Topic-aware text summarization based on BERT. *IEEE Trans. Comput. Soc. Syst.*, 9(3):879–890, June 2022.

[23] Himashi Peiris, Munawar Hayat, Zhaolin Chen, Gary Egan, and Mehrtash Harandi. A robust volumetric transformer for accurate 3D tumor segmentation. November 2021.

[24] Md Mostafijur Rahman and Radu Marculescu. Multi-scale hiERarchical vision transformer with cascaded attention decoding for medical image segmentation. March 2023.

[25] Yucheng Tang, Dong Yang, Wenqi Li, Holger Roth, Bennett Landman, Daguang Xu, Vishwesh Nath, and Ali Hatamizadeh. Self-supervised pre-training of swin transformers for 3D medical image analysis. November 2021.

[26] Taoling Tian, Chai Song, Jin Ting, and Hongyang Huang. A french-to-english machine translation model using transformer network. *Procedia Comput. Sci.*, 199:1438–1443, 2022.

[27] Wenxuan Wang, Chen Chen, Meng Ding, Jiangyun Li, Hong Yu, and Sen Zha. TransBTS: Multi-modal brain tumor segmentation using transformer. March 2021.

[28] Xiaolong Wang, Yangyang Qi, Xin Zhou, Geyang Zhang, and Caiyu Fu. Corrigendum to 'alteration of scaffold: Possible role of MACF1 in alzheimer's disease pathogenesis' [med. hypoth. 130 (2019) 109259]. *Med. Hypotheses*, 136(109509):109509, March 2020.

[29] Yiqing Wang, Zihan Li, Jieru Mei, Zihao Wei, Li Liu, Chen Wang, Shengtian Sang, Alan Yuille, Cihang Xie, and Yuyin Zhou. SwinMM: Masked multi-view with swin transformers for 3D medical image segmentation. July 2023.

[30] Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, and Yifan Liu. SegViT: Semantic segmentation with plain vision transformers. October 2022.

[31] Shuai Zheng, Zhenfeng Zhu, Zhizhe Liu, Zhenyu Guo, Yang Liu, Yuchen Yang, and Yao Zhao. Multi-modal graph learning for disease prediction. March 2022.

[32] Hong-Yu Zhou, Jiansen Guo, Yinghao Zhang, Xiaoguang Han, Lequan Yu, Liansheng Wang, and Yizhou Yu. NnFormer: Volumetric medical image segmentation via a 3D transformer. *IEEE Trans. Image Process.*, 32:4036–4045, July 2023.

Multi-objective scheduling optimization of manufacturing systems

GUNADIZ Safia[1] and BERRICHI Ali[2]

[1] *Department of Computer Science, Limose laboratory, University M'hamed bouguerra of Boumerdes, Algeria,* s.gunadiz@univ-boumerdes.dz

[2] *Department of Computer Science, Limose laboratory, University M'hamed bouguerra of Boumerdes, Algeria,* ali.berrichi@univ-boumerdes.dz

Abstract

In this paper our objective is to develop a new method to improve scheduling optimization in manufacturing systems. Firstly, we started with a literature search to determine what type of system our study will focus on. Then we studied different types of methods implemented in this area. In the second step we proposed the implementation of a new metaheuristic which is a Multi-objective Grey Wolf Optimizer [11] to solve the job shop scheduling problem with two classical objective functions, Makespan Cmax and Mean flow Time MFT. Our study resulted to publish our paper titled Grey Wolf Optimizer with Multi Step Crossover for Biobjective job shop scheduling problem[4]. After that we propose another way to exploit this algorithm in order to improve its performance. The flowchart and experimental results are mentioned in this paper. We are conducting an experiment based on best solutions and execution time to demonstrate the performance of the proposed approach. The obtained results show that the proposed method is promising.

Keywords: multi-objective optimization, manufacturing system, Metaheuristic, scheduling problem.

1 Introduction

Scheduling is a decision-making process used in many manufacturing and services industries. Job shop scheduling problem JSSP is a well-known NP-hard problem [6]. In fact, optimizing several objectives at the same time can lead to a solution set called non-dominated solutions in Pareto sense. Bi-objective job shop scheduling problem BJSSP can be solved either by exact methods or by meta-heuristic approaches. Exact methods should generate the set of non-dominated solutions, which is the Pareto front, but they are very time consuming. Meta-heuristics have shown their advantage and efficiency in multi-objective optimization by converging to a set of non- dominated solutions close to the Pareto front.

The JSSP have received a great attention regarding its applications in real world situations. It has been the subject of several works such as Demming et al.[7] where the authors presented a Pareto archive particle swarm optimization for BJSSP. The problem has been converted into a continuous problem and the results have been compared with those of Multiobjective Particle Swarm Optimizer MOPSO and Strength Pareto Evolutionary Algorithm SPEA 2. Majid K et al. [5] considered sequence dependent setup times as constraint in BJSSP. Maximum tardiness is also an often considered criterion with Cmax simultaneously. Hamed and Kuan [12] used Non Dominated Sorting Genetic Algorithm NSGA-II [2] to have a set of quality non-dominated solutions. Suresh et al.[14] propose Pareto archived simulated annealing PASA to minimize Cmax and MFT. Junqing et al. [8] presented a taboo search algorithm with a neighborhood structure based on the critical path theory to minimize Cmax and the total time spent on operations. In the same framework, Qiaofeng et al. [10] propose a hybridization of genetic algorithms with a local search method based on taboo search and simulated annealing. The MOGWO algorithm proposed by Mirjalili et al. [11] is mainly inspired by the hunting behavior of grey wolves. A state of the art of the application of MOGWO in optimization problems is given in [3].Moreover, in scheduling problems, a hybrid MOGWO algorithm is developed by Zhi Yang et al. [15] and G.M. Komakia et al. [6] for flow shop problem. Yuesheng Luo et al. [9] developed MOGWO for Green JSSP with Machines at Different Speeds. Zhenwei Zhu et al. [16] proposed GWO for multi- objective flexible job shop scheduling problem with hierarchical job precedence constraints. An algorithm based on GWO is developed by Saisumpan Sooncharoen et al. [13] to solve production scheduling for the capital goods industry. In [4], MOGWO algorithm is developed to solve a Bi-objective job shop scheduling problem taking into account the C_{max}(F1) and MFT (F2) criteria. The objective functions are formulated as

follows:

$$F1 = max_i = \{C_i, i = 1, ..., n\} \tag{1}$$

$$F_2 = \frac{1}{2}\sum_{i=1}^{n} C_i \tag{2}$$

Where n is the number of jobs to be scheduled on m machines. And Where C_i is the completion time of the i^{th} job.

2 Our contribution

The flowchart of proposed algorithm is presented in Figure 1 Initial population is carried out in two

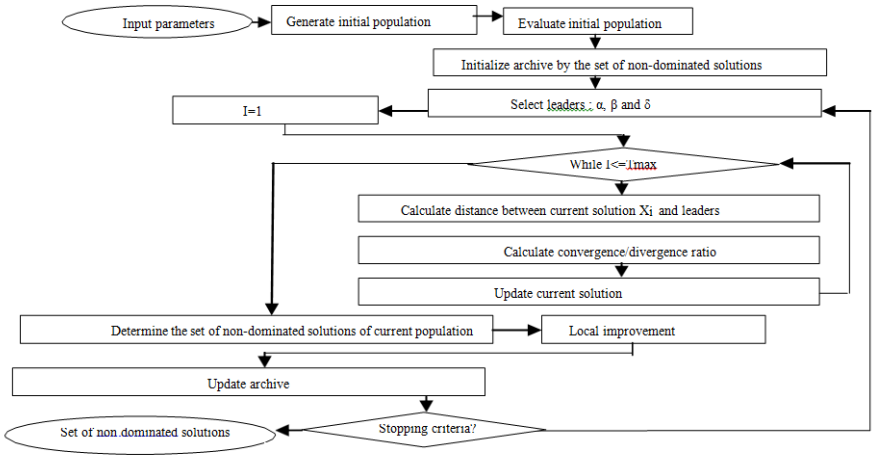


Figure 1: Flowchart MOGWO with MSX

steps. Let T_{max} be the number of solutions in initial population. In the first step, 50% of solutions are randomly generated. Afterwards, a rotation operator is applied on the solutions randomly generated. In order to encode solutions, an operation based representation is proposed. That means the search space is composed of discontinuous decision variables. For this, the Hamming distance is introduced to replace the Euclidean distance and the new concept of average convergence/divergence ratio is defined. This determines the wolves movement direction to the nearest or toward the farthest leader at the next iteration. The wolves position at the next iteration is determined by Multi step crossover MSX or multi step mutation MSM. An axample of solutions encoding is illustrated in Figure 2, with a problem of 3 jobs and 3 machines, operations are numbred from 1 to 9. Each three consecutive numbers are associated with a job.

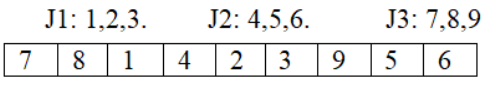


Figure 2: Feasible solution

For performance improvement of the proposed algorithm, a local search method based on a Simulated Annealing algorithm is hybridized with MOGWO. The proposed algorithm is tested on three sets of problem instances from benchmarks from OR Library [1]. And it is compared to two algorithms PASA[14] and Hybrid genetic algorithm HGA [09] in terms of best solutions, coverage metric C and dispersion spread delta of non- dominated solutions on the Pareto front. The results show that the proposed algorithm achieves a significant improvement in terms of the quality of solutions. The C metric shows

that the non-dominated solutions generated by MOGWO are better than or equal in 55% and 94% of test problems than those given by PASA and NSGA-II respectively. The comparison in terms of non-dominated solutions diversity shows that the proposed MOGWO gives dispersed solutions on the Pareto front close to those given by NSGA-II.

However, in terms of complexity, therefore the execution time, the proposed algorithm consists of MOGWO operators which direct the search for optimal solutions, the genetic operators MSX and MSM which explore and exploit the search spaces and finally the local search algorithm to improve the archives of the current iteration. This structure not only increases the efficiency of the algorithm but also its complexity, particularly with large problems. To deals with this problem we propose the second contribution.

In this proposition we use two populations at the same time, one codes solutions in real space and the second in discrete space which is scheduling set. Each solution in real space has corresponding solution in discrete space. The transformation of real solutions to discrete solutions is encoding process explained in the next paragraph. Algorithm flowchart is shown on Figure 4. The encoding process is divided on following steps:

1. Generate random NxM values which are real solution.
2. Arrange the NxM values of real solution in descending order.
3. Record, in discrete solution, the ranks of each element according to the sorting results of the step 2.
4. Order the operations of each job to obtain a feasible solution.

This process is used to generate initial population and repeated, for all real solutions generated by MOGWO operators, to transform them to discrete solutions on each iteration.

The encoding solution is illustrated in the following Figure 3.

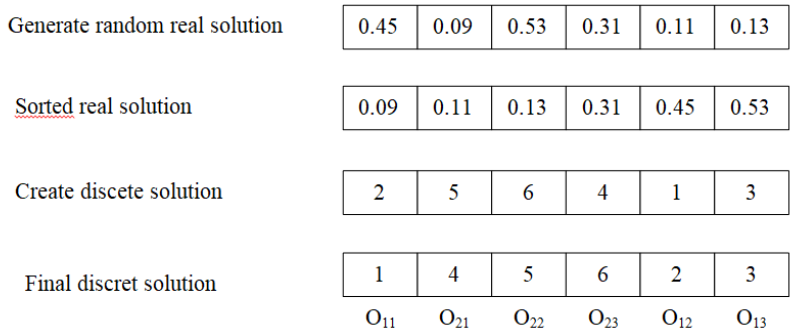


Figure 3: Example of encoding solution

Let a scheduling problem of 2jobs and 3 machines, the encoding process is shown in Figure 4.

In order to evaluate performance of proposed algorithm, we execute it 30 times using some benchmarks from OR Library [17], and then we compare its results with those mentioned in [4]. The experimental results are shown in table 1.

3 Conclusion

In our study, we utilized the MOGWO algorithm in two distinct approaches to address the job shop scheduling problem. Our results indicate the efficacy of this algorithm, as we employed it both in its original form and enhanced its performance by hybridizing it with local search methods.

In our ongoing research, we aim to further enhance the algorithm's performance for tackling the job shop scheduling problem using other type of encoding solutions us encode base machine and using other selection mechanism to select leaders.

References

[1] http://people.brunel.ac.uk/ mastjjb/jeb/orlib/files/jobshop1.txt, 2024. Consulté le Avril 2024.

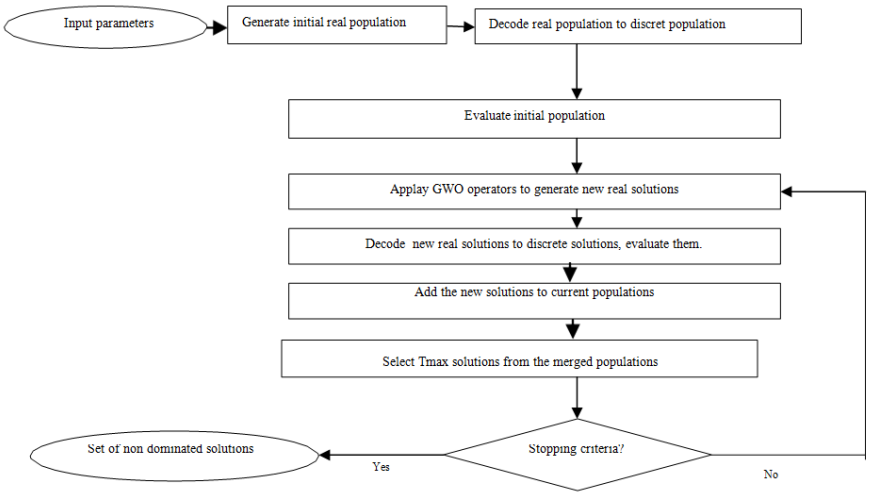


Figure 4: Flowchart of MOGWO with two populations.

Table 1: Comparison of two proposed algorithms in terms of objective functions and executing time

	GWO with MSX			GWO with two populations		
	C_{max}	MTFT	Run time (sec)	C_{max}	MTFT	Run time(sec)
FT 06	55/55	045.167	1424.42	55	046.000	308.028
FT10	956	0790.300	18315.3	1012	838.000	17088.7
LA 01	666	0420.000	1801.64	666	436.300	320.077
LA 06	926	0498.733	5839.17	926	555.600	
LA 11	1222	0613.650	5839.17	1222	720.700	594.015
LA 16	946	755.300	111334	979	820.100	587.392
LA21	1121	374.600	17827.1	1165	391.533	1458.2
LA26	1419	1010.800	81612.3	1352	1093.050	15926.5
LA31	1794	1240.730	104080	1848	1400.870	12054
LA36	1337	1188.470	51323.1	1385	1244.000	1795.64

[2] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multi-objective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[3] Hossam Faris, Ibrahim Aljarah, Mohammed Azmi Al-Betar, and Seyedali Mirjalili. Grey wolf optimizer: a review of recent variants and applications. *Neural computing and applications*, 30:413–435, 2018.

[4] Safia Gunadiz and Ali Berrichi. Grey wolf optimizer with multi step crossover for bi-objective job shop scheduling problem. In *International Conference on Computing Systems and Applications*, pages 261–272. Springer, 2022.

[5] Majid Khalili and Bahman Naderi. Multi-objective job shop scheduling problem with sequence dependent setup times using a novel metaheuristic. *International Journal of Intelligent Engineering Informatics*, 2(4):243–258, 2014.

[6] GM Komaki and Vahid Kayvanfar. Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. *Journal of computational science*, 8:109–120, 2015.

[7] Deming Lei. A pareto archive particle swarm optimization for multi-objective job shop scheduling. *Computers & Industrial Engineering*, 54(4):960–971, 2008.

[8] Junqing Li, Quanke Pan, Shengxian Xie, Kaizhou Gao, and Yuting Wang. A hybrid algorithm for multi-objective job shop scheduling problem. In *2011 Chinese Control and Decision Conference (CCDC)*, pages 3630–3634. IEEE, 2011.

[9] Yuesheng Luo, Chao Lu, Xinyu Li, Ling Wang, and Liang Gao. Green job shop scheduling problem with machine at different speeds using a multi-objective grey wolf optimization algorithm. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 573–578. IEEE, 2019.

[10] Qiaofeng Meng, Linxuan Zhang, and Yushun Fan. Approach of hybrid ga for multi-objective job-shop scheduling. *International Journal of Modeling, Simulation, and Scientific Computing*, 7(04):1643006, 2016.

[11] Seyedali Mirjalili, Shahrzad Saremi, Seyed Mohammad Mirjalili, and Leandro dos S Coelho. Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. *Expert systems with applications*, 47:106–119, 2016.

[12] Hamed Piroozfard and Kuan Yew Wong. Solving multi-objective job shop scheduling problems using a non-dominated sorting genetic algorithm. In *AIP Conference Proceedings*, volume 1660. AIP Publishing, 2015.

[13] Saisumpan Sooncharoen, Pupong Pongcharoen, and Christian Hicks. Grey wolf production scheduling for the capital goods industry. *Applied Soft Computing*, 94:106480, 2020.

[14] RK Suresh and KM Mohanasundaram. Pareto archived simulated annealing for job shop scheduling with multiple objectives. *The International Journal of Advanced Manufacturing Technology*, 29:184–196, 2006.

[15] Zhi Yang and Cungen Liu. A hybrid multi-objective gray wolf optimization algorithm for a fuzzy blocking flow shop scheduling problem. *Advances in Mechanical Engineering*, 10(3):1687814018765535, 2018.

[16] Zhenwei Zhu and Xionghui Zhou. An efficient evolutionary grey wolf optimizer for multi-objective flexible job shop scheduling problem with hierarchical job precedence constraints. *Computers & Industrial Engineering*, 140:106280, 2020.

Evolutionary intelligence: Exploring Genetic Algorithm involvement in AI and beyond

Salaheddine Bougouffa[1] **and Menouar Boulif**[2]

[1] ***LIMOSE laboratory, Department of computer science, M'hamed Bougara University Boumerdes, Algeria,*** *s.bougouffa@univ-boumerdes.dz*

[2] ***LIMOSE laboratory, Department of computer science, M'hamed Bougara University Boumerdes, Algeria,*** *boumen7@gmail.com*

Abstract

Evolutionary intelligence (EI) has advanced significantly, spreading into many fields and changing how problems are solved. This paper gives an overview of how Genetic Algorithms (GAs), an active research topic in EI, are used in various areas. Looking at examples from different fields, this paper demonstrates how GAs, by leveraging their flexible and exploratory abilities, can enhance the search for problem solutions. Furthermore, we explore how combining and/or integrating GAs into Artificial Intelligence techniques guides them even more effectively, resulting in improved performance to solving complex problems.

Keywords: Optimization, Complex problem solving, Evolutionary Intelligence, Genetic Algorithms.

1 Introduction

Genetic Algorithms (GAs), an active field of evolutionary intelligence (EI) [16], are a class of optimization and stochastic search algorithms. GAs in their modern form were introduced by JH. Holland in 1975 [7] and, thanks to the sustaining efforts of his PhD students like K. De Jong [4], DE. Goldberg [6] and M. Mitchell [13], have become one of the most abundant research-work-triggers in evolutionary intelligence.

GAs draw inspiration from the evolutionary process observed in the realm of living species, where natural selection favors the survival of individuals that are best adapted to their environment ("survival of the fittest"). In this process, advantageous characteristics are passed on to the next generations through genetic heredity during reproduction, contributing to the population's improved adaptive capabilities over time. Hence, best adapted individuals are more likely to be selected for reproduction, and to pass on their promising traits to their descendants.

This cycle of reproduction and transmission of advantageous characteristics leads to the emergence of optimized solutions within the population, thus mimicking the mechanism of natural evolution.

Through this iterative process of selection, reproduction, and mutation, GAs are capable of efficiently exploring complex solution spaces and finding satisfactory answers to various optimization problems.

GAs have recently found new applications when combined with AI algorithms, resulting in improved optimization outcomes. In order to explore this new research trend, this paper aims to expose the main new applications that combines GAs with AI capabilities and beyond.

The remainder of the paper will be organized as follows. In section I, we present various applications of GAs in different areas. Then, the next section, we introduce diverse techniques for integrating GAs with AI, emphasizing their effectiveness in enhancing the overall performance.

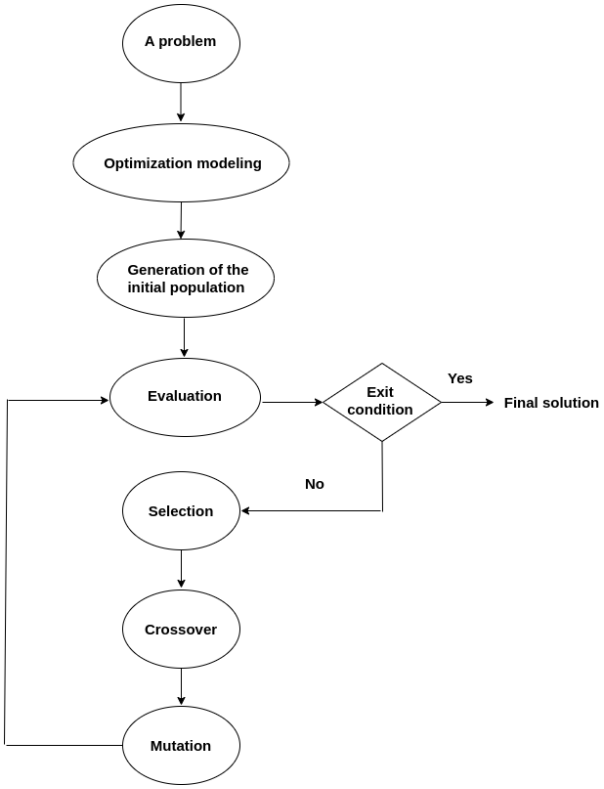


Figure 1: GA flowchart.

2 Applications of Genetic Algorithms

GAs have played a pivotal role in optimization since their development, leading to their integration into a wide range of AI applications. Their appealing way of modeling problems with chromosomes has contributed to their widespread use across different fields. Today, genetic algorithms remain a crucial and versatile tool in the optimization toolbox, contributing to advancements in various areas such as robotics, game design, forecasting, prediction, computer vision, and many others.

2.1 GAs involvement in optimization

The historical purpose of genetic algorithms was the optimization of systems, a role for which they were specifically designed. This versatility has made GAs invaluable across numerous fields, including transport networks optimization, supply chain management, image processing, renewable energy, pattern recognition, automotive design, and many others [1, 11].
Besides, GAs are frequently combined with various optimization techniques to enhance their effectiveness in exploring the search space and finding high-quality solutions. One commonly utilized strategy involves integrating GAs with local search techniques such as Hill climbing and Gradient descent. These techniques are employed to refine the solutions obtained by GAs, ensuring that the optimization process converges towards optimal or near-optimal solutions. Additionally, local search techniques play a crucial role in addressing the issue of disrupting building blocks within the genetic representation of solutions [9]. Traditional genetic operators like crossover and mutation may inadvertently break apart advantageous structures or patterns known as building blocks. By incorporating local search techniques, GAs can mitigate this disruption, promoting the preservation of advantageous structures throughout the optimization process.
Furthermore, GAs can be combined with metaheuristic optimization techniques like simulated annealing and ant colony optimization to facilitate the exploration of the search space, enhancing exploration-exploitation balance and improving the overall efficiency of the optimization process.

2.2 GAs involvement in machine learning

Genetic algorithms have found a valuable application in machine learning. In this context, GAs play a crucial role in optimizing the parameters of machine learning models [18], such as neural networks and support vector machines (SVM). By employing GAs, these models can be fine-tuned to improve their performance in recognizing complex patterns within datasets. This optimization process involves searching through a large space of possible parameter configurations, where GAs excel due their ability to handle high-dimensional and nonlinear search spaces. As a result, GAs have become an effective tool for enhancing the accuracy and efficiency of machine learning algorithms in various applications, including pattern recognition, natural language processing and data mining.

2.3 GAs involvement in robotics

Genetic Algorithms offer an effective approach to solving a wide range of robotics problems. GAs can optimize robot control and generate trajectories in dynamic environments considering many constraints such as obstacle avoidance, joint limits, terrain, and energy efficiency [14].

2.4 GAs involvement in game design

Genetic Algorithms can be highly beneficial in designing games, as they enable the evolution of strategies for game agents, ensuring character development and game balance [12]. By using GAs, game designers can create dynamic and challenging gameplay experiences where the strategies of non-player characters evolve over time. This approach enhances player engagement and satisfaction by providing a more immersive and adaptive gaming environment. Additionally, GAs can be used to fine-tune game parameters, such as difficulty levels or resource allocation, to create a balanced and enjoyable gaming experience.

2.5 GAs involvement in forecasting and prediction

GAs find application in forecasting or prediction tasks spanning diverse domains, including finance, economics, weather forecasting, and stock market prediction. They are employed in conjunction with a range of techniques such as time series analysis, statistical modeling, simulation, Monte Carlo methods, and optimization techniques. In fact, GAs can effectively complement statistical models [19] to improve forecasting, prediction, and optimization tasks. GAs optimize the parameters of statistical models and aid in variable selection by identifying the subset of predictors with the most significant impact on the outcome variable, thereby reducing model complexity. Additionally, GAs facilitate the comparison of statistical models by evaluating each model with different assumptions and tasks, helping to determine the most suitable model for a given prediction task.

2.6 GAs involvement in computer vision

Genetic Algorithms enjoy various applications in computer vision tasks, including feature selection, image classification, object detection, video analysis, and image segmentation. The integration of GAs in computer vision techniques facilitates the development of adaptive, robust, and efficient vision systems capable of addressing a wide range of image analysis and understanding tasks. By harnessing the search and optimization capabilities of GAs, practitioners can enhance the performance, accuracy, and reliability of computer vision algorithms across diverse applications and domains.

2.7 GAs involvement in Simulation-Optimization frameworks

Genetic Algorithms can be integrated into hybrid simulation-optimization frameworks, where simulation models are used to evaluate the performance of candidate solutions generated by the genetic algorithm [10]. This integration allows for the incorporation of complex system dynamics and constraints into the optimization process, leading to more realistic and effective solutions.

3 Techniques combining GAs and AI

In this section, we delve into the details of how GAs have been combined with AI, highlighting the benefits of this integration in enhancing the overall performance of AI algorithms.

3.1 Optimization of AI algorithms parameters

GAs provide a potent method for improving the effectiveness of AI algorithms, especially in optimizing parameters such as the number of layers, neurons per layer, types of connections in neural networks [8, 5, 18, 17], and the parameters associated to the regularization or the kernel functions in support vector machines. GAs facilitate the evolution of neural network structures and architectures by systematically exploring various parameter configurations. The solution pool comprises potential combinations of specific parameters, which are evaluated for their effectiveness, and are iteratively refined to maximize performance. Consequently, GAs play a significant role in finetuning AI algorithms and improve their performance.

3.2 Hybridization of GAs with machine learning

GAs can be effectively combined with various machine learning techniques, such as neural networks or support vector machines (SVMs) [8]. One notable advantage of this hybridization is the ability to address high-dimensional data by reducing its dimensionality while simultaneously improving performance. In these hybrid approaches, machine learning models, such as neural networks or SVMs, play a crucial role in guiding the search process within the GA framework. Firstly, these models assist in preprocessing tasks by reducing the dimensionality of the input data, which is especially beneficial for large-scale datasets with numerous features. By extracting relevant features or representations from the data, the machine learning component helps streamline the subsequent optimization process conducted by the GA.

3.3 Fuzzy Genetic Algorithms

Fuzzy genetic algorithms merge fuzzy logic with genetic algorithms, in order to incorporate uncertain and imprecise concepts found in many real-life problems [2, 3, 15]. By using fuzzy sets and rules, fuzzy GAs evolve candidate solutions over generations while considering fuzzy characteristics. These algorithms excel in optimizing fuzzy rule-based systems for tasks like control and decision-making. Their versatility extends to domains like data mining, control systems, and pattern recognition, thanks to their adaptability and learning capabilities.

4 Conclusion

This paper examined the integration of Genetic Algorithms with Artificial Intelligence. Through a review of existing applications, we emphasized the significance of this combination in conducting efficient space exploration, resulting in enhanced performance. The fusion of GAs and AI enhances problem solving by leveraging GAs' ability to explore vast solution spaces quickly, in contrast to other heuristic algorithms. Furthermore, what sets GAs apart is their ease of integration within any search algorithm and their potential for hybridization with other techniques, making them essential for advancing AI systems further. As technology progresses, the utilization of GAs with AI presents new opportunities for problem-solving and innovation across diverse fields.
As future research direction, we recommend to explore GA-AI combinations to deal with more complex problems having simultaneous challenging features such as multi-objective and real-time applications.

References

[1] Saber Hadj Abdallah and Souhir Tounsi. Optimal electric vehicle design tool using genetic algorithms. *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, 11(07-11-02-0010):109–122, 2018.

[2] Basma Alouane and Menouar Boulif. Fuzzy constraint prioritization to solve heavily constrained problems with the genetic algorithm. *Engineering Applications of Artificial Intelligence*, 119:105768, 2023.

[3] Menouar Boulif and Karim Atif. A new fuzzy genetic algorithm for the dynamic bi-objective cell formation problem considering passive and active strategies. *International Journal of Approximate Reasoning*, 2008.

[4] Kenneth De Jong. Learning with genetic algorithms: An overview. *Machine learning*, 3:121–138, 1988.

[5] Jenny V Domashova, Sofia S Emtseva, Vladislav S Fail, and Aleksandr S Gridin. Selecting an optimal architecture of neural network using genetic algorithm. *Procedia Computer Science*, 190:263–273, 2021.

[6] David E Golberg. *Genetic algorithms in search, optimization, and machine learning.* Addion Wesley, 1989.

[7] JH Holland. Adaptation in natural and artificial systems. ann arbor: University of michigan press. *Ann Arbor: The University of Michigan Press*, 1975.

[8] Yan Hu, Bingce Wang, Yuyan Sun, Jing An, and Zhiliang Wang. Genetic algorithm–optimized support vector machine for real-time activity recognition in health smart home. *International Journal of Distributed Sensor Networks*, 16(11):1550147720971513, 2020.

[9] Hitoshi Iba, Hugo de Garis, and Taisuke Sato. Genetic programming with local hill-climbing. In *International Conference on Parallel Problem Solving from Nature*, pages 302–311. Springer, 1994.

[10] S Khebbache-Hadji, Y Hani, N Lahiani, and A El Mhamedi. Genetic algorithm used in simulation model: Application in a maintenance process. *IFAC Proceedings Volumes*, 45(6):1047–1052, 2012.

[11] Florencia Lazzari, Gerard Mor, Jordi Cipriano, Francesc Solsona, Daniel Chemisana, and Daniela Guericke. Optimizing planning and operation of renewable energy communities with genetic algorithms. *Applied Energy*, 338:120906, 2023.

[12] Robert E Marks. Playing games with genetic algorithms. In *Evolutionary computation in economics and finance*, pages 31–44. Springer, 2002.

[13] Melanie Mitchell. *An introduction to genetic algorithms.* MIT press, 1998.

[14] Gihan Nagib and Wahied Gharieb. Path planning for a mobile robot using genetic algorithms. *IEEE Proceedings of Robotics*, 185189, 2004.

[15] Arup Kumar Nandi. Ga-fuzzy approaches: application to modeling of manufacturing process. In *Statistical and computational techniques in manufacturing*, pages 145–185. Springer, 2012.

[16] Sai Sumathi, T Hamsapriya, and P Surekha. *Evolutionary intelligence: an introduction to theory and applications with Matlab.* Springer Science & Business Media, 2008.

[17] Peilin Wang, Kuangkuang Ye, Xuerui Hao, and Jike Wang. Combining multi-objective genetic algorithm and neural network dynamically for the complex optimization problems in physics. *Scientific Reports*, 13(1):880, 2023.

[18] QJ Wang. Using genetic algorithms to optimise model parameters. *Environmental Modelling & Software*, 12(1):27–34, 1997.

[19] Zhongheng Zhang, Victor Trevino, Sayed Shahabuddin Hoseini, Smaranda Belciug, Arumugam Manivanna Boopathi, Ping Zhang, Florin Gorunescu, Velappan Subha, and Songshi Dai. Variable selection in logistic regression model with genetic algorithm. *Annals of translational medicine*, 6(3), 2018.

Formal Methods for Internet of Things: a Concise Classification

Ibtissem Talamali[1], Razika Lounas[1], and Mohamed Mezghiche[1]

[1] ***LIMOSE Laboratory, Computer Science Department, Faculty of Sciences University of M'hamed Bougara of Boumerdes, Independency Avenue, 35000 Algeria, i.talamali@univ-boumerdes.dz***

Abstract

The Internet of Things (IoT) has now become a key technology that can span several technology areas, from data discovery and processing to networking and data analysis. It is used in many applications ranging from home security and factory automation to healthcare provision and autonomous driving. Many IoT devices can connect and communicate at the same time, and this exchange of data enables better decision-making in an increasingly complex environment. However, for some safety-critical systems, any failure of a function can have very serious consequences. This is why we need to adopt appropriate and effective testing techniques to detect errors and flaws in the system design and correct them as soon as possible. The formal method is one of the crucial methods for detecting possible weaknesses and vulnerabilities at an early stage of the design in order to verify the correctness of the system. This article provides an overview of the use of model checking and theorem proving to establish correctness properties on IoT systems.

Keywords: Internet of Things, formal methods, model checking, theorem proving.

1 Introduction

The Internet of Things (IoT) is a new paradigm aimed at creating connectivity for "everything" that can support minimal storage and processing power. This allows these connected elements to work together anywhere, anytime and in any form within an application to cover different domains, health, transport infrastructure, smart home, smart shopping, e-commerce, etc. Ensuring the accuracy, reliability and the correctness of IoT systems is critical to advancing IoT projects. Sufficient verification before the actual introduction of the IoT system is very important to discover and improve system design errors and flaws as soon as possible. Moreover, some of these errors may cause catastrophic loss of money, time, or even human life. To help overcoming such problems, it has been suggested to use formal methods in the development of critical systems. Formal methods is a method using mathematic based languages, techniques, and tools for specifying and verifying such systems. It is an important means to improve the system safety and reliability. In fact, in various stages of development trend of formal methods gradually integrated into the software development process, from the demand analysis, function description (description), (Architecture / design, programming algorithm), testing and maintenance[15]. As an efficient means of pre-deployment inspection of Internet of Things systems, formal methods have received widespread attention in recent years[25].
In this paper, we provide a survey of the application of formal methods in Internet of Things systems. The objective of this paper is three fold:

- classify research papers according to the used formal techniques;
- gaining insight about the objective of applying formal methods on IoT through the established properties;
- draw recommendations about a future use at the light of the surveyed papers.

The rest of this paper is organized as follows. Section 2 presents the related Work. Internet of Things with architectures and protocols are presented in Section 3. In Section 4 the use of formal methods for IoT is described. The conclusion and discussion is given in Section 5.

2 Related work

Several researchers endeavored to analyse the application of formal methods for Internet of Things systems. In [11], the authors reviewed researches about the formal verification of IoT protocols. The authors

distinguished the main objectives for the formal verification: functional checks, security properties and gave suggestions for enhanced schemes and implementation checks of protocols."

In [6], the authors presented tools used in formal verification for distributed systems. These tools: Isabelle/HOL, Coq, Verdi, and TLA+, are compared in terms of functionality, interface, and application. The authors gave some recommendations of use according to the aim of formalisation. In [12], the authors proposes to formalize things information in three theories: graphs, sets, and abstract expressions. The description is related to things information in several situations such as coded by UID, stored in RFID. In [20], the authors studied the use of formal verification in IoT from an application point of view. The survey considered model checking , process algebraic, and automated theorem proving in different application areas such as monitoring, health, and protocols. The survey find out that security issues are the most studied issue with model checking. Existing surveys helped us in the assimilation of formal methods and techniques in the context of IoT, but there is an evident lack in response to the increasing need to establish formal correctness using theorem proving with regard to model checking. Our survey complements the existing surveys by bringing new points to the discussion about the established properties using theorem proving.

3 Internet of Things in Critical Applications

In recent years, several definitions of the Internet of Things have appeared. The ITU (International Telecommunication Union) defines IoT as:" a global infrastructure for the information society that enables advanced services by interconnecting objects (physical or virtual) through existing or evolving interoperable information and communication technologies". This section details the concept of IoT through its architecture, protocols, and applications.

3.1 IoT Architecture

Due to the fast development of IoT, it became essential to have a reference architecture that could standardize system design and facilitate communication and interoperability between different IoT ecosystems [18]. IoT architecture design involves many factors such as networking, communication, processes, etc. Scalability, extensibility, and interoperability amongst devices must all be taken into consideration while creating the IoT architecture. Due to the fact that things can move and need to interact with others in real time, the IoT architecture should be adaptive to make devices interact with others dynamically and support communication between them. In addition, the IoT should possess the decentralized and heterogeneous nature[7]. There is no single consensus on which IoT architecture is universally adopted. Different architectures have been proposed by different researchers[3]. The layered architecture of the Internet of Things is illustrated as assumed by the ITU-T (International Telecommunication Union - Telecommunications Standardization Sector) and is composed of four layers(Figure **??**).

1. **Sensor Layer (Device Layer/ Perception Layer)** : It consists of data sensors in various forms such as RFID tags, IR sensors or other sensor networks that could detect temperature, humidity, speed and location, etc. This layer collects useful information from objects from related sensors and converts the information into digital signals that are then transmitted to the network layer for further action[5].

2. **Network Layer** This layer is responsible for the reliable transmission of data generated in the perception layer as well as the assurance of connected inter-object connectivity and between smart objects and other Internet hosts. On the other hand, a massive volume of data will be produced by these tiny sensors, which requires a robust and efficient wired or wireless network infrastructure as a means of transport[16].

3. **Service and Middleware Layer** This layer receives data from the Network layer. Its purpose is service management and data storage. It also processes information and makes decisions automatically based on the results and passes the output to the next layer[13].

4. **Application Layer** This layer provides the different types of services requestd by the customer that depends on his specific use case adopted. For example, if the smart home is the use case, the customer may request specific parameters such as heating, ventilation and air conditioning (HVAC) measurements or temperature and humidity values[13].

3.2 IOT Applications domains

The potential applications of IoT are numerous reaching every area of the daily lives of individuals, businesses and society as a whole. This section provides an overview of the main application areas of IoT.

- **Smart Cities** A Smart City is a city that monitors and integrates the conditions of all critical infrastructure including road bridges, tunnels, rail/metro, airports, seaports, even large buildings, etc. Structural health, Digital video surveillance, fire management, intelligent and weather-adaptive lighting are examples of smart cities applications.
- **Smart Health** The IoT plays an important role in the health sector, putting in place new technologies not only in hospitals, but also in the workplace and at your fingertips, whether to keep track of medical records, monitor vital signs or treat remotely. There are some IoT technologies in this field[16] : Patient monitoring, medical refrigerators control, pharmaceuticals monitoring, and chronic disease management systems.
- **Smart industry** The Industrial Internet of Things (IIoT) or the factory of the future is the fourth industrial revolution or Industry 4.0 (started in Germany, 2010). It is basically characterized by intelligent automation and integration of new technologies in factories such as sensors and smart tools in general allow collecting more data about the manufacturing process to check compliance and optimize production in real time[14].
- **Smart Agriculture** The automation of agricultural events is moving the agricultural sector from a static and manual situation to a dynamic and intelligent automation, to facilitate farmer's tasks, such as the irrigation, fertilizer application and others, resulting in improved production with reduced human efforts. Some of the objectives of IoT usage in this sector include: Control of the microclimatic conditions to maximize the production of fruits and vegetables and their quality, control humidity and temperature levels to prevent microbial contaminants, and identification of animals grazing in open pastures.
- **Smart energy** The IoT allows the countless devices that make up the electricity grid to share information in real time to improve the efficiency of energy distribution and management, such as [16] : monitoring and analysis of wind turbine and power plant energy flows, and AC-DC power control, and monitoring and optimisation of solar power plant performance.

4 Formal methods and internet of Things

In complex systems, it is very important to ensure that there are no dangerous or unexpected behaviors. Therefore, errors need to be identified early in the system lifecycle. In such cases, formal methods have proven to be the most appropriate technique to ensure and guarantee the absence of bugs and defects[9]. A formal method is a mathematically-based technique and formal logic used in computer science to describe properties of hardware and/or software systems. A method is formal if it has a sound mathematical basis, typically given by a formal specification language[22]. This basis provides the means of precisely defining notions like consistency and completeness and, more relevantly, specification, implementation, and correctness. It provides the means of proving that a specification is realizable, proving that a system has been implemented correctly, and proving properties of a system without necessarily running it to determine its behavior[23]. In fact, there are several formal languages and techniques that allow different types of properties to be inspected at different levels of the development process.

4.1 Formal verification

Formal Verification is a promising method to provide security guarantees by mathematically ascertaining the correctness of designs using a diverse set of mathematical and logical methods. These methods are particularly useful in order to get quantitative statements about safety and security properties of digital systems[11]. There are two major state of the art approaches to formal verification: theorem proving and model checking.

1. **Model checking** Model checking is an automated approach to verify that a model of a finite state system satisfies a formal specification of requirements to the system. In this approach the models describe how the state of the system may evolve over time, and the requirements are some constraints on how the state of the system is allowed to evolve over time. Tools that automatically perform model checking are called model checkers[10]. In other way, the user specifies the system through a set of states connected by a set of transitions. Then, an algorithm is executed to enumerate the possible execution states of the system. This algorithm verifies if the model satisfies the different properties[9]. Among the model checking tools, we mention: SPIN, DIVINE, PAT, and UPPAAL. Now, we describe some related work in current literature. In [25], the authors proposes a hierarchical formal modeling approach for IoT systems that focuses on user behavior and improves pre-deployment correctness checking and reliability analysis. The hierarchical modeling of the three-layer architecture of the IoT system perception layer, middle layer, and application layer has been completed, and the model verification tool PAT was used to analyze and verify the model from aspects such as security, accessibility, and system consistency.

   In [19], the authors proposed a formal verification of interoperability in IoT systems. The proposal answers the question about consistency of IoT solutions in terms of interoperability. The system is formalized in terms of web services concepts. The authors used TLA+ specification and the TLA Checker to establish the interoperability property. In [24], the authors proposed a probabilistic model checking approach for run-time verification of industrial IoT. The approach combines sensor level and data-driven models to establish the property of quantified trustworthiness of sensors through the study of several types of sensor data faults. The framework is evaluated using a CNC turn-mill machine with the PRISM tool.

2. **Theorem proving**

   Theorem proving is a technique by which both the system and its desired properties are expressed as formulas in some mathematical logic. This logic is given by a formal system, which defines a set of axioms and a set of inference rules. Theorem proving is the process of finding a proof of a property from the axioms of the system. Steps in the proof appeal to the axioms and rules, and possibly derived definitions and intermediate lemmas. Theorem provers are increasingly being used today in the mechanical verification of safety-critical properties of hardware and software designs[2]. Examples of some notable proof checkers are MetaMath and Mizar. Examples of some notable interactive theorem provers are the PVS, Isabelle/ HOL, ACL2 and Coq. The rest of this part, we outline a few related works in the current literature.

   In [17] The authors define a formal model of real-time networks, in the field of embedded networks, based on the NC theory (computational network) a method of analysis of temporal properties based on the algebra of min-plus dioids. And for that, they formalized the proofs related to the network calculation in Coq to ensure guaranteed delays in embedded real-time networks. In this article [8], the authors propose a formal approach for the validation and certification of smart city systems by formalizing them as cyber-physical systems. These have been formalized as finite state machines and interpreted and formally verified by the Coq proof assistant. this article show that the Coq proof assistant plays an essential role in software validation in the smart cities domain. In this work [1], The authors present the IoT Conflict Checker (IoTC2) as a formal method to ensure the safe behavior of controllers and actuators in IoT systems. The study includes the definition of security policies, their implementation in Prolog for logical completeness, the implementation of detection policies in the Matlab Simulink environment, creating an intelligent home environment in Simulink to demonstrate conflict and test scaling, efficiency and accuracy in a simulated environment. In [21] the authors proposed a refinement-based approach for modeling IoT design patterns, which takes advantage of formal methods by the specification of design pattern models with the Event-B method. They checked the design correctness and verify properties of IoT design patterns using the model checking to check the correctness of the behavior of the pattern and absence of deadlocks with ProB tool and theorem proving to ensure the consistency of an Event-B pattern model with the Rodin platform. Finally, they illustrated their approach with a case study in healthcare domain.

At the end of this section, we have classified the articles already mentioned according to the formal method used, the tool and the properties checked (Table 1). This classification provides a clear vision of existing approaches from 2016 to 2020 to formally refine and verify IoT systems against a set of criteria.

Years	Approach	Formal method	Tools	Verified properties
2016	[4]	Theorem Proving	HOL	Evaluate the coverage properties
2018	[8]	Theorem Proving	The proof assistant CoQ	System proprieties
2018	[19]	Model checking	TLA Checker	interoperability property
2019	[17]	Theorem Proving	The proof assistant CoQ	Temporal properties
2019	[1]	Theorem Proving	Prolog tool	Security policies properties
2020	[25]	Model checking	PAT tool	Security, Accessibility ,Consistency
2020	[24]	Model checking	PRISM tool	quantified trustworthiness property
2023	[21]	Model checking/ Theorem proving	Rodin/ProB	Structural consistency, behavioral features, absence of deadlocks.

Table 1: Classification of pertinent works about IoT systems formal verification

5 Conclusion

The Internet of Things (IoT) has become an important part of our lives. However, the increasing complexity of IoT systems makes it essential to ensure their reliability and correctness. Formal methods have emerged as a powerful tool for verifying the correctness of these systems. In this paper, we have presented a survey of the use of formal methods, specifically model checking and theorem proving, in IoT systems. We have classified the research papers according to the formal techniques used and discussed the objectives of applying these methods. Our survey has shown that formal methods can significantly improve the safety and reliability of IoT systems. We recommend that formal methods be used in the early stages of IoT system design to identify errors and to ensure their correctness. Finally, we believe that this paper will stimulate further research in the use of formal methods in the development of IoT systems.

References

[1] Abdullah Al Farooq, Ehab Al-Shaer, Thomas Moyer, and Krishna Kant. Iotc 2: A formal method approach for detecting conflicts in large scale iot systems. In *Symposium on integrated network and service management*, pages 442–447. IEEE, 2019.

[2] Edmund M Clarke and Jeannette M Wing. Formal methods: State of the art and future directions. *ACM Computing Surveys (CSUR)*, 28(4):626–643, 1996.

[3] Sakina Elhadi, Abdelaziz Marzak, Nawal Sael, and Soukaina Merzouk. Comparative study of iot protocols. *Smart Application and Data Analysis for Smart Cities (SADASC'18)*, 2018.

[4] Maissa Elleuch, Osman Hasan, Sofiene Tahar, and Mohamed Abid. Formal probabilistic analysis of a wsn-based monitoring framework for iot applications. In *Formal Techniques for Safety-Critical Systems: 5th International Workshop, FTSCS 2016, Tokyo, Japan, November 14, 2016*, pages 93–108. Springer, 2017.

[5] M Umar Farooq, Muhammad Waseem, Sadia Mazhar, Anjum Khairi, and Talha Kamal. A review on internet of things (iot). *International journal of computer applications*, 113(1):1–7, 2015.

[6] Anna Fatkina, Oleg Iakushkin, Dmitry Selivanov, and Vladimir Korkhov. Methods of formal software verification in the context of distributed systems. In *Computational Science and Its Application: 19th International Conference, Saint Petersburg, Russia, July 1–4, Proceedings, Part II 19*, pages 546–555. Springer, 2019.

[7] Pradyumna Gokhale, Omkar Bhat, and Sagar Bhat. Introduction to iot. *International Advanced Research Journal in Science, Engineering and Technology*, 5(1):41–44, 2018.

[8] Erick Simas Grilo and Bruno Lopes. Formalization and certification of software for smart cities. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[9] Marwa Hachicha, Riadh Ben Halima, and Ahmed Hadj Kacem. Formal verification approaches of self-adaptive systems: A survey. *Procedia Computer Science*, 159:1853–1862, 2019.

[10] Anne E Haxthausen. An introduction to formal methods for the development of safety-critical applications. *Technical University of Denmark*, 2010.

[11] Katharina Hofer-Schmitz and Branka Stojanović. Towards formal verification of iot protocols: A review. *Computer Networks*, 174:107233, 2020.

[12] Yinghui Huang and Guanyu Li. Descriptive models for internet of things. In *2010 International Conference on Intelligent Control and Information Processing*, pages 483–486. IEEE, 2010.

[13] Muhammad Azhar Iqbal, Sajjad Hussain, Huanlai Xing, and Muhammad Ali Imran. *Enabling the internet of things: fundamentals, design and applications*. John Wiley & Sons, 2020.

[14] AKSA Karima, Khayreddine BOUHAFNA, Souleymen BELAYATI, and Dina DJEGHAR. Vers une nouvelle révolution industrielle: Industrie 4.0. *Revue Méditerranéenne des Télécommunications*, 11(1), 2021.

[15] Chunlin Kuang and Weiling Li. Application of formalization method in construction zigbee technology and rfid system in internet of things. In *2015 International Conference on Mechatronics, Electronic, Industrial and Control Engineering (MEIC-15)*, pages 874–879. Atlantis Press, 2015.

[16] Keyur K Patel, Sunil M Patel, and P Scholar. Internet of things-iot: definition, characteristics, architecture, enabling technologies, application & future challenges. *International journal of engineering science and computing*, 6(5), 2016.

[17] Lucien Rakotomalala, Marc Boyer, and Pierre Roux. Formal verification of real-time networks. In *JRWRTC 2019, Junior Workshop RTNS 2019*, 2019.

[18] Imad Saleh. Les enjeux et les défis de l'internet des objets (ido). *Internet des objets*, 1(1):5, 2017.

[19] Vadym Shkarupylo, Ravil Kudermetov, Tetiana Golub, Olga Polska, and Mariia Tiahunova. Towards model checking of the internet of things solutions interoperability. In *2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*, pages 465–468. IEEE, 2018.

[20] Alireza Souri and Monire Norouzi. A state-of-the-art survey on formal verification of the internet of things applications. *Journal of Service Science Research*, 11(1):47–67, 2019.

[21] Imen Tounsi, Abdessamad Saidi, Mohamed Hadj Kacem, and Ahmed Hadj Kacem. Internet of things design patterns modeling proven correct by construction: Application to aged care solution. *Future Generation Computer Systems*, 148:395–407, 2023.

[22] Jeannette M Wing. *What is a formal method?* Carnegie-Mellon University. Department of Computer Science, 1989.

[23] Jeannette M Wing. A specifier's introduction to formal methods. *Computer*, 23(9):8–22, 1990.

[24] Xin Xin, Sye Loong Keoh, Michele Sevegnani, and Martin Saerbeck. Dynamic probabilistic model checking for sensor validation in industry 4.0 applications. In *2020 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pages 43–50. IEEE, 2020.

[25] Lei Yu, Yang Lu, Benhong Zhang, Lei Shi, Fangliang Huang, Ya Li, and Yulian Shen. Hierarchical formal modeling of internet of things system oriented to user behavior. In *2020 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pages 282–289. IEEE, 2020.

Leveraging Artificial Intelligence for enhanced cybersecurity in FANETs

Mouzai Mustapha[1] **and Riahla Mohamed Amine** [1]

[1] ***Department of Computer Science, University of Boumerdes,*** *m.mouzai@univ-boumerdes.dz*

Abstract

In recent years, the deployment of unmanned aerial networks has experienced a significant growth, UAVs have shown a remarkable efficiency in their military missions which led to their expansion in other civilian fields. Unfortunately, such an advancement brings many consequences with it, many constraints and challenges inhibit UAV networks progression, most importantly the cyber-attacks. Recently cyber security researchers have shown interest in designing security mechanisms, intrusion detection and cyber defense systems conceived to fit these networks. This paper presents a literature review about some works leveraging Artificial Intelligence (AI) techniques to face the different cyber security attacks and vulnerabilities targeting UAVs.

Keywords: FANETs, UAV, Cyber-security, artificial intelligence, intrusion detection system .

1 Introduction

Unmanned aerial vehicles (UAVs), or drones, are aircraft that fly without a pilot onboard. They are either remotely operated through a ground control station (GCS) or autonomously navigate using pre-programmed flight paths.
In other hand, Flying Ad Hoc Networks(FANETs) frequently cited in literature under various terms such as UAV clusters, UAV groups, UAV swarms and Fleet of drones are a set of two or more UAV nodes, characterized by the decentralized architecture of Ad Hoc networks where UAV nodes acts as Routers and Hosts at the same time. They interconnect wirelessly through the FANET, creating a highly dynamic topology, where network routes are constantly changing.
Unmanned aerial vehicles come in various sizes, speeds, computational capabilities, and battery capacities, depending on their intended application, they are pervasive in various domains including military such as tactical operations [1] , Border surveillance, Search and rescue [2], additionally to civilian fields namely Delivery and Logistics[3], Agriculture[4] and Environmental Monitoring. However, their intrinsic vulnerabilities and susceptibility to security threats pose significant challenges to maintaining the integrity, confidentiality, and availability [5] of sensitive information transmitted within these networks.
In response to these challenges, there is a growing interest in harnessing the power of Artificial Intelligence techniques to bolster cybersecurity in FANETs. AI, encompassing machine learning, deep learning, and other cognitive computing approaches, offers a transformative approach to detecting, analyzing, and mitigating security threats within dynamic and decentralized network environments. The objective of this study is to offer a comprehensive overview of existing research efforts aimed at countering various cybersecurity threats targeting UAVs through the application of Artificial Intelligence (AI) techniques. Additionally, we indicated our proposed direction to contribute to the security enhancement of UAV networks through the utilization of AI-driven approaches. The remainder of this paper is organized as follows; in section 2 we reviewed the existing literature on the subject. then we discussed the challenges and the lacks still encountered in section 3. finally, we concluded this work by outlining our future directions in section 4.

2 Literature analysis

Various studies in the literature have extensively discussed the most challenging security threats related to UAVs and have proposed numerous detection and mitigation techniques leveraging artificial intelligence algorithms.

In [6], Authors proposed a lightweight module against Global Positioning System (GPS) Spoofing attack integrated within UAVs, their approach consists of modeling the detection process as a Bayesian Network with pearl's message-passing algorithm. Different variables namely the new received GPS Information, the current GPS Information and neighbor's GPS Information represent the main factors that affect the final decision whether a new received signal is threatening or not.

The designed model was trained using SatGrid: G22 and SatGRID: S7 Datasets and the performance metrics used for validation were Precision, Recall, and False Positive Rate (FPR). Authors claimed that their solution exhibits 96.2

A hierarchical detection and response system combining rules-based detection and anomaly detection techniques was proposed by [7], to enhance the security of UAV networks, this work is designed for opportunistic networks composed of a fleet of UAVs conducting exploration missions and assisted by a set of ground stations in case of disasters. The authors defined a series of rules and thresholds running at every node referred to as UDAs to detect various attacks, including GPS spoofing, jamming, false information dissemination, gray hole and black hole attacks. These thresholds are updated frequently using an Support Vector Machine (SVM) algorithm running on base stations.

In [8], the authors investigated the detection of one of the stealthiest jamming attacks in UAV assisted wireless communications which is reactive short period jamming attack (RSPJ), the advantage of their solution lies in its effectiveness and performance even though the absence of prior knowledge about signal and channel characteristics (DataSet),The solution design starts by processing the signal received as a sequence of logarithmic received energy, which will then be clustered as either jammed or unjammed signal using Hidden Markov Model (HMM).

Authors of [9], have provided a survey about cybersecurity attacks and defenses for unmanned aerial systems, The authors summarized a systematic approach of threat assessment models that concern Unmanned aerial systems. moreover, they partitioned the attacks against UAVs into four categories as follows, network communications attacks, software attacks, payload attacks and machine learning attacks.in further, they offered valuable references describing the most common attacks and vulnerabilities threatening UAVs at each category mentioned above, along with an examination of the potential damages they may inflict. On the other hand, they explored the existing security countermeasures tailored to these threats, conducting an in-depth analysis of their strengths and weaknesses.

A Q-learning two-layer cooperative intrusion detection system(Q-TCID) was introduced in [10], the authors designed an intelligent voting Q-learning algorithm running at host level, it interacts with the network environment and cooperates with other nodes. Q-TCID launches voting sessions every T interval of time using Q-learning techniques in order to enhance the voting results of malicious nodes detection. As a supplementary security layer, an additional intelligent auditing Q-learning algorithm is implemented at software level to assess host level voting results. This approach aims to achieve a high detection rate that prolongs the Mean Time to Failure (MTTF) of the IoD network as much as possible alongside with reduction of energy consumption, effectively the simulation results showed a higher accuracy rate and less energy consumption compared to other studies.

In [11], a security framework has been developed within a smart farming environment assisted by UAVs that gather data from sensors deployed in farms, and offload it at fog sites. The authors designed an hyperparameter optimization-based intrusion detection system, where multiple machine learning and clustering models were stacked across many tiers of their solution. The purpose of this approach is to increase the intrusion detection accuracy while simultaneously reducing the classification errors for known and zero-days attacks. effectively, this framework was evaluated in terms of accuracy, precision, recall and other metrics, and it outperformed other existing solutions, furthermore it showed optimization in energy consumption and computational overhead.

In [12], Authors designed an Intrusion detection system based on open-set recognition and an active learning approach within a centralized UAV network. They trained their Intrusion Detection System (IDS) on CTU and CICIDS2017 datasets to detect and classify known attacks, while on the other hand, they selected the most informative observations of unknown attacks to create novel attacks classes, hence they are used to retrain the IDS for future detections. This solution is mainly based on Convolutional Neural Network (CNN) algorithm, it was designed to classify (N+1) different attack models, where N is the number of known attacks and 1 represents a single unknown attack class. Subsequently, the OpenMax layer is used instead of SoftMax layer because it offers more calibrated and informative predictions, particularly when dealing with unknown attack samples. The results showed that the IDS proposed has a better accuracy and F-measures values of unknown attacks detection compared to other methods.

In [13], the authors designed a secure system dedicated to IoD networks. It consists of an intrusion detection system based on blockchain and radial basis function neural networks called BIIR. The proposed system obliges every drone to register in a Safety Authority (SA) by following Zero Knowldge Proof (ZKP) process in order to access the network, consequently the SA grants new identification to drones successfully registered. Furthermore, Drone-to-Drone (D2D) and Drone-to-Everything (D2X) communications are validated using blockchain, the latter guarantee the privacy and the immutability of data exchanged between drones and keeps illegal nodes out of the network. Subsequently, an IDS based on reinforcement learning method and Radial vasis function neural network (RBFNN) as an agent was developed, while pre-trained data was imported using Transfer learning to reduce network training time. Consequently, the results demonstrated that this

Study	Threat	Algorithm	Dataset	Evaluation Metrics
[6]	GPS Spoofing attack	Bayesian Network	SatGrid : G22 & G7	Precision, Recall, FPR,Accuracy
[7]	jamming false information-dissemination gray hole black hole	Support Vector Machine	No Dataset required	Detection rate FPR Efficiency Communications overhead
[8]	reactive short period jamming attack	Hidden Markov Model	No Dataset required	Detection probability False alarm probability
[10]	Nodes Misbehavior	Q-learning	No Dataset required	mean time to failure energy consumption Accuracy
[11]	Denial of service Spoofing attacks Jamming attacks	XGBoost random forest decision tree extra tree k -means	Local data	accuracy precision recall and other metrics
[12]	Multiple Known and Unknown attacks	convolutional neural network	CTU CICIDS2017	accuracy F-measures
[13]	DDOS BOT Port Scan	reinforcement learning Transfer learning	NSL-KDD UNSW-NB15 AWD CICIDS2017 CICDOS2019	Accuracy Precision recall F1

Table 1: Summary of reviewed papers

solution outperformed other methods in terms of various metrics when tested with different datasets.

3 Conclusion and future direction

In conclusion, this paper has explored the various security threats faced by UAV networks. We have reviewed existing literature on UAV security challenges and discussed the importance of developing robust detection and mitigation techniques to counter such threats. Moving forward, our future research direction will focus on addressing the specific challenges of GPS spoofing attacks in UAV networks. We propose to develop a novel detection system leveraging machine learning. Additionally, we aim to design effective mitigation strategies to prevent or mitigate the impact of GPS spoofing attacks on UAV operations.

References

[1] Atif Ali, Yasir Khan Jadoon, Sabir Ali Changazi, and Muhammad Qasim. Military operations: Wireless sensor networks based applications to reinforce future battlefield command system. In *2020 IEEE 23rd International Multitopic Conference (INMIC)*, pages 1–6. IEEE, 2020.

[2] Sunghun Jung and Hyunsu Kim. Analysis of amazon prime air uav delivery service. *Journal of Knowledge Information Technology and Systems*, 12(2):253–266, 2017.

[3] Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, Thomas Lagkas, and Ioannis Moscholios. A compilation of uav applications for precision agriculture. *Computer Networks*, 172:107148, 2020.

[4] Sonia Waharte and Niki Trigoni. Supporting search and rescue operations with uavs. In *2010 international conference on emerging security technologies*, pages 142–147. IEEE, 2010.

[5] Hassan Jalil Hadi, Yue Cao, Khaleeq Un Nisa, Abdul Majid Jamil, and Qiang Ni. A comprehensive survey on security, privacy issues and emerging defence technologies for uavs. *Journal of Network and Computer Applications*, 213:103607, 2023.

[6] Chafiq Titouna and Farid Naït-Abdesselam. A lightweight security technique for unmanned aerial vehicles against gps spoofing attack. In *2021 International Wireless Communications and Mobile Computing (IWCMC)*, pages 819–824. IEEE, 2021.

[7] Hichem Sedjelmaci, Sidi Mohammed Senouci, and Nirwan Ansari. A hierarchical detection and response system to enhance security against lethal cyber-attacks in uav networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(9):1594–1606, 2017.

[8] Chen Zhang, Leyi Zhang, Tianqi Mao, Zhenyu Xiao, Zhu Han, and Xiang-Gen Xia. Detection of stealthy jamming for uav-assisted wireless communications: An hmm-based method. *IEEE Transactions on Cognitive Communications and Networking*, 9(3):779–793, 2023.

[9] Zhaoxuan Wang, Yang Li, Shihao Wu, Yuan Zhou, Libin Yang, Yuan Xu, Tianwei Zhang, and Quan Pan. A survey on cybersecurity attacks and defenses for unmanned aerial systems. *Journal of Systems Architecture*, 138:102870, 2023.

[10] Moran Wu, Zhiliang Zhu, Yunzhi Xia, Zhengbing Yan, Xiangou Zhu, and Nan Ye. A q-learning-based two-layer cooperative intrusion detection for internet of drones system. *Drones*, 7(8):502, 2023.

[11] Junaid Sajid, Kadhim Hayawi, Asad Waqar Malik, Zahid Anwar, and Zouheir Trabelsi. A fog computing framework for intrusion detection of energy-based attacks on uav-assisted smart farming. *Applied Sciences*, 13(6):3857, 2023.

[12] Zhao Zhang, Yong Zhang, Jie Niu, and Da Guo. Unknown network attack detection based on open-set recognition and active learning in drone network. *Transactions on Emerging Telecommunications Technologies*, 33(10):e4212, 2022.

[13] Arash Heidari, Nima Jafari Navimipour, and Mehmet Unal. A secure intrusion detection platform using blockchain and radial basis function neural networks for internet of drones. *IEEE Internet of Things Journal*, 10(10):8445–8454, 2023.

A Hybrid Transformer-SVM Model for Intrusion Detection in IoT Networks using NSL-KDD and CICIDS2018

Leila Boutaleb[1], Anis Cheklat[2], and Tayeb Benzenati[3]

[1] *University of Boumerdes, LIMOSE LABORATORY , Boumerdes, Algeria.*
boutalebleila99@gmail.com
[2] *University of Boumerdes, LIMOSE LABORATORY, Boumerdes, Algeria.*
anisscheklat6@gmail.com
[3] *University of Boumerdes, LIMOSE LABORATORY, Boumerdes Algeria, and Digital Research Centre of Sfax, SM@RTS laboratory, Sfax, Tunisia.*tayeb.benzenati@gmail.com

Abstract

The Internet of Things (IoT) is rapidly expanding, connecting a variety of smart devices. However, this also exposes IoT systems to potential threats and attacks. To ensure the security of IoT networks, artificial intelligence (AI) plays a crucial role, including machine learning and deep learning. These techniques enable the detection of attacks by analyzing large amounts of data generated by IoT devices, thereby identifying abnormal patterns. Specific datasets such as NSL-KDD and CICIDS2018 are used to train and test the models. In the last few years, the Transformer, a model based on attention and self-learning mechanisms, shows a remarkable performance in several fields. In this work, we propose a novel approach that integrates the Transformer to extract abundant features, and the well-established machine learning classifier Support Vector Machine (SVM) to develop an attack detection system. The experimental results, based on evaluations conducted on two distinct datasets, show that the proposed method effectively increases the accuracy of attack detection in comparison to classic machine learning algorithms.

Keywords: Machine Learning, SVM ,NSL-KDD, CICIDS2018, self-attention, intrusion detection.

1 Introduction

The proliferation of interconnected devices in Internet of Things (IoT) networks has led to the formation of Smart Distributed IoT (SDIoT) networks, presenting both opportunities and security challenges. The complexity of SDIoT networks, characterized by diverse devices and dynamic connections, creates a fertile ground for malicious attacks. Detecting these attacks in real-time is crucial for maintaining network integrity.

To address this challenge, researchers are exploring advanced techniques such as transformer-based models and Support Vector Machine (SVM) algorithms. Our research aims to optimize attack detection in SDIoT networks by combining the strengths of transformer models, which capture complex dependencies in network traffic, with the precision of SVM in establishing decision boundaries between attack classes.

In this article, we detail our proposed approach, highlighting implementation steps and anticipated benefits. We also present experimental outcomes and comparisons with existing detection methods. By enhancing SDIoT network security, we contribute to the development of more effective cybersecurity solutions.

2 Related Works

In this study, several machine learning algorithms were analysed for intrusion analysis, including Support Vector Machine (SVM), k-nearest neighbors (KNN), Linear Regression (LR), Naive Bayes (NB), Random Forest (RF), and Decision Tree. **SVM** is suitable for both linear and non-linear data and utilizes a radial basis kernel function to handle non-linear data. **KNN** is a supervised classifier that uses Euclidean distance and a user-defined K value to classify data into different classes. **LR** calculates the probability of a target variable based on predictive analysis and describes the relationship between dependent and

independent variables. **NB**, specifically Gaussian NB, was chosen for its performance with numeric data, classifying records independently based on Bayes' theorem. **DT** creates a tree-like structure to solve classification problems, while **RF** considers the output of multiple decision trees to classify data, selecting significant attributes from a random set for improved classification. These machine learning techniques were applied to classify data in the study, enabling the accurate detection of network intrusions. In our research, we conducted experimental analysis to compare machine learning techniques on two datasets. Table 1 and 2 depict the performance metrics (Accuracy, Precision and Recall) for each machine learning algorithm applied to the respective datasets.

Table 1: Result on CICIDS2018 dataset

Method	Accuracy	Precision	Recall
LR	0.99	0.99	1.0
SVM	0.97	0.85	0.99
DT	0.94	0.88	1.0
RF	0.95	1.0	0.94
NB	0.99	0.83	0.94
KNN	0.99	0.98	0.93

Table 2: Result on NSL-KDD dataset

Method	Accuracy	Precision	Recall
LR	0.99	0.96	0.96
SVM	0.76	0.67	0.88
DT	0.85	0.95	0.70
RF	0.97	0.88	0.73
NB	0.70	0.64	0.99
KNN	0.92	0.87	0.86

3 Background

3.1 Transformers

The Transformer model, introduced by Vaswani et al. in 2017, revolutionized natural language processing and artificial intelligence with its attention-based architecture. Unlike previous models relying on RNNs or convolutions, the Transformer efficiently captures long-term dependencies and relationships in sequences, thanks to attention mechanisms. By focusing on relevant elements, it achieves superior performance in tasks like machine translation. Widely adopted across various domains, the Transformer has set new benchmarks in AI, showcasing its prowess in sequence modeling and driving advancements in the field [6].

The core idea of the Transformer is to preserve the interdependence of words (tokens) in a sequence by utilizing the attention mechanism at the heart of its architecture. This concept of attention measures the relationship between two elements from two sequences. In the context of Natural Language Processing (NLP), the attention mechanism allows information to be conveyed to the model so that it focuses its attention on the right words in sequence A when processing a word from sequence B. Self-attention is the same attention mechanism, but applied to a single sequence. In the Transformer model, three essential components facilitate attention mechanisms:

- **Key:** Represents a transformed version of an input element, encoding relevant information used to calculate attention weights.
- **Value:** Also a transformed version of the input element, containing additional information aiding in weighting during the attention process.
- **Query:** Represents the current context to be encoded, crucial for determining the significance of each element in the current context by calculating attention weights.

These key, value, and query vectors are fundamental to the Transformer's attention mechanism, enabling it to capture relationships and dependencies between elements in sequences, making it a powerful tool in natural language processing and various other tasks.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where Q indicates the Query vector, K represents the Key vector, V denotes the Value vector, and d_k is the dimension of key vector.

The calculation of the attention coefficient is performed using the formula above, where the three vectors are multiplied by the embedding of the input sequence. To simplify, in a search engine context, Q would represent the search query, K could be seen as the features (text, images, etc.) associated with the most relevant results, and V can be viewed as those results. The attention weights are divided by $\sqrt{d_k}$. to stabilize gradients during training, and then passed through a softmax function to normalize the weights, effectively selecting the values to retain [6].

3.2 The SVM classifier

The SVM, also known as Support Vector Machine, is one of the most commonly used algorithms in supervised learning, primarily for classification. Its primary goal is to find the best hyperplane to separate data into multiple classes, thereby facilitating the classification of new data points. The extreme points play a crucial role in constructing this hyperplane and are referred to as support vectors.

In theory, SVMs are more effective when the number of features is relatively low compared to the dataset's size. They work well with small to medium-sized datasets and can handle complex feature spaces.

In our study, we combine the advantages of transformers, which capture short and long-term data interdependencies, with the promising features of SVMs. This combination is expected to enhance performance, since the quality of the feature vector is crucial for a good classifier. Transformers provide a new representation rich in underlying features that reflect problem-solving logic and hidden data structures.

4 Datasets

4.1 NSL-KDD dataset

The statistical analysis showed that there are important issues in the data set which highly affects the performance of the systems, and results in a very poor estimation of anomaly detection approaches. To solve these issues, a new data set as, NSL-KDD [4] is proposed, which consists of selected records of the complete KDD data set. The advantage of NSL KDD dataset are :

1. No redundant records in the train set, so the classifier will not produce any biased result
2. No duplicate record in the test set, which have better reduction rates.
3. The number of selected records from each difficult level group is inversely proportional to the percentage of records in the original KDD data set.

The training dataset consists of 21 different attack types out of the 37 present in the test dataset. The attack types in the training dataset are considered "known" attacks, while the attack types in the test dataset that are not present in the training dataset are referred to as "novel" attacks. These novel attacks represent new, previously unseen attack types during training.

The attack types are grouped into four categories:

- DoS (Denial of Service) (Land, Neptune, Pod, Smurf, Teardrop, Apache2, Udpstorm, Processtable, Worm)
- Probe (Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint)
- U2R (User to Root) (Buffer Overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps)
- R2L (Remote to Local) (Guess_Password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Warezclient, Spy, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named)

These categories represent the major attack types provided by the authors in both the training and testing datasets.

4.2 CICIDS2018 dataset

The CICIDS2018 dataset provides a wide range of usable attack profiles in the field of smart security applied to network topologies and protocols in a generic approach. This dataset has been improved while considering the standards of CSE-CIC IDS2017. CSE-CIC IDS2018 is a currently utilized public dataset that comprises 2 categorized profiles and consists of 5 different attack methods. Various data scenarios were collected, and raw data was edited on a daily basis. 80 statistical properties such as packet length, packet count, byte count, etc., were separately calculated for both the forward and backward directions during data creation.

Ultimately, the dataset has been made publicly available on the internet for all researchers. The dataset is provided in two formats: CSV and PCAP, containing approximately 5 million records. The CSV format is primarily used in the field of artificial intelligence, while the PCAP format is used for extracting new features [3].

The attack types are grouped into categories: Force brute (SSH-Bruteforce, FTP-BruteForce Robot Réseau de zombies) Attaque DOS (DoS - Hulk, DoS - Test HTTP lent, DoS - Slowloris, DoS - GoldenEye) Attaque DDoS (DDOS - HOIC, DDOS - LOIC-UDP, DDOS - LOIC-HTTP) Attaque Web (Force Brute - XSS, Force Brute –Web, Injection SQL).

4.3 Preprocessing

Preprocessing is an essential step in which symbolic or non-numeric attributes are replaced or removed. To reduce the computational intricacy of the model and simultaneously boost its accuracy, in the current work, four feature subsets presented in Table I were taken into consideration. These feature subsets included one-hot encoding for categorical attributes, ensuring that only relevant attributes were used to train/test the model. This approach not only significantly reduced training time but also enhanced the model's performance. Thus, preprocessing of data is imperative for dimensional reduction, thereby minimizing the overhead of processing the data.

5 Proposed method

This research aims to develop a precise classification model for network traffic instances as normal or intrusive by combining a transformer-based neural network with an SVM classifier. The NSL-KDD and CSE-CIC-IDS2018 datasets, consisting of N traffic instances, are used, with each instance represented as a d-dimensional attribute vector, denoted as X. The Transformer model, formulated as a function f(X), extracts features using multi-head attention mechanisms, normalization, and dense layers. The extracted features, denoted as h, are then used as input for the SVM classifier, which predicts the class of each traffic instance. Performance is evaluated through k-fold cross-validation, calculating metrics such as accuracy, loss, and confusion matrices.

5.1 Architecture

This description outlines the architecture of a classification model that relies on a Transformer and a Support Vector Machine (SVM) classifier.

Input Data: Input data represents the information or features on which the model will perform classification. They are typically structured as tensors or matrices, depending on the type of data being used.

Transformer: The Transformer is the central component of the model's architecture. It consists of multiple stacked Transformer layers. Each Transformer layer uses multi-head attention mechanisms to capture relationships and dependencies among different parts of the input data. These mechanisms enable the model to learn rich and hierarchical representations of the data, taking into account complex interactions among elements.

Transformer Layer Components: Each Transformer layer includes the following elements:

1. **Multi-Head Attention**: Allows the model to make comparisons and attention calculations among different parts of the input data to focus on the most important parts.

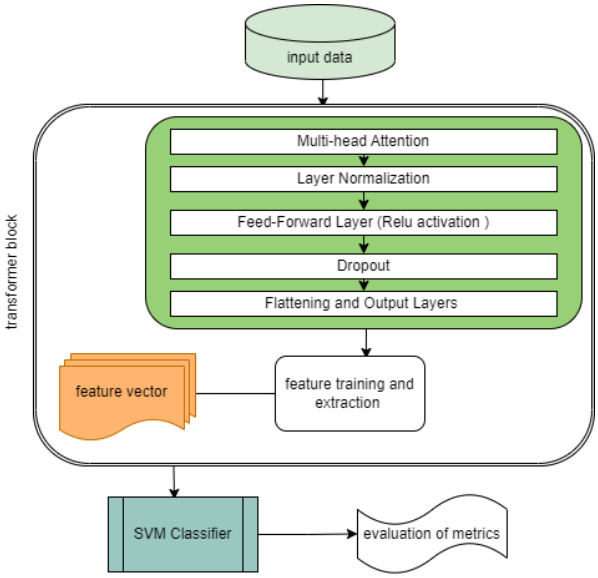


Figure 1: Overall framework of the proposed method.

2. **Layer Normalization**: This layer normalizes the outputs of multi-head attention to improve training stability. Dense Layer: Applied after layer normalization to introduce non-linearity and perform more complex transformations of data representations.

3. **Dense Layer** Applied after layer normalization to introduce non-linearity and perform more complex transformations of data representations. The Dense Layer often includes the Rectified Linear Unit (ReLU) activation function, which adds non-linearity to the transformation.

4. **Dropout**: Used as a regularization technique to reduce overfitting by randomly deactivating some neurons in the dense layer during training. By stacking multiple Transformer layers, the model can learn increasingly abstract and complex representations of the input data, leveraging relationships at different scales.

5. **Output Layer**

   The Output Layer of the Transformer takes the representations learned by the preceding layers and produces the final output of the model. Depending on the task, this layer may include different activation functions (e.g., softmax for classification).

Feature Vectors: After the Transformer layers, feature vectors are extracted using a flattening layer. This operation transforms the output tensors of the Transformer layers into a 2D representation where each input example is represented by a feature vector.

SVM Classifier: The extracted feature vectors are used as input for an SVM classifier (Support Vector Machine). This classifier learns to separate different classes optimally in the feature space by constructing a hyperplane that maximizes the margin between examples from different classes. When a new example is presented, the SVM classifier projects it into the feature space and classifies it based on its position relative to the learned hyperplane.

By combining the powerful feature extraction capabilities of the Transformer with the precise classification abilities of the SVM, this architecture enables the model to learn to represent data in an expressive manner and achieve accurate classification based on the extracted features.

6 Experimental results

Our hybrid approach provides an innovative solution to enhance the security of IoT systems by accurately and reliably detecting potential attacks. The results obtained on NSL-KDD and CICIDS2018 are shown in table 3 and 4, respectively.

The results of this comparative study underscore the exceptional performance of our Transformer-SVM model in detecting attacks on the NSL-KDD and CICIDS2018 datasets. This accomplishment

Table 3: Obtained scores on NSL-KDD

Classifier	Accuracy
Our	0.97
RF [4]	0.71
SVM [4]	0.93
LR [1]	0.86

Table 4: Obtained scores on CICIDS2018

Classifier	Accurarcy
Our	0.98
RF [4]	0.94
SVM [2]	0.89
LR [5]	0.94

highlights the Transformer's capability to capture intricate relationships within heterogeneous IoT network data, while the SVM classifier enhances reliability in attack detection. These findings advocate for further exploration of this methodology in other domains of cybersecurity and pave the way for future research aimed at bolstering defenses against attacks in IoT networks. In conclusion, our study makes a significant contribution to advancing IoT system security and presents promising avenues for more effective and resilient solutions in the realm of cybersecurity.

7 Conclusion

In this work, we proposed a novel technique that leverages the capabilities of both transformer models and SVM. The proposed model showed its ability to effectively capture complex relationships within heterogeneous IoT network data, outperforming traditional models through attention and self-learning mechanisms. This resulted in improved detection accuracy and a notable reduction in false positives and false negatives. Compared to previous approaches, the Transformer excelled in identifying intricate abnormal patterns, enhancing precision in attack detection. The integration with the SVM classifier further boost the detection capabilities, leveraging the effective classification capabilities thanks to SVM. In our future works, we intend to evaluate the proposed method in practice on some real life tasks, with a focus on reducing the complexity of the model.

References

[1] Iram Abrar, Zahrah Ayub, Faheem Masoodi, and Alwi Bamhdi. A machine learning approach for intrusion detection system on nsl-kdd dataset. In *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, pages 919–924, 2020.

[2] Aqib Ali, Samreen Naeem, Sania Anam, and MM Ahmed. Machine learning for intrusion detection in cyber security: Applications, challenges, and recommendations. *UMT Artif. Intell. Rev*, 2(2):41–64, 2022.

[3] Subiksha Srinivasa Gopalan, Dharshini Ravikumar, Dino Linekar, Ali Raza, and Maheen Hasib. Balancing approaches towards ml for ids: a survey for the cse-cic ids dataset. In *2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, pages 1–6. IEEE, 2021.

[4] Anish Halimaa and K Sundarakantham. Machine learning based intrusion detection system. In *2019 3rd International conference on trends in electronics and informatics (ICOEI)*, pages 916–920. IEEE, 2019.

[5] Anish Halimaa and K Sundarakantham. Machine learning based intrusion detection system. In *2019 3rd International conference on trends in electronics and informatics (ICOEI)*, pages 916–920. IEEE, 2019.

[6] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Enhancing visible light communication systems through artificial intelligence approaches

Abdelbaki Benayad[1], Amel Boustil[2], and Yassine Meraihi[3]

[1]*LIMOSE Laboratory, University of M'Hamed Bougara, Boumerdes,* a.benayad@univ-boumerdes.dz
[2]*LIMOSE Laboratory, University of M'Hamed Bougara, Boumerdes,* a.boustil@univ-boumerdes.dz
[3]*LIST Laboratory, University of M'Hamed Bougara, Boumerdes,* y.meraihi@univ-boumerdes.dz

Abstract

Visible Light Communication (VLC) emerges as a promising frontier in contemporary wireless communication. VLC boasts notable advantages, including abundant spectrum resources, immunity to electromagnetic interference, and robust security measures. However, VLC systems grapple with challenges inherent across various system components. The potential of Artificial Intelligence (AI) in mitigating nonlinear effects has garnered attention for its inherent adaptability to diverse transfer functions, thereby potentially catalyzing advancements in VLC research. This paper delineates the fundamental components of VLC systems, delves into AI applications within VLC, examines recent research on optimization methods, Machine Learning (ML), and Deep Learning (DL) applications, and discusses the challenges and prospects associated with this rapidly growing wireless transmission technique.

Keywords: Visible Light Communication, Artificial Intelligence, Machine Learning, Deep Learning, Optimization Methods.

1 Introduction

In recent years, interest in Visible Light Communication (VLC) technology has grown significantly. The reason behind this growing fascination is the large license-free bandwidth it offers, all within the unregulated optical spectrum. VLC emerges as a technology that can be seen as an alternative or complementary to traditional radio communications [17]. This technology finds application in various environment scenarios such as localization [6, 12], home networking systems [28], and communication in places where radio frequency (RF) radiation is prohibited such as hospitals and aircraft cabins [27]. Built upon Light-Emitting Diodes (LEDs), known for their sustainability and energy efficiency [1], this technology offers dual functionality. LEDs serve not only as illuminators but also as transmitters, enabling data transmission while providing illumination.

Artificial intelligence methods have demonstrated successful applications in prediction, classification, and optimization problems [5], among others. Firstly, these methods offer useful algorithms to address non-linearity issues in optical communication, including parameter estimation from noise, determination of complex mapping relationships between input and output signals, inference of probability distributions of received signals, and estimation of output values based on input samples [5, 29]. Secondly, machine learning (ML) and deep learning (DL) algorithms can be employed to monitor communication performance. For instance, neural networks and the K-means algorithm can aid in estimating various channel impairments and efficiently managing optical networks [13]. Furthermore, ML algorithms such as support vector machine (SVM) and K-means can accurately identify modulation formats and bit rates [23]. ML and DL algorithms have the capability to model the relationship between inputs and outputs using samples and labels, without requiring a detailed analysis of the complex relationships between individual features. Therefore, they represent promising tools for enhancing transmission performance in VLC systems.

Moreover, optimization methods tackle NP-Hard problems linked with Visible Light Communication challenges, like optimizing LED placement to maximize coverage and ensure uniform illumination across the intended area. By defining coverage objectives and constraints, optimization algorithms can ascertain the optimal placement and orientation of LEDs to attain the desired coverage levels [4, 3].

In this paper, we first explain the principles and mechanisms of VLC systems in Section II. Section III provides an overview of the challenges and applications of VLC. In Section IV, we explore the potential of AI methods in VLC. Finally, we conclude with our insights on the topic.

2 THE PRINCIPLE AND THE MECHANISM OF VLC SYSTEM

The VLC system is comprised of two distinct components, each serving a specific function. These components include the VLC transmitter and the VLC receiver [23].

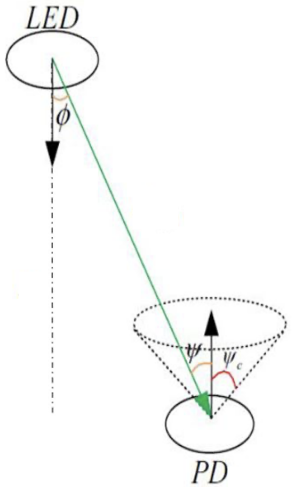


Figure 1: Transmitter(LED) and Receiver(photo-diodes) of VLC System

- **The VLC transmitter:** The digital signal processing procedures encompass various stages, comprising encoding binary data for transmission, modulating signals, applying pre-equalization when necessary, up-converting signals, and ultimately producing the digital signal ready for transmission to the transmitter circuit. Typically, VLC transmitters utilize Light Emitting Diodes (LEDs) as the light source. LEDs are favored for VLC applications owing to their rapid switching speed, efficiency, and suitability for digital modulation methods.
- **The VLC receiver:**: Receivers play a pivotal role in capturing light and converting it into electrical signals. Typically, VLC systems employ photodiodes as receivers [26]. However, photodiodes exhibit high sensitivity and can detect light waves outside the visible spectrum, such as ultraviolet and infrared. They are also prone to saturation, especially in external environments exposed to sunlight, leading to potential data reception issues due to interference. To address these challenges, alternative components can be utilized for light capture. For instance, smartphone cameras can serve as receivers, enabling any mobile device to receive data transmitted by a VLC transmitter. Moreover, LEDs themselves possess photo-sensing properties and can be employed as receivers.

3 Applications and Challenges of VLC

VLC offers a broad spectrum of applications, spanning from fast Internet connectivity via LED bulbs to communication across planets. These applications cover a wide array of scenarios, introducing novel dimensions to what is conventionally understood as ubiquitous and pervasive computing. In the subsequent section, we explore the potentials of Visible Light Communication, emphasizing its applications in indoor, outdoor, and underwater environments, each presenting distinct features and obstacles [23].

3.1 Indoor

Indoor VLC integration into the Internet of Things (IoT) ecosystem is on the rise. Incorporating VLC transmitters and receivers into IoT devices like sensors, smart appliances, and wearable enables seamless data communication and control, fostering interconnected indoor environments. Notably, Indoor VLC offers inherent security advantages due to the limited reach of light propagation. In contrast to radio frequency signals that can breach walls and be illicitly intercepted, VLC signals typically stay confined within room boundaries, enhancing data transmission privacy and security.[16].

3.1.1 Applications

VLC serves as a communication conduit for IoT devices, linking sensors, actuators, and smart appliances across smart homes, buildings, and industrial settings. In healthcare, VLC can bolster various applications including patient monitoring, asset tracking, and inter-device communication within medical equipment. Similarly, in educational settings, VLC fosters interactive learning by enabling instantaneous communication among students, educators, and digital learning materials.

3.2 Outdoor

Outdoor VLC applications encompass the utilization of street lamps for data transmission and vehicle-to-vehicle (VTV) communication. When employing street lamps for data broadcasting, similar principles to indoor VLC channel modeling apply, though the impact of reflected signal components is diminished in outdoor environments due to greater distances. In VTV VLC applications, vehicle lights (brakes and headlights) or traffic lights serve as data transmitters, with high-speed cameras employed as receivers [14]. Nonetheless, there currently exists no theoretical channel model for vehicular VLC applications, considering factors such as vehicle speed and atmospheric conditions like rain, snow, and fog. Moreover, outdoor applications are susceptible to significant shot noise during daylight hours. Consequently, the reliability of such links under varying atmospheric conditions and times of day remains an important area for ongoing research.

3.2.1 Applications

Outdoor VLC is integral to the advancement of smart city initiatives, facilitating data communication, sensor connectivity, and intelligent infrastructure management. Its applications span a range of areas, encompassing smart lighting systems, environmental monitoring, public safety, traffic management, and smart parking solutions. VLC enables seamless communication between vehicles and various infrastructure elements like traffic lights and road signs, thereby bolstering road safety, optimizing traffic flow, and bolstering emerging applications in connected and autonomous vehicles [23].

3.3 Underwater

Underwater communications find utility across various domains, including oil and gas exploration, remotely operated vehicles (ROVs), and facilitating communication among divers. However, water presents distinct challenges compared to traditional wired and wireless communications in the atmosphere. For instance, radio frequency signals experience rapid fading underwater and are largely unsuitable for most underwater applications. Therefore, underwater communication necessitates sophisticated devices to achieve relatively low transmission rates, even over short distances [25].

3.4 Challenges

In this section, we outline and deliberate upon the primary challenges encountered in VLC communications, spanning from light-related concerns like flickering, dimming, line of sight limitations, and interference, to broader wireless communication challenges that must be tackled to ensure optimal performance, such as up-link and mobility issues. Additionally, we examine existing strategies proposed to surmount these challenges and highlight unresolved matters.

3.4.1 Line of Sight (LOS)

In indoor VLC systems, it's typically assumed that the user maintains a direct line of sight with the light source. However, in numerous indoor settings, illumination is deliberately directed through reflection or refraction, often obscured by lamp shades or covers and oriented toward walls or other surfaces. These lighting configurations, which hinder direct visibility of the light source from indoor positions, can significantly influence communication quality and user experience [23].

3.4.2 Flickering

Flickering poses a significant obstacle to VLC, characterized by fluctuations in light brightness discernible to humans. This issue is commonly tackled in implementations of indoor VLC systems, like those found in offices or supermarkets. Depending on the modulation method of light waves, oscillations may become

visible to the human eye, potentially leading to discomfort and health concerns. Consequently, there's a necessity to modulate the waves to ensure their frequencies surpass the threshold at which they are perceptible to humans [31].

3.4.3 Noise and interference

In WiFi networks, devices transmitting on the same frequency can disrupt each other. When light serves as the communication medium, natural light becomes an interference source, particularly affecting VLC, especially outdoors. Alongside natural light, artificial lights can also disrupt communication, potentially overwhelming the receiver. Another interference factor at the receiver is the multi-path issue. Unlike wired communication, where signal propagation is mostly confined to the wire, in Visible Light Communication, the signal can propagate through the environment toward the LED lamps, involving refraction and reflection.

3.4.4 Up-link

A practical VLC communication system should support both up-link and down-link transmissions. LED light bulbs can fulfill both roles, functioning as VLC transmitters and light sources. For reception, a basic photo-diode can capture modulated light, which is then decoded. Hence, down-link transmission, from LED lights to devices, is straightforward. However, transmitting data from devices to LED light bulbs poses greater challenges.

3.4.5 Dimming

In a VLC system employing LED lamps, the power of the communication signal correlates directly with the intensity of light emitted. Consequently, theoretically, lower light intensities result in reduced communication range and data transmission rates. Dimming, which adjusts the perceived brightness of the light source based on user preferences, plays a crucial role in many settings. Dimming is often a vital feature, offering advantages such as enhanced comfort and energy efficiency by creating pleasant environments [30].

3.4.6 Mobility

To ensure the widespread adoption of Visible Light Communication as a technology, mechanisms are necessary to guarantee uninterrupted, high-speed connections within the system's coverage area. This entails the receiver's ability to detect light signals from the transmitter across any location within a room, necessitating a wider emission angle at the transmitter and a broader Field Of View (FOV) at the receiver. However, these adjustments may lead to increased interference from refracted waves. Notably, Visible Light Communication diverges significantly from radio-frequency systems in terms of signal propagation, relying heavily on Line of Sight (LOS) transmission and the relative orientation of the receiver to the transmitter. Consequently, the Signal-to-Noise Ratio (SNR) of the light can vary considerably as the receiver moves, even within the coverage area of the light. [7].

4 Prospects of AI methods in VLC

In this section, we will demonstrate how Artificial Intelligence can play a significant role in tackling the challenges encountered in Visible Light Communication technology.

4.1 Machine Learning and Deep Learning in VLC

Machine Learning (ML) and deep learning (DL) constitute an interdisciplinary domain integrating statistics, probability, optimization theory, and algorithm complexity theory. The application of ML and DL in VLC can be classified into the following four categories.

4.1.1 Nonlinear mitigation

In recent years, artificial neural networks (ANNs) have gained widespread adoption in VLC systems to alleviate nonlinear signal distortion. By utilizing a segment of the transmitted signal as a reference, ANNs can effectively grasp the system's characteristics through their robust nonlinear mapping capabilities,

thus mitigating nonlinear signal distortion. Studies referenced from [10, 8, 18] underscore the efficacy of deep learning methodologies, such as deep neural networks (DNNs), in faster-than-Nyquist VLC systems, Gaussian kernel-based DNNs (GK-DNN), and long short-term memory (LSTM) networks in phase-amplitude modulation (PAM) based VLC systems. These findings underscore the superiority of deep learning techniques in nonlinear compensation.

4.1.2 Jitter compensation

In VLC systems, signal distortion can arise from system jitter, potentially leading to misinterpretation of the signal when traditional decision strategies are employed. Authors in references [19, 22, 35] utilize the 2-dimensional density-based spatial clustering of applications with noise (2D-DBSCAN) and 3D-DBSCAN algorithms to blindly equalize PAM and quadrature amplitude modulation (QAM) signals. This method adeptly mitigates false decisions stemming from signal jitter, thereby substantially enhancing system performance.

4.1.3 Modulation format identification

Non-linearity within VLC systems results in a mismatch of constellation points, which can lead to misinterpretation of the received signal. In references [20, 21], authors introduce the Cluster Algorithm of Perception Decisions (CAPD), employing K-means clustering to enhance VLC system performance. Likewise, in reference [33], the author employs the Gaussian Mixture Model (GMM) to reconstruct the decision boundary for signals, effectively reducing misinterpretations arising from constellation mismatch.

4.1.4 Phase estimation

Within VLC systems, non-linearity induces phase deviation in the received signal. Employing ML algorithms such as SVM, K-means [32], and GMM [34], enables effective compensation for the nonlinear degradation induced by phase deviation in VLC systems.

4.2 Optimization methods in VLC

Optimization methods are vital for improving the performance, reliability, and efficiency of VLC systems. We introduce some examples where optimization methods are commonly applied in VLC.

4.2.1 Power Allocation

Power allocation in VLC systems involves determining the best distribution of power among various light sources or channels within the system. This allocation is vital for optimizing the overall performance, reliability, and efficiency of the VLC system while adhering to specific requirements and constraints. In this context, the authors in [2] introduced meta-heuristic optimization algorithms to compute the power allocation coefficients for users.

4.2.2 Signal to noise ratio (SNR)

In VLC systems, the signal-to-noise ratio (SNR) signifies the relationship between the power of the transmitted signal and the ambient noise within the communication channel. This metric serves as a fundamental gauge for evaluating communication quality in VLC systems. A higher SNR denotes a more robust signal compared to the noise. In this context, the authors in [24] introduced a hybrid optimization algorithm designed to maximize both the SNR and received signal strength (RSS).

4.2.3 Connectivity and quality

Optimization techniques can enhance node placement algorithms to strategically deploy VLC transmitters, receivers, and relays, thereby maximizing coverage, connectivity, throughput, and communication quality in VLC networks. In this context, researchers in [9] utilized PSO (Particle Swarm Optimization), VLP-IACS (Cuckoo Search Algorithm) [11] and WOA (Whale Optimization Algorithm) [15] to address the connectivity and quality challenges, taking into account the average outage area rate and SNR.

Finally, we proposed an indoor VLC system model, simulating an empty conference room with dimensions of 10 m $\times$ 10 m $\times$ 3 m as shown in Fig 2. The room is equipped with multiple LEDs (L) which act as access points, and multiple receiving users (U) that are randomly distributed within the room.

Additionally, we introduced a novel algorithm called ECHIO (Enhanced Chaos-based Herd Immunity Optimizer), which incorporates the concept of chaotic maps and the Opposition-Based Learning (OBL) mechanism. This algorithm aims to tackle connectivity and quality challenges, while considering metrics such as throughput and user coverage. [3].

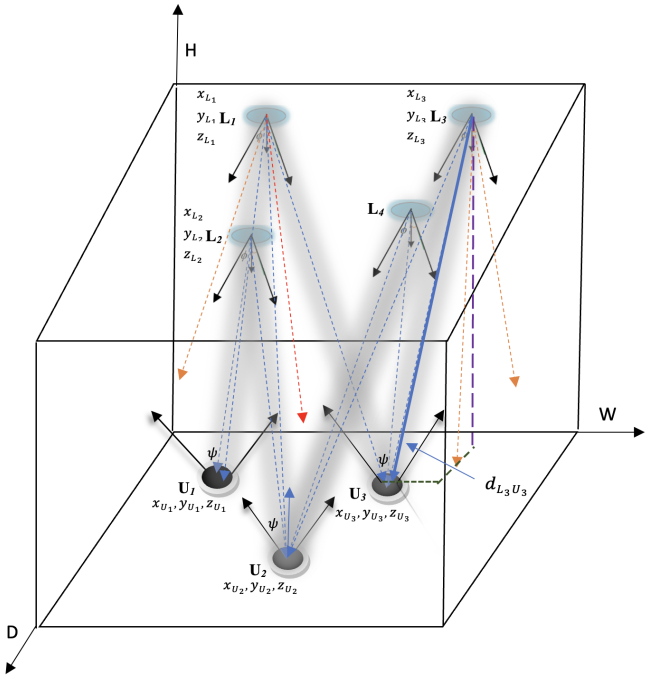


Figure 2: Indoor VLC Room Model

5 Conclusion

Visible Light Communication (VLC) stands as a promising technology offering numerous advantages, including high-speed data transmission, security, and immunity to electromagnetic interference. However, several challenges persist, notably the line-of-sight requirement, scalability issues, and interference from ambient light sources. To address these challenges, researchers have increasingly turned to machine learning, deep learning and optimization techniques. AI methods, such as artificial neural networks (ANNs) and clustering algorithms, have been employed to mitigate nonlinear distortion, compensate for system jitter, and enhance decision-making processes in VLC systems. Optimization methods have also been utilized to optimize system parameters and improve overall performance. Despite these advancements, several challenges remain unresolved. For instance, achieving seamless handover and mobility management in VLC-enabled environments, particularly for vehicular applications, remains a challenge. Additionally, the impact of atmospheric conditions on outdoor VLC performance requires further investigation.

References

[1] Vaneet Aggarwal, Zhe Wang, Xiaodong Wang, and Muhammad Ismail. Energy scheduling for optical channels with energy harvesting devices. *IEEE Transactions on Green Communications and Networking*, 2(1):154–162, 2017.

[2] Yasin Altunbas and Kadir Turk. Power allocation for indoor noma based vlc systems with meta-heuristic optimization algorithms. *IEEJ Transactions on Electrical and Electronic Engineering*, 18(11):1799–1805, 2023.

[3] Abdelbaki Benayad, Amel Boustil, Yassine Meraihi, Seyedali Mirjalili, Selma Yahia, and Sylia Mekhmoukh Taleb. An enhanced whale optimization algorithm with opposition-based learn-

ing for leds placement in indoor vlc systems. In *Handbook of Whale Optimization Algorithm*, pages 279–289. Elsevier, 2024.

[4] Abdelbaki Benayad, Amel Boustil, Yassine Meraihi, Selma Yahia, Sylia Mekhmoukh Taleb, Amylia Ait Saadi, and Amar Ramdane-Cherif. Solving the leds placement problem in indoor vlc system using a hybrid coronavirus herd immunity optimizer. *Journal of Optics*, pages 1–32, 2024.

[5] Christopher M Bishop. Pattern recognition and machine learning. *Springer google schola*, 2:645–678, 2006.

[6] Panagiotis Botsinis, Dimitrios Alanis, Simeng Feng, Zunaira Babar, Hung Viet Nguyen, Daryus Chandra, Soon Xin Ng, Rong Zhang, and Lajos Hanzo. Quantum-assisted indoor localization for uplink mm-wave and downlink visible light communication systems. *IEEE Access*, 5:23327–23351, 2017.

[7] Chen Chen, Wen-De Zhong, Dehao Wu, and Zabih Ghassemlooy. Wide-fov and high-gain imaging angle diversity receiver for indoor sdm-vlc systems. *IEEE Photonics Technology Letters*, 28(19):2078–2081, 2016.

[8] Nan Chi, Yiheng Zhao, Meng Shi, Peng Zou, and Xingyu Lu. Gaussian kernel-aided deep neural network equalizer utilized in underwater pam8 visible light communication system. *Optics express*, 26(20):26700–26712, 2018.

[9] Rui Guan, Jin-Yuan Wang, Yun-Peng Wen, Jun-Bo Wang, and Ming Chen. Pso-based led deployment optimization for visible light communications. In *2013 International Conference on Wireless Communications and Signal Processing*, pages 1–6, 2013.

[10] Yinaer Ha, Wenqing Niu, and Nan Chi. Frequency reshaping and compensation scheme based on deep neural network for a ftn cap 9qam signal in visible light communication system. In *17th International Conference on Optical Communications and Networks (ICOCN2018)*, volume 11048, pages 468–474. SPIE, 2019.

[11] Chaochuan Jia, Yang Ting, Wang Chuanjiang, and Sun Mengli. High-accuracy 3d indoor visible light positioning method based on the improved adaptive cuckoo search algorithm. *Arabian Journal for Science and Engineering*, pages 1–20, 10 2021.

[12] Musa Furkan Keskin, Sinan Gezici, and Orhan Arikan. Direct and two-step positioning in visible light systems. *IEEE Transactions on Communications*, 66(1):239–254, 2017.

[13] Faisal Nadeem Khan, Chao Lu, and Alan Pak Tao Lau. Machine learning methods for optical communication systems. In *Signal Processing in Photonic Communications*, pages SpW2F–3. Optica Publishing Group, 2017.

[14] Deok-Rae Kim, Se-Hoon Yang, Hyun-Seung Kim, Yong-Hwan Son, and Sang-Kook Han. Outdoor visible light communication for inter-vehicle communication using controller area network. In *2012 Fourth International Conference on Communications and Electronics (ICCE)*, pages 31–34. IEEE, 2012.

[15] Ishwar Ram Kumawat, Satyasai Nanda, and Ravi Maddila. *Positioning LED Panel for Uniform Illuminance in Indoor VLC System Using Whale Optimization*, pages 131–139. 01 2018.

[16] Liqun Li, Pan Hu, Chunyi Peng, Guobin Shen, and Feng Zhao. Epsilon: A visible light based positioning system. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pages 331–343, 2014.

[17] Ying-Dar Lin. Third quarter 2018 ieee communications surveys and tutorials. *IEEE Communications Surveys & Tutorials*, 20(3):1607–1615, 2018.

[18] Xingyu Lu, Chao Lu, Weixiang Yu, Liang Qiao, Shangyu Liang, Alan Pak Tao Lau, and Nan Chi. Memory-controlled deep lstm neural network post-equalizer used in high-speed pam vlc system. *Optics express*, 27(5):7822–7833, 2019.

[19] Xingyu Lu, Liang Qiao, Yingjun Zhou, Weixiang Yu, and Nan Chi. An iq-time 3-dimensional post-equalization algorithm based on dbscan of machine learning in cap vlc system. *Optics Communications*, 430:299–303, 2019.

[20] Xingyu Lu, Kaihui Wang, Liang Qiao, Wen Zhou, Yiguang Wang, and Nan Chi. Nonlinear compensation of multi-cap vlc system employing clustering algorithm based perception decision. *IEEE Photonics Journal*, 9(5):1–9, 2017.

[21] Xingyu Lu, Mingming Zhao, Liang Qiao, and Nan Chi. Non-linear compensation of multi-cap vlc system employing pre-distortion base on clustering of machine learning. In *Optical Fiber Communication Conference*, pages M2K–1. Optica Publishing Group, 2018.

[22] Xingyu Lu, Yingjun Zhou, Liang Qiao, Weixiang Yu, Shangyu Liang, Mingming Zhao, Yiheng Zhao, Chao Lu, and Nan Chi. Amplitude jitter compensation of pam-8 vlc system employing time-amplitude two-dimensional re-estimation base on density clustering of machine learning. *Physica Scripta*, 94(5):055506, 2019.

[23] Luiz Eduardo Mendes Matheus, Alex Borges Vieira, Luiz FM Vieira, Marcos AM Vieira, and Omprakash Gnawali. Visible light communication: concepts, applications and challenges. *IEEE Communications Surveys & Tutorials*, 21(4):3204–3237, 2019.

[24] Suganya Pandarinathan and R. K. Jeyachitra. Optimization of received signal strength in visible light communication using the combination of heuristic algorithms. *International Journal of Communication Systems*, 37(7):e5728, 2024.

[25] Ian C Rust and H Harry Asada. A dual-use visible light approach to integrated communication and localization of underwater robots with application to non-destructive nuclear reactor inspection. In *2012 IEEE International Conference on Robotics and Automation*, pages 2445–2450. IEEE, 2012.

[26] Stefan Schmid, Josef Ziegler, Giorgio Corbellini, Thomas R Gross, and Stefan Mangold. Using consumer led light bulbs for low-cost visible light communication systems. In *Proceedings of the 1st ACM MobiCom workshop on Visible light communication systems*, pages 9–14, 2014.

[27] D Tagliaferri and C Capsoni. High-speed wireless infrared uplink scheme for airplane passengers' communications. *Electronics Letters*, 53(13):887–888, 2017.

[28] Suseela Vappangi and Venkata Mani Vakamulla. Synchronization in visible light communication for smart cities. *IEEE Sensors Journal*, 18(5):1877–1886, 2017.

[29] Yiguang Wang, Li Tao, Xingxing Huang, Jianyang Shi, and Nan Chi. 8-gb/s rgby led-based wdm vlc system employing high-order cap modulation and hybrid post equalizer. *IEEE photonics journal*, 7(6):1–7, 2015.

[30] Zixiong Wang, Wen-De Zhong, Changyuan Yu, Jian Chen, Chin Po Shin Francois, and Wei Chen. Performance of dimming control scheme in visible light communication system. *Optics express*, 20(17):18861–18868, 2012.

[31] Arnold Wilkins, Jennifer Veitch, and Brad Lehman. Led lighting flicker and potential health concerns: Ieee standard par1789 update. In *2010 IEEE Energy Conversion Congress and Exposition*, pages 171–178. IEEE, 2010.

[32] Xingbang Wu and Nan Chi. The phase estimation of geometric shaping 8-qam modulations based on k-means clustering in underwater visible light communication. *Optics Communications*, 444:147–153, 2019.

[33] Xingbang Wu, Fangchen Hu, Peng Zou, and Nan Chi. Application of gaussian mixture model to solve inter-symbol interference in pam8 underwater visible light system communication. *IEEE Photonics Journal*, 11(6):1–10, 2019.

[34] Xingbang Wu, Fangchen Hu, Peng Zou, Xingyu Lu, and Nan Chi. The performance improvement of visible light communication systems under strong nonlinearities based on gaussian mixture model. *Microwave and Optical Technology Letters*, 62(2):547–554, 2020.

[35] Weixiang Yu, Xingyu Lu, and Nan Chi. Signal decision employing density-based spatial clustering of machine learning in pam-4 vlc system. In *Fiber Optic Sensing and Optical Communication*, volume 10849, pages 295–299. SPIE, 2018.

AI and Colorectal Polyp Detection: A Review

Mamar Khaled[1], Djamel Gaceb[1], and Fayçal Touazi[1]

[1] ***Department of Computer Science, University of M'hamed Bougara, Boumerdes, Algeria***

Abstract

Colorectal cancer ranks second in terms of the number of cancer-related deaths worldwide and third in terms of the frequency of cancer cases, making up 10% of all cancer cases. Numerous reported cases constitute an increasing global public health challenge. In order to reduce ColoRectal Cancer (CRC) morbidity and death in the future, it is imperative to raise awareness about CRC to encourage healthy lifestyle choices, innovative CRC management techniques, and the adoption of global screening programs. Early detection can greatly boost the survival percentage; several medical imaging tools are available to aid in the diagnosing process. Generally, colorectal polyps are benign tumors that can turn into CRC and their early detection based on medical imaging is very complex even for experienced Doctor, this is why it is necessary to develop innovative detection techniques using computer vision. This paper presents a review of some approaches proposed in the literature for automatic detection, localization , segmentation and classification of colorectal polyps.

Keywords: Colorectal cancer, AI, Deep learning, Polyp detection, Polyp Localization, Polyp Classification.

1 Introduction

After breast and lung cancer, ColoRectal Cancer (CRC) was the third most prevalent cancer in 2020 in terms of new cases (1. 93 million cases) and the second-leading cause of death in the world (916 000 deaths). Cancer claims the lives of millions of people each year[27, 28] . CRCs are usually developed from precursors, such as adenomatous polyps with a rather slow rate of progression, and sessile serrated lesions. The risk factors include smoking, eating red or processed meat, drinking, living a sedentary lifestyle, being overweight, and having hereditary illnesses. However, less than 5% of colon cancer cases are linked to hereditary issues. An accurate diagnosis of a CRC can be made using biopsy-proven tissues found during a colonoscopy. A CRC frequently changes from a benign polyp to a malignant one [19]. Colorectal polyps are benign tumors that can turn into CRC and their early detection based on medical imaging is very complex even for experienced Doctor, this is why it is necessary to develop innovative detection techniques using computer vision. This paper presents a review of different approaches proposed in the literature for automatic detection, classification, localization and segmentation of colorectal polyps. This review paper is organized as follow: In the second section we will describe the polyp types , the different colorectal cancer screening techniques and the tow categories of existed computer-aided systems and their tasks. The third section presents the different datasets used in the literature to learn a deep learning based model for the classification or segmentation of polyps. Finally, the section four is reserved for related works, presenting the different approaches developed for colon cancer diagnosis using Deep Learning (DL).

2 Polyp types, colorectal cancer screening and computer aided systems

Polyps (show Figure 1) grow in two different shapes: sessile polyps are more common and harder to detect, they lie flat against the surface of the colon's lining, pendunculated polyps are mushroom-like tissue growths that attach to surface of the colon's mucous membrane by a long thin stalk. The Gastrointestinal Endoscopy researchers have classified polyps into two main types :

- Non-neoplastic polyps: This type of polyp can appear in any part of the colon, with a diameter less than 1 cm, it does not cause complications because it is considered non-cancerous, periodic examinations are recommended in this case.
- Neoplastic polyps: This type is recognized as an important precursor of the majority of colorectal cancers, depending on the villous tissues, and pathological classification.

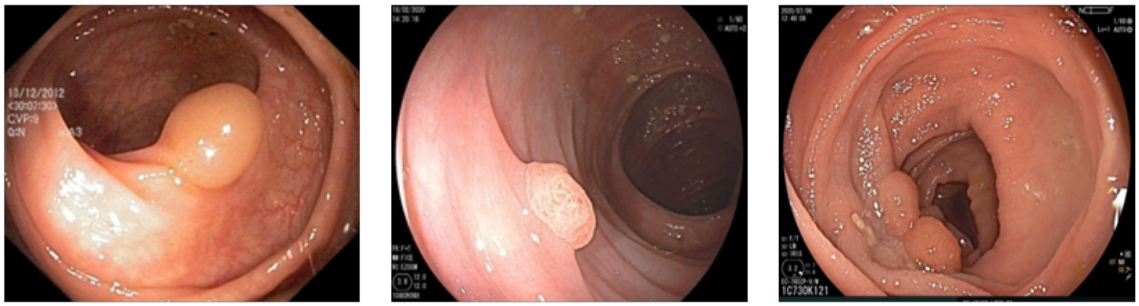


Figure 1: Polyps examples

The screening programs are crucial in greatly enhancing the survival rate, because they assist patients in starting treatments even before these symptoms manifest. Several tests available for colorectal cancer screening can detect precancerous polyps and can lead to cancer prevention and/or detect cancers at an early, more treatable stage:

- Colonoscopy: allows a clinician to see the lining of the entire colon, including the rectum.
- Sigmoidoscopy: rarely used, allows a clinician to directly view the lower part of the colon.
- CT colonography: Computed Tomography Colonography (CTC, sometimes called "Virtual Colonoscopy") is a test that uses a CT scanner to take images of the entire colon. These images are two- and three-dimensional and are reconstructed to allow a radiologist to determine if polyps or cancers are present. The major advantages of CTC are that it does not require sedation, it is noninvasive, the entire bowel can be examined, and abnormal areas can be detected about as well as with traditional (optical) colonoscopy.
- Wireless Capsule Endoscopy: Known as capsule endoscopy, it's a technique that uses a pill camera to take images of the intestinal lumen.

In addition, computer aid system is used, there are two categories:

- CADe: Computer-Aided Detection systems geared for the location of lesions in medical images, it aims to decrease the rate of missed polyps during colonoscopy and ultimately increase the performance of the endoscopists.
- CADx: Computer-Aided Diagnosis systems perform the characterization of the lesions, for example, the distinction between benign and malignant tumors, it has the real-time interpretation of the polyp optical diagnosis, potentially able to reduce the rate of unnecessary polypectomies of non-neoplastic lesions.

3 Datasets used

The following available datasets can be used by computer aided system to learn DL models for the detection or localization of colorectal polyps during colonoscopy:

1. CVC-EndoSceneStill [25] is a manually segmented dataset that combines and enhances the two CVC-ColonDB and CVC-ClinicDB datasets. It has 912 images from 44 video clips of 36 patients. A binary mask for the lumen, specular lights, polyp, and vacant areas is included in the annotations. Additionally, the owners provide separation into training, validation, and test sets.
2. CVC-VideoClinicDB[2, 6] assembles 18 video clips that demonstrate a polyp. Annotations in this instance are binary masks that display ellipses that roughly represent the polyp area. Out of a total of 10.924 frames, 9.221 polyp frames are annotated in this manner.
3. ETIS-Larib [13, 10] offers 44 distinct polyps in 196 White Light (WL) images from 34 sequences. The specified ground truth consists of manually annotated binary masks.
4. ASU-Mayo Clinic [24] gives binary masks for 18 additional test videos and 20 annotated videos for which there is no ground truth. WL and Narrow Band Imaging are present in the frames (NBI).

5. Kvasir-SEG [11] gives 1.000 polyp images, a carefully delimited binary mask together with the accompanying bounding boxes in a JSON file. Additionally, it is a portion of the HyperKvasir gastrointestinal endoscopy dataset [4].

6. PICCOLO WL and NBI (Narrow Band Imaging) colonoscopic dataset [22] consists of 3.433 polyp frames from 76 lesions on 40 patients make up this dataset. It is further segmented into training, validation, and test sets to ensure patient independence. Additionally, the lesions' sizes, Paris and NICE (NBI International Colorectal Endoscopic) classifications, and histological diagnoses are provided in this dataset's clinical metadata.

7. The Colonoscopic dataset proposed by Mesejo et al. [14] is composed of 15 serrated adenomas, 21 hyperplastic lesions, and 40 adenomas, all have WL and NBI videos available.

Designation	Type	Download Link
CVC-EndoSceneStill[25]	Public	https://pages.cvc.uab.es/CVC-Colon/index.php/databases/cvc-endoscenestill/
CVC-VideoClinicDB[2, 6]	Public	http://mv.cvc.uab.es/projects/colon-qa/cvccolondb
ASU-Mayo Clinic	Public	https://polyp.grand-challenge.org/AsuMayo/
ETIS-Larib[13, 10]	Public	https://polyp.grand-chall enge.org/EtisLarib/
Kvasir-SEG[11]	Public	https://datasets.simula.no/kvasir-seg/download
PICCOLO WL and NBI (Narrow Band Imaging) colonoscopic dataset[22]	Public	https://www.biobancovasco.bioef.eus/en/Sample-and-data-catalog/Databases/PD178-PICCOLO-EN.html
The Colonoscopic dataset proposed by Mesejo et al.[14]	Public	https://www.depeca.uah.es/colonoscopy$_d ataset$/

Table 1: Most used datasets

4 Related works

In literature, the CRC related works can be organized into three categories: polyp detection/ localization, segmentation and classification.

4.1 Polyp detection and localization

The models dedicated to this task, aims to detect and locate a polyp are the trigger event for CRC diagnosis. Therefore, this is directly related to improving the ADR (Adenoma Detection Rate), as DL models will help endoscopists to ensure that no lesions are missed during the colonoscopy procedure that would eventually develop CRC. So, these DL methods have a highly important clinical implication and the eventual increment of the ADR, will lead to a reduction of interval CRC and its associated mortality. In this regard, subtle lesions are prone to be missed even for trained endoscopists, so CADe systems would play an important role. In this regard, a YOLOv3 CADe system, which reached a sensitivity comparable to experts and better than physicians in training, was examined by Guo et al. [9] using 50 short videos of one or two polyps with a mean size of 3.5 $\pm$ 1.5 mm. This fact highlights the usefulness of AI and DL for small, subtle lesions that even highly skilled endoscopists might overlook. After analyzing five randomized studies, Barua et al.[1] observed in their review that while the number of big adenomas (¿5 mm) remained constant, the number of tiny adenomas (5 mm) identified during colonoscopy helped by AI methods was higher than without AI assistance. Conversely, Sanchez-Peralta et al. [19], focusing on the technical aspects of the methods used for DL, thoroughly examined the state of the art and discovered 26 papers pertaining to localization and detection. They discovered that while both end-to-end and hybrid approaches are present the usage of end-to-end approaches is on the rise. In the same way, Nogueira Rodríguez et al. [15] included 21 works for detection and localization in conventional colonoscopy, identifying those able to detect multiple polyps in real time. Pacal et al. [16] examined 35 studies using a variety of imaging modalities. However, the majority of the articles included in the evaluations focused on still frames from the datasets, either public or private. So, they concluded that it is not possible to take advantage of the temporal coherence of a polyp's presence in a movie.

4.2 Polyp segmentation

After a polyp's presence in a certain frame has been determined, it may be helpful to identify the area that is thought to be a lesion in order to help with its removal if needed. If this is done, it's also important to consider a safety margin. Consequently, segmentation techniques ought to be used for this task in order to pinpoint the polyp's exact outline. A precise polyp segmentation will help endoscopists evaluate the resection margins. So, it's critical to remove the lesion entirely, making sure to eliminate any traces that can spread or develop into CRC. Comparable to the methods for localization and detection, segmentation models have also been reviewed in two works [20, 31]. Data augmentation is one of the most often used techniques for polyp segmentation [20, 29], though the most advantageous transformations are not universally agreed upon and are primarily chosen through trial and error and the experience of the searcher. This is demonstrated by a study [23] that examined the effects of various transformations on two publicly accessible datasets and found that, although polyp segmentation using CVC-EndoSceneStill benefits from pixel-based transformations like brightness and contrast adjustments, image-based transformations—particularly rotation and shear—are advised if Kvasir-SEG is used. In any case, it is crucial to note that improved performance is not always the result of including more intense data augmentation[21]. The most popular methodology is the combination of end-to-end and semantic segmentation models [3], which indicates that the work is completed in its entirety. Consequently, it is recommended to adopt encoder-decoder designs. The decoder retrieves the spatial information that was lost during the preceding processing while the encoder converts the input image into a feature vector that captures the context information. These encoder-decoder architectures can be specifically built for polyp segmentation, or they can be based on off-the-shelf networks. Furthermore, overlap measures are typically considered. The Jaccard index, sometimes called the Intersection over Union (IoU), or the Dice index are commonly employed; however, it is also preferable to incorporate additional distance metrics that are appropriate for tiny segments, including the Hausdorff or Mahalanobis distances[19]. The metrics listed in the polyp detection section may also be applied, provided that they are computed at the pixel-level. This is because semantic segmentation may also be thought of as pixel-level detection, where each pixel is labeled as either belonging to the polyp class or not. Since the polyp area is typically considerably smaller than the background class, it is vital to stress that in this circumstance, metrics involving true negatives, like the specificity, may get high values even when the polyp is poorly or not detected at all due to the unbalanced situation.

4.3 Polyp Classification

The third task called polyp characterisation or "optical biopsy" or "optical diagnosis," which provides a clinical diagnosis instantaneously without requiring the removal of the lesion and sending the sample for histological examination, is made possible by classification methods [26]. In this approach, time and expenses can be decreased without sacrificing patient safety by avoiding excising lesions that have the potential to become malignant. Real-time, in-situ correct diagnosis would be beneficial for the "diagnose and leave behind" strategy, provided that the techniques exceed the 90% minimum value for the Negative Predictive Value (NPV) that the American Society for Gastrointestinal Endoscopy has suggested [8].In this section, we refer to the process of identifying the type of lesion once it has been found, while polyp detection can also be viewed as a classification between healthy tissue and lesion. Reviews [29, 31] consistently reported much fewer works for colonoscopy WL images-based polyp classification, which may be because most publicly available datasets lack clinical information. The current state of affairs impedes the broader advancement of CADx systems across the scientific community and highlights the necessity of collaborative efforts to generate these kinds of datasets. Furthermore, the utilization of other imaging modalities for polyp characterization, such as NBI, confocal endo-microscopy, or magnification chromo endoscope, is another reason for this lack of techniques [30, 7]. Due to the fact that in a clinical setting, WL is typically used for lesion detection and, if available, NBI imaging is switched to for diagnosis because it highlights patterns that are helpful for diagnosis, these classification methods may therefore use any of the various approaches for classifying colorectal lesions, but they typically rely on NBI images rather than WL images [11]. For instance, Patino-Barrientos et al. [17] classified polyps in accordance with the Kudo's pit pattern schema by using a pretrained VGG model as a feature extractor. Their method beat conventional procedures where features are manually retrieved, and they attained 83% accuracy and F1-score. Rodriguez-Diaz et al. [18] used a new scale to categorize previously segmented lesions into non-neoplastic (such as hyperplastic polyps and polypoid seeming normal colonic mucosa) and neoplastic (such as tubular adenomas, tubule villous adenomas, and adenocarcinomas). Although the NPV in this instance was 0.91, the model can only be used with near-focus NBI polyp images. More

recently, Jinet al. [12] went one step farther and overlaid a heat map indicating the probability within the image to provide an interpretable explanation to the optical diagnostic. Although it would be more convenient to establish a diagnosis per polyp, Byrne et al. [5] trained a CNN to classify NBI frames into type 1 and 2 of the NICE classification, achieving an accuracy of 94% in 106 diminutive polyps. Metrics for polyp classification are typically calculated at the frame level.

5 Conclusion

Colonoscopy is a reference technique in screening programs that are used for early detection of precursor lesions, and it helps to reduce the death rate caused by CRC that is known by its high value in the worldwide compared to other types of cancers. This review underscores the critical role of artificial intelligence in detection and segmentation of colorectal polyps, the crucial precursors of colorectal cancer. Promising results in identifying and classifying polyps have been observed through the application of deep learning models and classification methods. Reducing the mortality rate from colorectal cancer requires the application of early detection of precursor lesions. This review also presents various existing screening techniques, polyp types, and the importance of accurate diagnosis in improving patient outcomes. Overall, the research highlighted in this paper demonstrates the potential of AI in enhancing survival rates for individuals at risk of colorectal cancer. Forther advancements in AI technology hold great promise for improving the detection and localization of colorectal polyps, ultimately leading to better patient care and outcomes.

References

[1] Indra Barua, Daniel G. Vinsard, Hans Christian Jodal, Magnus Løberg, Mette Kalager, Øyvind Holme, Masaki Misawa, Magnus Bretthauer, and Yutaka Mori. Artificial intelligence for polyp detection during colonoscopy: a systematic review and meta-analysis. *Endoscopy*, 53(3):277–284, March 2021.

[2] Jorge Bernal, Alexandre Histace, Marc Masana, Quentin Angermann, Carlos Sánchez-Montes, Carmen Rodríguez de Miguel, Mohamed Hammami, Abel García-Rodríguez, Hector Córdova, Olivier Romain, Gloria Fernández-Esparrach, Xavier Dray, and F. Javier Sánchez. GTCreator: a flexible annotation tool for image-based datasets. *International Journal of Computer Assisted Radiology and Surgery*, 14(2):191–201, February 2019.

[3] Jorge Bernal, Nima Tajkbaksh, Francisco Javier Sánchez, Bogdan J. Matuszewski, Hao Chen, Lequan Yu, Quentin Angermann, Olivier Romain, Bjørn Rustad, Ilangko Balasingham, Konstantin Pogorelov, Sungbin Choi, Quentin Debard, Lena Maier-Hein, Stefanie Speidel, Danail Stoyanov, Patrick Brandao, Henry Córdova, Cristina Sánchez-Montes, Suryakanth R. Gurudu, Gloria Fernández-Esparrach, Xavier Dray, Jianming Liang, and Aymeric Histace. Comparative validation of polyp detection methods in video colonoscopy: Results from the miccai 2015 endoscopic vision challenge. *IEEE Transactions on Medical Imaging*, 36(6):1231–1249, 2017.

[4] Håvard Borgli, Vegard Thambawita, Pia H. Smedsrud, Steven Hicks, Debesh Jha, Sindre L. Eskeland, Kristoffer R. Randel, Kirill Pogorelov, Markus Lux, Duy-Ton Nguyen, Dag Johansen, Christian Griwodz, Henrik K. Stensland, Eduardo Garcia-Ceja, Petter T. Schmidt, Hans L. Hammer, Michael A. Riegler, Pål Halvorsen, and Thomas de Lange. Hyperkvasir, a comprehensive multiclass image and video dataset for gastrointestinal endoscopy. *Scientific Data*, 7(1):283, August 2020.

[5] M.F. Byrne, N. Chapados, F. Soudan, C. Oertel, M. Linares Pérez, R. Kelly, N. Iqbal, F. Chandelier, and D.K. Rex. Real-time differentiation of adenomatous and hyperplastic diminutive colorectal polyps during analysis of unaltered videos of standard colonoscopy using a deep learning model. *Gut*, 68:94–100, 2019.

[6] C. Cao, R. Wang, Y. Yu, H. Zhang, Y. Yu, and C. Sun. Gastric polyp detection in gastroscopic images using deep neural network. *PLoS One*, 16(4):e0250632, April 2021.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Vittorio Ferrari,

Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 833–851, Cham, 2018. Springer International Publishing.

[8] Michael Ferlitsch, Alan Moss, Claudio Hassan, Pankaj Bhandari, Jean-Marc Dumonceau, George Paspatis, Rafael Jover, Christoph Langner, Marcel Bronzwaer, Karthik Nalankilli, Peter Fockens, Rami Hazzan, Israel M. Gralnek, Michael Gschwantler, Eva Waldmann, Philipp Jeschek, Dietrich Penz, Didier Heresbach, Luc Moons, Anna Lemmers, Konstantinos Paraskeva, Jens Pohl, Thierry Ponchon, Jan Regula, Antonio Repici, Mark D. Rutter, Nicholas G. Burgess, and Michael J. Bourke. Colorectal polypectomy and endoscopic mucosal resection (emr): European society of gastrointestinal endoscopy (esge) clinical guideline. *Endoscopy*, 49(3):270–297, March 2017.

[9] Zhiqiang Guo, Daiki Nemoto, Xiang Zhu, Qing Li, Masashi Aizawa, Kazuhiko Utano, Nobuyuki Isohata, Sho Endo, Akira Kawarai Lefor, and Kenji Togashi. Polyp detection algorithm can detect small polyps: Ex vivo reading test compared with endoscopists. *Digestive Endoscopy*, 33(1):162–169, January 2021.

[10] Qibin Hou, Daquan Zhou, and Jiashi Feng. Coordinate attention for efficient mobile network design. *CoRR*, abs/2103.02907, 2021.

[11] Debesh Jha, Pia H. Smedsrud, Michael A. Riegler, Pål Halvorsen, Thomas de Lange, Dag Johansen, and Håvard D. Johansen. Kvasir-seg: A segmented polyp dataset, 2019.

[12] E.H. Jin, D. Lee, J.H. Bae, H.Y. Kang, M.S. Kwak, J.Y. Seo, J.I. Yang, S.Y. Yang, S.H. Lim, J.Y. Yim, J.H. Lim, G.E. Chung, S.J. Chung, J.M. Choi, Y.M. Han, S.J. Kang, J. Lee, H.C. Kim, and J.S. Kim. Improved accuracy in optical diagnosis of colorectal polyps using convolutional neural networks with visual explanations. *Gastroenterology*, 158:2169–2179.e8, 2020.

[13] Glenn Jocher. YOLOv5 by Ultralytics, 2020. Accessed: 2024-10-12.

[14] Pablo Mesejo, Daniel Pizarro, Armand Abergel, Olivier Rouquette, Sylvain Beorchia, Laurent Poincloux, and Adrien Bartoli. Computer-aided classification of gastrointestinal lesions in regular colonoscopy. *IEEE Transactions on Medical Imaging*, 35(9):2051–2063, 2016.

[15] Alba Nogueira-Rodríguez, Rubén Domínguez-Carbajales, Hugo López-Fernández, Águeda Iglesias, Joaquín Cubiella, Florentino Fdez-Riverola, Miguel Reboiro-Jato, and Daniel Glez-Peña. Deep neural networks approaches for detecting and classifying colorectal polyps. *Neurocomputing*, 423:721–734, 2021.

[16] Ishak Pacal, Dervis Karaboga, Alper Basturk, Bahriye Akay, and Ufuk Nalbantoglu. A comprehensive review of deep learning in colon cancer. *Computers in Biology and Medicine*, 126:104003, 2020.

[17] S. Patino-Barrientos, D. Sierra-Sosa, B. Garcia-Zapirain, C. Castillo-Olea, and A. Elmaghraby. Kudo's classification for colon polyps assessment using a deep learning approach. *Applied Sciences*, 10:501, 2020.

[18] E. Rodriguez-Diaz, G. Baffy, W.K. Lo, H. Mashimo, G. Vidyarthi, S.S. Mohapatra, and S.K. Singh. Real-time artificial intelligence-based histologic classification of colorectal polyps with augmented visualization. *Gastrointestinal Endoscopy*, 93:662–670, 2021.

[19] Luisa F. Sánchez-Peralta, Luis Bote-Curiel, Artzai Picón, Francisco M. Sánchez-Margallo, and J. Blas Pagador. Deep learning to find colorectal polyps in colonoscopy: A systematic literature review. *Artificial Intelligence in Medicine*, 108:101923, 2020.

[20] Younghak Shin, Hemin Ali Qadir, Lars Aabakken, Jacob Bergsland, and Ilangko Balasingham. Automatic colon polyp detection using region based deep cnn and post learning approaches. *IEEE Access*, 6:40950–40962, 2018.

[21] Younghak Shin, Hemin Ali Qadir, Lars Aabakken, Jacob Bergsland, and Ilangko Balasingham. Automatic colon polyp detection using region based deep cnn and post learning approaches. *IEEE Access*, 6:40950–40962, 2018.

[22] Luisa F. Sánchez-Peralta, J. Blas Pagador, Artzai Picón, Ángela J. Calderón, Francisco Polo, Nicolás Andraka, Raquel Bilbao, Ben Glover, Clara L. Saratxaga, and F. Manuel Sánchez-Margallo. Piccolo white-light and narrow-band imaging colonoscopic dataset: A performance comparative of models and datasets. *Applied Sciences*, 10(23):8501, 2020.

[23] Luisa F. Sánchez-Peralta, Artzai Picón, Francisco M. Sánchez-Margallo, and J. Blas Pagador. Unravelling the effect of data augmentation transformations in polyp segmentation. *International Journal of Computer Assisted Radiology and Surgery*, 15(12):1975–1988, December 2020.

[24] Nassir Tajbakhsh, Sanjiv R. Gurudu, and Jianming Liang. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE Transactions on Medical Imaging*, 35(2):630–644, February 2016.

[25] David Vázquez, Jorge Bernal, F. Javier Sánchez, Gloria Fernández-Esparrach, Antonio M. López, Adriana Romero, Michal Drozdzal, and Aaron Courville. A benchmark for endoluminal scene segmentation of colonoscopy images. *Journal of Healthcare Engineering*, 2017(1):4037190, 2017.

[26] Ana Wilson. Optical diagnosis of small colorectal polyps during colonoscopy: When to resect and discard? *Best Practice Research Clinical Gastroenterology*, 29(4):639–649, 2015. Image-enhanced endoscopy: clinical frontier and future perspectives.

[27] World Health Organization. World health organization, 2024. Accessed: 2024-03-12.

[28] Yue Xi and Pengfei Xu. Global colorectal cancer burden in 2020 and projections to 2040. *Translational oncology*, 14(10):101174, 2021.

[29] Lequan Yu, Hao Chen, Qi Dou, Jing Qin, and Pheng Ann Heng. Integrating online and offline three-dimensional deep learning for automated polyp detection in colonoscopy videos. *IEEE Journal of Biomedical and Health Informatics*, 21(1):65–75, 2017.

[30] Yixuan Yuan, Wenjian Qin, Bulat Ibragimov, Bin Han, and Lei Xing. Riis-densenet: Rotation-invariant and image similarity constrained densely connected convolutional network for polyp detection. In Alejandro F. Frangi, Julia A. Schnabel, Christos Davatzikos, Carlos Alberola-López, and Gabor Fichtinger, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pages 620–628, Cham, 2018. Springer International Publishing.

[31] Ruikai Zhang, Yali Zheng, Carmen C.Y. Poon, Dinggang Shen, and James Y.W. Lau. Polyp detection during colonoscopy using a regression-based convolutional neural network with a tracker. *Pattern Recognition*, 83:209–219, 2018.

Artificial-Intelligence-enhanced solving methods for the Vehicle Routing Problem

Souad Abdoune and Menouar Boulif

LIMOSE laboratory
Department of Computer Science, M'hamed Bouguerra University Boumerdes, Algeria
s.abdoune@univ-boumerdes.dz, boumen7@gmail.com

Abstract

The Vehicle Routing Problem (VRP) is among the most challenging NP-hard problems, that attracted the attention of the supply-chain research community.Consequently, numerous extensive research efforts have been undertaken in the last decades to develop resolution approaches to solve it. AI approaches such as machine learning and evolutionary algorithms offer promising methods to enhance solution quality and reduce time complexity during VRP resolution. In this paper, we provide an overview of recent VRP works with a particular focus on those employing artificial intelligence approaches.

Keywords: Vehicle Routing Problem, Solving methods, Artificial Intelligence, Machine Learning, Deep Learning.

1 Introduction

The Vehicle Routing Problem (VRP) emerged as a generalization of the Travelling Salesman Problem (TSP) [10], which was first conceptualized in 1959 by Dantzig and Ramser [5]. Originally devised to optimize the delivery of oil demands to a network of gas stations from a central terminal while minimizing travel distance [10], VRP has since evolved into a cornerstone problem within the field of operational research for logistics management.

VRP represents a fundamental challenge in efficiently managing transportation routes for fleets of vehicles tasked with serving a diverse set of customers across different geographical locations from a central depot. This optimization task is compounded by a myriad of constraints, including time windows, periodicity constraints, vehicle capacity limitations, and considerations pertaining to the number and locations of depots. The overarching objective of VRP is to optimize one or more factors, such as minimizing travel time, distance and cost, while concurrently maximizing customer satisfaction and resource utilization.

To resolve the complexities inherent in VRP and achieve optimal or good solutions, diverse resolution methods have been developed. These methods encompass both exact and approximate algorithms.

Recent research in the field of VRP has increasingly emphasized hybrid algorithms, combining multiple resolution methods to efficiently enhance solution quality. Our paper concentrates on those that leverage Artificial Intelligence (AI) tools. These approaches aim to optimize VRP solutions by integrating AI techniques, such as machine learning and evolutionary algorithms, promising superior solutions with reduced computational burden.

The rest of this paper is structured as follows: Section 2 gives a comprehensive overview of the Vehicle Routing Problem, delineating its foundational principles. The next section elucidates several resolution methods employed in solving VRP. Subsequently, Section 4 delineates AI-based techniques for addressing VRP. Finally, we provide a conclusion in the last section.

2 VRP background

The Vehicle Routing Problem is a classical optimization challenge with widespread applications across various industries. In its standard form, VRP involves determining the most efficient manner for a fleet of homogeneous vehicles to service a set of customers situated at various locations from a single depot [4], while adhering to the following constraints [44, 49]:

- Each vehicle's route must commence and conclude at the depot, which itself requires no service.

- Each customer must be visited exactly once and by only one vehicle.
- The total demand along each route must not exceed the capacity of the corresponding vehicle.

VRP is a complex combinatorial optimization problem within the domain of transportation logistics. It represents a combination of two well known NP-hard problems [21]:

- `The Travelling Salesman Problem:` determination of the optimal routes that visit all customers precisely once while minimizing the travelling distance [42].
- `The Bin Packing Problem:` seeks for the optimal arrangement of a set of predefined objects into a given number of bins, each with a predetermined capacity. [3].

To understand VRP, we must shed some light on its fundamental components, each being an integral part of the problem definition. These components encompass [15]:

- `Customers:` a set of clients or designated delivery points necessities visits by the fleet of vehicles.
- `Depots:` stores of products locations each representing both the starting and the ending point of a vehicle trip.
- `Vehicles:` available transportation means used for product delivery to customers, characterized by a loading capacity.
- `Demands:` customer's requests of one or more products.
- `Routes:` a route is an order of customers to be visited by a vehicle in that order to satisfy their demands in one trip.
- `Constraints:` restrictions that must be satisfied by each solution in order to be accepted.
- `Objective function:` a function that measures the quality of solutions. The objective function definition can be done by using one or more criteria.

Real-life applications often necessitate the consideration of various restrictions to meet specific operational requirements. These may include constraints for:

- `Vehicles:` deal with restrictions associated to vehicles, such as: capacity, volume, speed, number of vehicles used during the delivery process, vehicle homogeneity or heterogeneity, etc.
- `Depots:` define the number of depots from which servicing customer starts and then ends.
- `Return:` By adding these constraints, we can impose that instead of using depots as concluding points for vehicle routes, we resort to any other designated locations.
- `Time windows:` Time intervals are assigned to each customer, and must be respected during the delivery process.
- `Periodicity:` represents the planning horizon during which the customer requests will reoccur.

These constraints and/or additional considerations give rise to various extensions and variants of the VRP problem, including:

- Capacitated Vehicle Routing Problem (CVRP): involves a fleet of vehicles with limited capacity. Each vehicle in the fleet has a maximum capacity, and the total demand served by each vehicle must not exceed its capacity [46].
- Vehicle Routing Problem with Time Windows VRPTW): each customer has a specified time interval during which it can be serviced. These time windows represent the availability of customers to receive their requests and must be respected during the delivery process [14]. There are two types of VRPTW which are rigid VRPTW and released VRPTW. The first dictates serving operations to happen within their specified time windows [24], whereas the second allows for a certain margin of deviation in the timing but with additional penalties [24].
- Multi Depot Vehicle Routing Problem (MDVRP): servicing customers can emanate from multiple depots. Each depot serves a subset of customers, and vehicles must plan routes to efficiently serve customers from different depots [46].

- Periodic Vehicle Routing Problem (PVRP): conduct the delivery process by taking into account the fact that customers require their orders to be honored periodically during a predefined planning horizon [7].
- Dynamic Vehicle Routing Problem (DVRP): customer requests are not predetermined before the distribution, and continue to evolve throughout the delivery process[37].
- Vehicle Routing Problem with Pickup and Delivery (VRPPD): vehicles retrieve items or goods from specific locations (pickup stops) then transport them to other specific destinations (drop-off stops). In comparison to others, this variant impose precedence constraints between route stops [48, 13].
- Open Vehicle Routing Problem (OVRP): After delivery operations, vehicles have the option to return to the initial depot or continue to a different destination [22].
- Split Delivery Vehicle Routing Problem (SDVRP): Allows splitting customer requests and servicing them in multiple tours from different depots, using multiple vehicles [2].
- Electric Vehicle Routing Problem (EVRP): involves servicing customers using electric vehicles, with the primary objective of minimizing energy consumption [39].
- Green Vehicle Routing Problem (GVRP): seeks to reduce CO_2 emissions generated by the fleet of vehicles. This objective can be evaluated by considering the amount of consumed fuel and/or by choosing routes with low congestion rate. The importance accorded to environmental issues triggered several related works, and gave rise to several sub-variants to GVRP[40].

3 VRP solving methods

Since the establishment of VRP, the research community proposed a multitude of algorithms to solve the problem in its original form. Afterwards, several methods were developed to tackle its numerous subsequent variants. These approaches can be broadly categorized into two main classes: exact and approximate.

3.1 Exact methods

Exact methods refer to a collection of algorithmic approaches designed to determine the optimal solution to a given problem. When tackling complex problems, these methodologies are only effective when applied to small-scale VRP instances containing no more than 50 customers. Their main characteristic lies in their ability to rigorously explore the solution space to guarantee the identification of the globally optimal solution [27]. In the literature, there are a lot of such methods such as [36, 30]:

3.1.1 Branch and Bound

The Branch and Bound method, introduced by Land and Doig in 1960 [35], is an algorithmic technique for solving combinatorial optimization problems based on the divide and conquer technique. It divides the problem into smaller sub-problems giving rise to a tree structure. By systematically exploring the tree, the algorithm eliminates branches that cannot lead to an optimal solution thanks to a bounding scheme. The algorithm ends when, after conducting an exhaustive search of the tree, it is able to locate an optimal solution. Among the works that adopted such an approach we have [50, 47].

3.1.2 Branch and cut for integer linear programming formulations

Branch and Cut is an advanced optimization technique that employs a specialized tree structure. This method iteratively generates sub-problems within the "Branch and Cut" tree, consisting of current, active, and interactive nodes. The algorithm iterates by exploring feasible solutions and iteratively refining them to approach the optimal solution. This process continues until convergence or predefined termination conditions are met. Among the works that adopted such an approach we can cite [34, 45].

3.1.3 Dynamic programming

Dynamic programming, introduced by Richard Bellman in 1950, relies on Bellman's principle of optimality, stating that the optimal solution to a dynamic optimization problem is the combination of the optimal solutions to its sub-problems. This principle guides the process of deriving the optimum. In fact, solutions are computed incrementally, starting from the smallest sub-problems and gradually extending to larger ones. This systematic approach enables the derivation of optimal solutions for complex problems through the aggregation of optimal solutions to simpler sub-problems. In the VRP context, [20, 32] are among the works related to such an approach.

3.2 Approximate methods

Approximate methods, distinct from exact techniques, comprise algorithms aimed at finding good feasible solutions to problems within limited computational resources and time frames, without guaranteeing the optimality of the final solution. They are classified into two main categories: heuristic and meta-heuristic methods [19].

3.2.1 Heuristic methods

Heuristic methods are tailored to efficiently solve a specific problem, and so they have little or no ability to be generalized. Examples of heuristic algorithms commonly used to solve VRP (VRP) include the Nearest Neighbor, Insertion Heuristic, and Sweep Algorithm. However, as mentioned earlier, it is important to note that while these methods often provide efficient solutions, they do not guarantee optimality [23]. Some works related to this category can be found in [31, 52].

3.2.2 Meta-heuristic methods

Meta-heuristics represent search methodologies designed specifically for addressing intricate optimization challenges. They serve as viable alternatives to heuristic techniques, particularly in situations where problem-specific heuristics are not readily available [36]. These methods offer versatile approaches to solving complex optimization problems by iteratively exploring solution spaces and converging towards optimal or near-optimal solutions [23]. The majority of VRP related research works fall into this category. Examples of widely used meta-heuristics include Simulated Annealing, Tabu Search, and Particle Swarm Optimization. The works [17, 38, 51] pertain to this category.

4 Artificial intelligence based Approaches

Many techniques and methods have been applied over time to address the complex issues raised by the Vehicle Routing Problem (VRP). Specifically, a variety of artificial intelligence (AI) techniques have been created and utilized to tackle this intricate optimization challenge. These methods not only seek to optimize conventional VRP parameters but also broaden the scope of VRP optimization to include new challenges in autonomous vehicle navigation. For instance, enabling vehicles to find their travel paths with less complexity requires exploring innovative algorithms and optimization techniques within the realm of AI, ensuring real-time adaptability to dynamic environments. By integrating trajectory discovery into AI-based VRP solutions, researchers aim to enhance navigation capabilities and improve autonomous vehicle technology. Below, we outline some of these approaches.

4.1 Evolutionary Algorithms

Evolutionary Algorithms are approximate optimization methods introduced by JH. Holland in 1975 [26]. Evolutionary methods draw inspiration from biological processes. They represent a prominent area of study within the field of optimization algorithms, focusing on manipulating a set of candidate solutions using genetic principles and natural selection mechanisms to identify solutions [58] which is a kind of intelligent behavior [12, 33]. These algorithms aim to emulate the evolutionary process observed in nature to address complex problem instances [58]. The evolutionary process within these methods typically encompasses four fundamental concepts: representation, evaluation, selection and reproduction [1].

- `Representation:` defines how to encode the solutions to get a chromosomal representation. By using this encoding scheme, a population of chromosomes called individuals is generated.
- `Evaluation:` assigns a measure of quality called fitness to each individual. It is usually directly derived from the value of the objective function, but for constrained problems like VRP, it must also deal with the magnitude of infeasibility of the solution.
- `Selection:` choose a subset of best fitted individuals. These are selected from the population with replacement and will be allowed in the subsequent reproduction phase to make offspring.
- `Reproduction:` encompasses mainly two genetic operators: crossing-over and mutation. In the first, selected individuals are combined to make new ones. For the other, the genetic code of some randomly chosen individuals is altered.

In the literature, there are various evolutionary algorithms, including Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Differential Evolution (DE), and Evolution Strategies (ES), are utilized to tackle the Vehicle Routing Problem (VRP). Some of the works pertaining to this category are [25, 41, 9, 43, 55].

4.2 Fuzzy logic approaches

Fuzzy Logic (FL) was proposed by L. Zadeh in 1965 [56]. Gradually FL became an important modeling tool to deal with systems containing imprecise or ill-defined concepts. In VRP related works, FL can deal for example notions such as high or low demand, route congestion, acceptable vehicle load, high or low fuel consumption to cite a few. Among the works that use FL in tackling VRP, we have [47, 6]

4.3 Machine learning algorithms

Using Machine Learning (ML) techniques to address VRP constitutes a strategic approach towards optimizing the routes taken by vehicles during customer service operations. ML models serve as powerful tools for mitigating the complexities inherent in diverse VRP variants. Numerous ML algorithms are commonly employed to solve VRP, including the following [8, 18]:

4.3.1 Deep learning (DL)

Deep learning techniques are multi-layer neural networks. They have been applied to VRP to model intricate relationships within the data. By employing DL, organizations can optimize vehicle routes considering multiple constraints and objectives, thereby improving routing efficiency and resource utilization. Some of the works that tackled VRP using DL are [16, 53].

4.3.2 Reinforcement Learning (RL)

Reinforcement Learning entails iteratively learning optimal decision-making policies through trial and error. Within VRP, reinforcement learning is utilized to determine the most efficient sequence of customer visits for vehicles, facilitating streamlined route planning and resource allocation. Among the works using such an approach we can cite [54, 28, 57, 29, 11].

After elucidating some AI-based techniques for addressing the Vehicle Routing Problem (VRP), we further explore the comparative performance of these methods alongside traditional resolution techniques. This comparative analysis evaluates the solution quality and computational time of both traditional and AI-based approaches, shedding light on their respective strengths and limitations. Exact methods are efficient approaches that yield optimal solutions within a reasonable time frame, but they are primarily suitable for small-scale problems instances. Conversely, approximate methods provide solutions of good quality quickly, without guaranteeing optimality. Meanwhile, AI-based approaches provide higher solution quality but necessitate substantial computational resources and time for training and optimization. The selection of a resolution approach depends on the problem specifications such as problem size, complexity, and available resources... However, hybridization of different methods remains a preferable strategy for achieving high-quality solutions within a reasonable time frame.

5 Conclusion

The Vehicle Routing Problem (VRP) represents a challenging combinatorial optimization task aimed at efficiently servicing a set of clients using a fleet of vehicles, while adhering to specific constraints and optimizing one or more objectives. The diverse optimization constraints and objectives gave rise to various VRP variants.

As VRP is NP-hard problem, its resolution algorithms are more directed towards approximate methods instead of exact approaches. Furthermore, in recent years, significant attention has been directed towards the development of hybrid algorithms for VRP resolution. These hybrid approaches usually combine Artificial Intelligence modeling tools with a solving method to expedite the resolution process and enhance the quality of the obtained solutions. Such hybridization strategies have shown promising results in addressing the challenges posed by VRP and its variants, especially for the new ones, such as the green VRP or the routing with electrical vehicles.

References

[1] Basma Alouane and Menouar Boulif. Fuzzy constraint prioritization to solve heavily constrained problems with the genetic algorithm. *Engineering Applications of Artificial Intelligence*, 119:105768, 2023.

[2] C. Archetti, M. G. Speranza, and A. Hertz. A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science*, 40(1):64–73, 2006.

[3] L Arnaud and K Nathalie. Bin packing problem with priorities and incompatibilities using pso: application in a health care community. *IFAC-PapersOnLine*, 52(13):2596–2601, 01 2019.

[4] S Bansal and R Goel. Multi objective vehicle routing problem: A survey. *Asian Journal of Computer Science and Technology*, 7(3):1–6, 2018.

[5] K Braekers, K Ramaekers, and N. I. Van. The vehicle routing problem: State of the art classification and review. *Computers and Industrial Engineering*, 99:300–313, December 2016.

[6] Merve Cengiz Toklu. A fuzzy multi-criteria approach based on clarke and wright savings algorithm for vehicle routing problem in humanitarian aid distribution. *Journal of Intelligent Manufacturing*, 34(5):2241–2261, 2023.

[7] Sofie Coene, A Arnout, and Frits CR Spieksma. On a periodic vehicle routing problem. *Journal of the Operational Research Society*, 61(12):1719–1728, 2010.

[8] Przemyslaw Czuba and Dariusz Pierzchala. Machine learning methods for solving vehicle routing problems. In *International Business Information Management Association (IBIMA)*, 36, pages 12990–12996, November (2020).

[9] Hajem Daham and Husam Jasim Mohammed. An evolutionary algorithm approach for vehicle routing problems with backhauls. *Materials Today: Proceedings*, 2021.

[10] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.

[11] Qiu Dawei, Wang Yi, Hua Weiqi, and Strbac Goran. Reinforcement learning for electric vehicle applications in power systems:a critical review. *Renewable and Sustainable Energy Reviews*, 173(32):113052, 2023.

[12] Kenneth De Jong. Learning with genetic algorithms: An overview. *Machine learning*, 3:121–138, 1988.

[13] Guy Desaulniers, Jacques Desrosiers, Andreas Erdmann, Marius M Solomon, and François Soumis. Vrp with pickup and delivery. *The vehicle routing problem*, 9:225–242, 2002.

[14] Martin Desrochers, Jacques Desrosiers, and Marius Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations research*, 40(2):342–354, 1992.

[15] A. M. Djebbar. *Une approche heuristique pour l'ordonnancement du transport*. PhD thesis, University of Science and Technology of Oran Mohamed Boudiaf, Oran, Algeria, (2021).

[16] Lu Duan, Yang Zhan, Haoyuan Hu, Yu Gong, Jiangwen Wei, Xiaodong Zhang, and Yinghui Xu. Efficiently solving the practical vehicle routing problem: A novel joint learning approach. In *26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3054–3063, 2020.

[17] Simsir Fuat and Ekmekci Dursun. A metaheuristic solution approach to capacitied vehicle routing and network optimization. *Engineering Science and Technology, an International Journal*, 22(3):727–735, 2019.

[18] T. Gayatri, G. Srinivasu, D. Krishna Chaitanya, and V.K. Sharma. A review on optimization techniques of antennas using ai and ml / dl algorithms. *International Journal of Advances in Microwave Technology*, 07:288–295, 01 2022.

[19] A. Gerboudj. *Méthode de résolution de problèmes difficiles académiques*. PhD thesis, Constantine 2 University, Constantine, Algeria, 2013.

[20] Joaquim Gromicho, Jelke J van Hoorn, Adrianus Leendert Kok, and Johannes MJ Schutten. Restricted dynamic programming: a flexible framework for solving realistic vrps. *Computers & operations research*, 39(5):902–909, 2012.

[21] Lahcene Guezouli. *Développement d'une approche hybride multi-critères basée clustering pour la résolution d'un problème de distribution de produits*. PhD thesis, Mustapha Ben Boulaid Batna 2 university, Batna, Algeria, (2019).

[22] Singh. Gurpreet and Dhir Vijay. Open vehicle routing problem by ant colony optimization. *International Journal of Advanced Computer Science and Applications*, 5(3):63–68, 2014.

[23] Mais Haj Rachid. *Les problèmes de tournées de véhicules en planification industrielle : classification et comparaison d'opérateurs évolutionnaires*. PhD thesis, Franche Comté University, Besançon, FRANCE, 2010.

[24] Ming Han and Yabin Wang. A survey for vehicle routing problems and its derivatives. In *IOP conference series: materials science and engineering*, volume 452, page 042024. IOP Publishing, (2018).

[25] Jiang Hao, Lu Mengxin, Tian Ye, Qiu Jianfeng, and Zhang Xingyi. An evolutionary algorithm for solving capacitated vehicle routing problems by using local information. *Applied Soft Computing*, 117:108431, 2022.

[26] JH Holland. Adaptation in natural and artificial systems. ann arbor: University of michigan press. *Ann Arbor: The University of Michigan Press*, 1975.

[27] Abdullahi Ibrahim, Rabiat Abdulaziz, Jeremiah Ishaya, and Samuel Sowole. Vehicle routing problem with exact methods. *IOSR Journal of Mathematics*, 15(3):05–15, May - June 2016.

[28] Zangir Iklassov, Ikboljon Sobirov, Ruben Solozabal, and Martin Tak?? Reinforcement learning approach to stochastic vehicle routing problem with correlated demands. *IEEE Access*, 11:87958–87969, 2023.

[29] Tobias Jacobs, Francesco Alesiani, and Gulcin Ermis. Reinforcement learning for route optimization with robustness guarantees. pages 2592–2598, (2021).

[30] Brian Kallehauge, Jesper Larsen, Oli BG Madsen, and Marius M Solomon. *Vehicle routing problem with time windows*. Springer, 2005.

[31] Bariş Keçeci, Fulya Altıparmak, and İmdat Kara. A mathematical formulation and heuristic approach for the heterogeneous fixed fleet vehicle routing problem with simultaneous pickup and delivery. *Journal of industrial and management optimization*, 17(3):1069–1100, 2021.

[32] Wouter Kool, Herke van Hoof, Joaquim Gromicho, and Max Welling. Deep policy dynamic programming for vehicle routing problems. In *International conference on integration of constraint programming, artificial intelligence, and operations research*, pages 190–213. Springer, 2022.

[33] Ela Kumar. *Artificial intelligence*. IK International Pvt Ltd, (2013).

[34] Edward Lam, Guy Desaulniers, and Peter J Stuckey. Branch-and-cut-and-price for the electric vehicle routing problem with time windows, piecewise-linear recharging and capacitated recharging stations. *Computers & Operations Research*, 145:105870, 2022.

[35] Ailsa H Land and Alison G Doig. *An automatic method for solving discrete programming problems*. Springer, 2010.

[36] Gilbert Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, 1992.

[37] Allan Larsen. *The Dynamic Vehicle Routing Problem*. PhD thesis, Technical University of Denmark (DTU), Kgs. Lyngby, Denmark, (2001).

[38] Guoming Li and Junhua Li. An improved tabu search algorithm for the stochastic vehicle routing problem with soft time windows. *IEEE Access*, 8:158115–158124, 2020.

[39] Jane Lin, Wei Zhou, and Ouri Wolfson. Electric vehicle routing problem. *Transportation Research Procedia*, 12:508–521, June 2016.

[40] Reza Moghdani, Khodakaram Salimifard, Emrah Demir, and Abdelkader Benyettou. The green vehicle routing problem: A systematic literature review. *Journal of Cleaner Production*, 279:123691, 2021.

[41] Samuel Nucamendi-Guillén, Diego Flores-Diaz, Elias Olivares-Benitez, and Abraham Mendoza. A memetic algorithm for the cumulative capacitated vehicle routing problem including priority indexes. *Applied Sciences*, 10(11), 2020.

[42] P Oberlin, S Rathinam, and S Darbha. Today's traveling salesman problem. *Robotics & Automation Magazine*, 17:70–77, 01 2011.

[43] Bo Peng, Lifan Wu, Yuxin Yi, and Xiding Chen. Solving the multi-depot green vehicle routing problem by a hybrid evolutionary algorithm. *Sustainability*, 12(5), 2020.

[44] Citra Dewi Purnamasari and Amelia Santoso. Vehicle routing problem (vrp) for courier service: A review. In *MATEC Web of Conferences*, 204, pages 1–9, (2018).

[45] Arne Schulz. Using infeasible path cuts to solve electric vehicle routing problems with realistic charging functions exactly within a branch-and-cut framework. *EURO Journal on Transportation and Logistics*, page 100131, 2024.

[46] Satyendra Kumar Sharma, Srikanta Routroy, and Utkarsh Yadav. Vehicle routing problem: recent literature review of its variants. *International Journal of Operational Research*, 33(1):1–31, Augst 2018.

[47] Vishnu Pratap Singh, Kirti Sharma, and Debjani Chakraborty. A branch-and-bound-based solution method for solving vehicle routing problem with fuzzy stochastic demands. *Sādhanā*, 46:1–17, 2021.

[48] Marcone Souza, Marcio Mine, Matheus Silva, Luiz Ochi, and Anand Subramanian. A hybrid heuristic, based on iterated local search and genius, for the vehicle routing problem with simultaneous pickup and delivery. *International Journal of Logistics Systems and Management*, 10(12):142–157, 2011.

[49] S. Y. Tan and W. C. Yeh. The vehicle routing problem: State-of-the-art classification and review. *Applied Sciences*, 11(21):0–28, 2021.

[50] Franziska Theurich, Andreas Fischer, and Guntram Scheithauer. A branch-and-bound approach for a vehicle routing problem with customer costs. *EURO Journal on Computational Optimization*, 9:100003, 2021.

[51] F Yu Vincent, Hadi Susanto, Panca Jodiawan, Tsai-Wei Ho, Shih-Wei Lin, and Yu-Tsung Huang. A simulated annealing algorithm for the vehicle routing problem with parcel lockers. *IEEE Access*, 10:20764–20782, 2022.

[52] Li Wang, Yifan Ding, Zhiyuan Chen, Zhiyuan Su, and Yufeng Zhuang. Heuristic algorithms for heterogeneous and multi-trip electric vehicle routing problem with pickup and delivery. *World Electric Vehicle Journal*, 15(2):69, 2024.

[53] Liang Xin, Wen Song, Zhiguang Cao, and Jie Zhang. Step-wise deep learning models for solving routing problems. *IEEE Transactions on Industrial Informatics*, 17:4861–4871, 2020.

[54] Yunqiu Xu, Meng Fang, Ling Chen, Gangyan Xu, Yali Du, and Chengqi Zhang. Reinforcement learning with multiple relational attention for solving vehicle routing problems. *IEEE Transactions on Cybernetics*, 52(10):11107–11120, 2022.

[55] Niu Yunyun, Shao Jie, Xiao Jianhua, Song Wen, and Cao Zhiguang. Multi-objective evolutionary algorithm based on rbf network for solving the stochastic vehicle routing problem. *Information Sciences*, 609:387–410, 2022.

[56] Lotfi Asker Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[57] Zong Zefang, Tong Xia, Zheng Meng, and Li Yong. Reinforcement learning for solving multiple vehicle routing problem with time window. *ACM Transactions on Intelligent Systems and Technology*, 15(32):119, 2024.

[58] A. Zinfiou. *Algorithmes évolutionnaires pour l'ordonnancement industriel : application à l'industrie automobile.* PhD thesis, Québec University, Québec, Canada, (2008).

Detection of payload injection attacks using a transformer based model

Abdelmalek Djezar
djezar_abdelmalek.fs@univ-boumerdes.dz

Amine T. Guellab
t.guellab.fs@univ-boumerdes.dz

Abdelhak Mesbah
abdelhak.mesbah@univ-boumerdes.dz

M'hamed Bougara University Boumerdes
Faculty of Science

Abstract

As the popularity of web apps continues to increase, surpassing traditional desktop and mobile applications, the threat of web-based assaults is becoming more prevalent and poses a significant risk to both individuals and organizations. According to the OWASP Top 10 project, injection-type attacks are ranked third on the list, including the likes of Cross Site Scripting (XSS) and SQL injection (SQLi). While conventional signature-based Web Application Firewalls offer protection against known attacks, they falter against zero-day attacks. This work proposes the use of a Transformer-based deep learning model to detect injection-type attacks. Using the HttpParams Dataset, our model achieves an accuracy of 99.81% with an average classification compute time of 10ms.

Keywords— Deep learning ,Transformer, Web Application Firewall, Binary Classification, HttpParams

1 Introduction

In recent times, web applications have begun to replace conventional desktop applications, enabling software to be utilized on multiple platforms via the browser. However, this pattern introduces new security vulnerabilities for both individuals and organizations, namely in the form of web-based attacks. These attacks often consist of the introduction of harmful code payloads, such as SQLi and XSS, which are classified as the third most common risks in the OWASP Top 10 Project[4].Web Application Firewall (WAF) are deployed to safeguard online applications from many types of web-based attacks, including payload injection. Generally, an approach that relies on signatures is employed to identify and detect attacks. Nevertheless, this approach proves inadequate when it comes to countering zero-day assaults, as it necessitates periodic updates to address emerging threats[1].

In this work, we explore the utilization of a deep learning model based on the Transformer architecture [9], with a Gradient-Based Sub-word Tokenizer (GBST) [8]. This tokenizer enables the model to operate at the character level, allowing for the detection of payload injection attacks. Architectures designed for Natural Language Processing (NLP) approaches, like the Transformer, are expected to achieve high accuracy in categorizing typical user input against payload injections due to their human-readable structure. We deliberately selected it over other well-known deep learning architectures focused on natural language processing, such as Recurrent Neural Network (RNN) incorporating Long Short-Term Memory (LSTM) and gGated Recurrent Unit (GRU), due to its capability to process input in parallel rather than sequentially[6].

In [10] Shihao et al. develop a WAF using a deep learning model trained on a dataset of of malicious URLs pre-processed into one hot encoding making them suitable for a Convolutional Neural Network (CNN), the model itself is structured with convolutional layers for feature extraction, pooling layers to filter these features, and fully connected layers to integrate these features and classify input data achieving high accuracy in distinguishing between benign and malicious web requests.

In [2] an anomaly-based web application firewall model was developed using NLP techniques and a linear support vector machine (SVM) algorithm. The study focused on comparing word n-gram and character n-gram methods for feature extraction, finding that character n-grams significantly enhanced detection performance.

In [7], Aref et al. employ a machine learning technique to categorize malicious payloads. They constructed a parsing unit to extract features like input length, the ratio of alphanumeric characters to special characters, attack weight—which is determined by adding together four additional sub-features discussed in detail within the article. They then apply Naive Bayes for classification, yielding an accuracy of 97.61% on the HttpParams dataset[3], also Mohamed and al. in [5] after experiment with different approaches achieve their highest accuracy of 99.66% with multi-class classification using a Bidirectional long short-term memory (Bi-LSTM) based model on HttpParams, Conversely our model yields an accuracy of 99.81% on the same dataset.

The remaining sections of the paper are organized in the following manner: Section 2 provides an overview of the structure of our model, while section 3 showcases the outcomes of our experiment. Our conclusion and future areas of focus are discussed in part 4.

2 Our Approach

Our approach consists of 4 steps, encoding the input, Tokenization using the GBST, extracting contextual information from the tokens using the Transformer encoder, finally classifying the results using Multilayer Perceptron (MLP) into a binary output with a sigmoid function.

Tokenization is the first stage in the process of using NLP approaches and the Transformer architecture. It involves converting plain text into embedding vectors that may be utilized by the

Transformer. We choose to employ a GBST for this undertaking, as it necessitates no data preparation. It accomplishes this by utilizing the characters directly to produce a soft sub-word sequence that is down-sampled beforehand being inputted into the Transformer.

As we can see in Figure 1, each character in the plain text input is initially encoded into its UTF-8 representation. Additionally, padding is added and an attention mask is generated where the false values represent the padding indexes, this ensures that no attention is given to the padding. To ensure that the entire dataset is covered, we have chosen a maximum length of 2048 characters, which is also the maximum size for a GET Hyper Text Transfer Protocol (HTTP) parameter, considering that the longest value in our dataset is 1058 characters. After that, the GBST receives this and uses it to create a soft sub-word sequence with the proper padding mask, which is then supplied to the Transformer encoder, note that usually positional encoding is first applied to the input embedding before being fed to the encoder but in our case using the GBST removes that need. The Transformer encoder uses multiple layers, each with a multi-head self-attention mechanism that processes the input sequence in parallel, allowing it to attend to different parts of the sequence simultaneously. The resulting output is a sequence of vectors where each vector corresponds to an input token, enriched with contextual information gathered from all other tokens in the sequence, this is then compressed using mean pooling into a 1 dimensional vector and then fed into two linear layers with a dropout layer in between. Ultimately, a sigmoid activation function is employed at the output of the final layer to facilitate binary classification.

The input parameter for the Neural Network (NN) is displayed in Table 1, while the input parameters for the Transformer are shown in Table 2. The selection of these parameters was based on their potential impact on both computational time and classification performance.

Table 3 represents the GBST parameters , we have selected a maximum block size of 4 and a down-sampling factor of 4. As a consequence, the transformer now receives inputs of length 512 instead of 2048. This adjustment is important for lowering computing time.

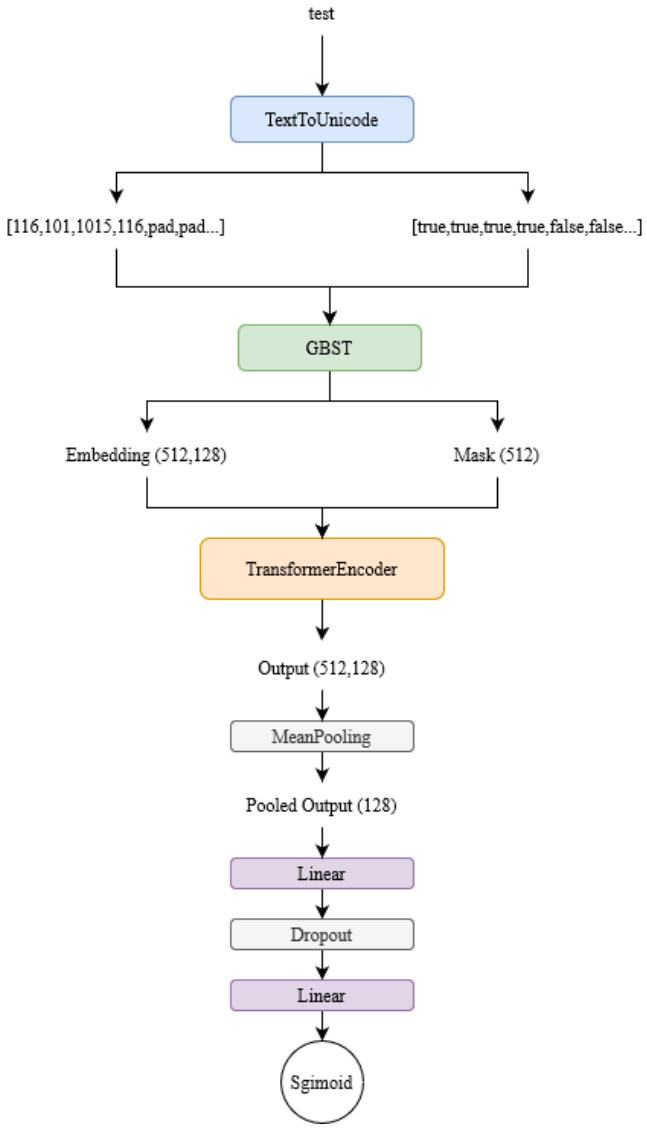


Figure 1: Model Architecture

3 Results

The model is trained using the HttpParams Dataset, which consists of 19,304 benign values and 11,763 malicious payloads. The malicious payloads include 10,852 instances of SQLi, 532 instances of XSS, 89 instances of command injections, and 290 instances of path traversal attacks. Seventy percent, twenty percent, and ten percent of the dataset were assigned for training, validation, and testing, respectively.

Upon completing 30 epochs of training, we observe the outcomes depicted in Figure 3 and Figure 4. The model achieved an accuracy of 99.91% and an F1 score of 99.88% on the training data. It also achieved an accuracy of 99.81% and an F1 score of 99.71% on the test data. The precision on the training data was 99.77%, while on the test data it was 99.50%.

In figure 2 our results are compared with the other works mentioned before. The confusion matrix displayed in Table 4 presents the outcomes obtained from the test data. The model achieved an average classification time of 10 ms using an NVIDIA P100 GPU.

Table 4: Confusion matrix

		Actual	
		norm	**anom**
Predicted	**norm**	1900	0
	anom	6	1200

Table 1: Hyper Parameters

Optimizer	Adam
Loss	Binary cross entropy
Learn rate	1e-4
Epochs	30
Activation Function	Sigmoid

h

Table 2: Transformer Parameters

Model Dimension	128
Number of Heads	2
Feed Forward Dimension	256

h

Table 3: GBST Parameters

Number of Tokens	Unicode 2.1
Dimension	128
Down Sample Factor	4
Max Block Size	4

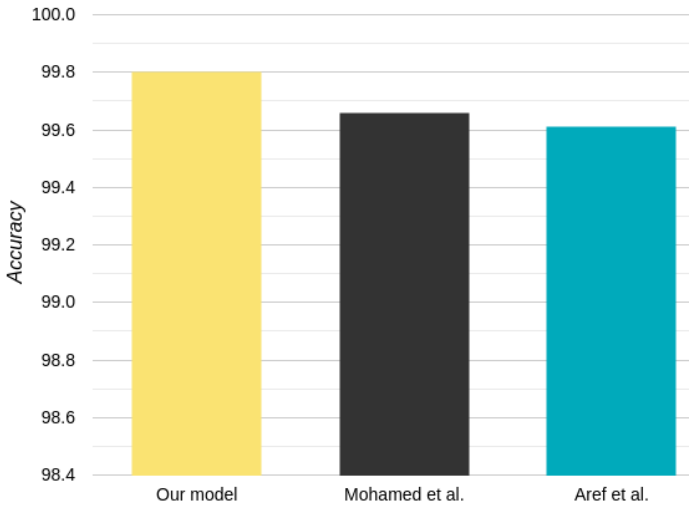


Figure 2: Accuracy comparison

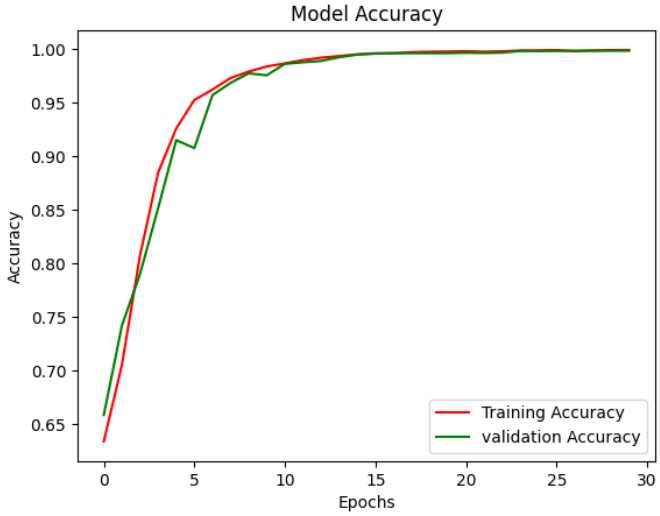


Figure 3: Model accuracy curve

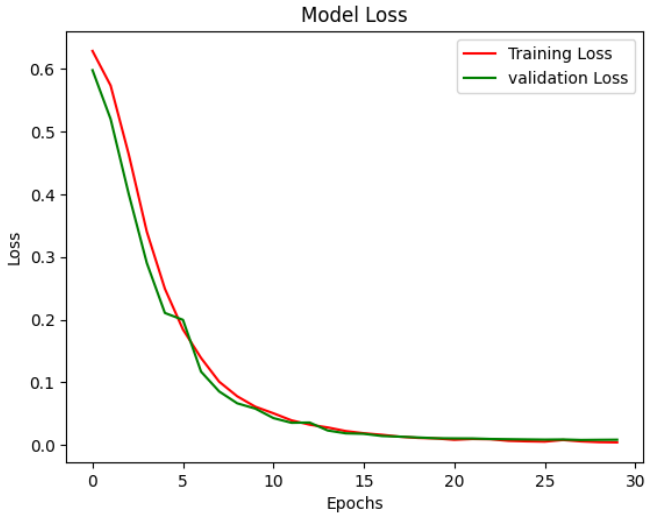


Figure 4: Model loss curve

4 Conclusion

The results of our study show that deep learning models based on Transformers are effective in detecting web-based injection attacks. By using the HttpParams dataset, our model achieves a remarkably high accuracy and F1 score. This emphasizes the potential of NLP-oriented deep learning models in improving web application security against evolving attacks. Potential areas for future areas of focus could involve exploring and other datasets, enhancing the model to minimize computational time, investigating its scalability, and examining its compatibility with established security frameworks.

References

[1] S. Applebaum, T. Gaber, and A. Ahmed. Signature-based and machine-learning-based web application firewalls: A short survey. *Procedia Computer Science*, 189:359–367, 2021. AI in Computational Linguistics.

[2] B. IŞiker and SoĞukpinar. Machine learning based web application firewall. In *2021 2nd International Informatics and Software Engineering Conference (IISEC)*, pages 1–6, 2021.

[3] Morzeux. Httpparamsdataset, 2016. https://github.com/Morzeux/HttpParamsDataset (Accessed on April 22, 2024).

[4] OWASP. Owasp top ten, 2021. https://owasp.org/www-project-top-ten (Accessed on April 22, 2024).

[5] M. R. Sara Mohamed. Multi-class intrusion detection system using deep learning. *Journal of Al-Azhar University Engineering Sector*, 18(69):869–883, 2023.

[6] R. M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview, 2019.

[7] A. Shaheed and M. H. D. B. Kurdy. Web application firewall using machine learning and features engineering. *Security and Communication Networks*, 2022:5280158, Jun 2022.

[8] Y. Tay, V. Q. Tran, S. Ruder, J. Gupta, H. W. Chung, D. Bahri, Z. Qin, S. Baumgartner, C. Yu, and D. Metzler. Charformer: Fast character transformers via gradient-based subword tokenization, 2022.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.

[10] S. Wang, R. Liu, X. Guo, and G. Wei. Design of web application firewall system through convolutional neural network and deep learning. In *2022 International Conference on Computers, Information Processing and Advanced Education (CIPAE)*, pages 454–457, 2022.