Smart Technologies Academic Press

STAP Journal of Security Risk Management

https://www.jsrm.thestap.com/

PhishGuard: AI-Driven Graph-Based Analysis for Smarter Email Security

Harchana Ramesh[1], Noris Ismail[1], Nor Azlina Abd Rahman[1], Aitizaz Ali[1*]

[1] *Faculty of Computing, Engineering and Technology at Asia Pacific University of Technology and Innovation (APU), Malaysia*

ARTICLE INFO

Article History

Received: 07-10-2025

Vol.2026, No.1

DOI:

Corresponding author. Email:
aitizaz.ali@apu.edu.my

Orcid:
https://orcid.org/0000-0002-4853-5093

ABSTRACT

This research presents a phishing detection system that integrates graph analytics and machine learning to improve email security. As phishing tactics become more sophisticated, traditional filters often fail to detect such threats effectively. This project proposes a dual-model solution: a RoBERTa-based transformer is used to classify the email body content, while a Neo4j-powered graph model analyses sender-receiver domain relationships using graph metrics such as PageRank, ArticleRank, and Degree Centrality. The rule-based system intelligently combines the predictions of the two models. Highly confident RoBERTa results are accepte d directly, whereas for the remaining cases, scores from the graph model are applied. For mid-confidence cases, a fixed rule-based thresholding logic is used to ensure robust classification. For real-time detection, a web interface was developed using Streamlit, integrating Gmail API and Google Apps Script for email quarantine. The system achieved an F1 score above 0.99 in testing, marking it as a fully stable system for spam identification. By combining content and relational signals, the work advances email security and accordingly fulfils Sustainable Development Goal 9 by fostering innovation infrastructure in digital safety.

Keywords: Phishing Detection, RoBERTa, Graph Analytics, Neo4j, Text Classification, SDG 9

How to cite the article

1. Introduction

Phishing has significantly evolved since the 1990s, particularly due to advancements in technology and the increasing complexity of cyberattacks. Although the term “phishing” is not easily defined, it can be described as a type of cyber fraud that exploits social relationships among individuals, using malicious software and deceptive techniques to mislead unsuspecting users and gain unauthorized access to their personal information (Hewage, Khan, Nawaf, & Alkhalil, 2021). Besides, this type of threat usually involves impersonating trusted companies and tricking people into providing personal information such as account credentials and credit PINs. Practical phishing attacks can occur in various forms, including emails, websites, and telephone calls, often involving social engineering tactics to manipulate the victim's natural trust.

Phishing attacks can take various forms, including emails, websites, and phone calls, and frequently involve social engineering tactics to manipulate the victim’s natural sense of trust. There are many ways an unethical threat actor can compromise an organization, with emails and social or professional networks being among the most vulnerable. The widespread use of information technology has further exacerbated this issue. Email phishing, the oldest form of phishing, has existed since the early days of the internet, though it initially involved much simpler techniques (Nadeem, et al., 2023). As technologies such as artificial intelligence and automation continue to evolve, phishing attacks have not only become easier to execute but also increasingly difficult to detect and prevent (Putra, Ubaidi, Zulfikri, Arifin, & Ilhamsyah, 2024). A significant number of internet users are still oblivious to the risks associated with phishing. Their lack of adequate training to spot the red flags makes them more vulnerable to these advanced attacks. As a result, phishing has become a crucial focal point in the field of cybersecurity, propelling initiatives aimed at recognizing current attack patterns and creating effective preventive strategies to protect both individuals and organizations from their detrimental effects.

Due to the above changes, it can be safely known from the above developments; phishing continues to be a potential threat in the cybersecurity era which is changeable and depends on the available security measures. As these threats continue to advance with advancements in technology and social engineering tactics. This research which is the focus of this study, “PhishGuard: AI-Driven Graph Based Analytics for Smarter Email Security”, seeks to rise to the occasion by incorporating modern techniques that have AI in it and are graph-based. The project aims to go a step further by considering the complicated relational patterns observed in email communications to further enhance the chances of detection and improve email security operations. The line of this study is to upgrade the extent of defense systems of phishing attack without endangering the web users which include both individuals and organizations.

PhishGuard is designed with simplicity and accessibility in mind. The system features a streamlined Streamlit web interface, allowing users to monitor incoming emails and view detection results in real time. The system has foregone any manual operations possible by utilizing the Gmail API and Google Apps Script for email scanning. In other words, the user does not perform any other tasks other than having to fill in or correct the information provided. PhishGuard automatically segregates any phishing or suspicious emails into a spam folder, providing a familiar experience as per common commercial email clients. There is no need to first train any machine learning technique or theory to operate the system. Besides, the whole deployment could be made available to an end user with one click on the cloud or onto a local machine. With its clean and intuitive UI, PhishGuard limits user configuration to only essential actions, ensuring ease of use. All in all, PhishGuard endeavors to uphold the accuracy and robustness of phishing detection while catering to non-technical stakeholders, such as small business employees or the layman email user.

The main contribution of this work is the development of a two-tiered phishing detection system that combines content-based analysis with graph-based behavioral analysis. The first-tier detection layer uses a fine-tuned RoBERTa transformer model to classify the email body as either spam or ham based on its content. The second layer performs behavioral analysis by modeling sender and receiver domains in Neo4j and applying graph algorithms such as PageRank, ArticleRank, and Degree Centrality.
A set of rules combines the output of these two independent models. When the confidence score returned by the RoBERTa model is very high (≥ 0.9), that classification is accepted as is. Otherwise, the graph-based method's classification is used. For the cases falling in-between, a weighted scoring approach is combined.

This system is supported by a prediction engine on the backend, while an easy-to-use frontend is built using Streamlit and integrated with Gmail via Google Apps Script. Incoming emails are scanned in near real-time and quarantined if flagged as suspicious. The dataset used in this system consists of labeled phishing and ham emails, along with sender-receiver metadata. The proposed model achieved a test F1 score greater than 0.99 across all test sets and significantly reduced false negatives compared to traditional methods. While model performance is a key focus, this study also emphasizes system deployment, real-world email processing, and the broader implications for secure digital infrastructure.

2. Literature Review

2.1. How do Emails Work?

Email is a fundamental component of modern communication. Email messages are transferred from the sender's email client to the recipient's email server using the Simple Mail Transfer Protocol (SMTP). SMTP establishes a reliable connection for data transfer through the Transmission Control Protocol (TCP) (SiteGround, 2024). Protocols such as Internet Message Access Protocol (IMAP) and Post Office Protocol version 3 (POP3) are used to retrieve emails once they reach the recipient's server. IMAP synchronizes emails across multiple devices while maintaining copies on the server, whereas POP3 transfers emails from the server to the client's device, often deleting them from the server afterward.

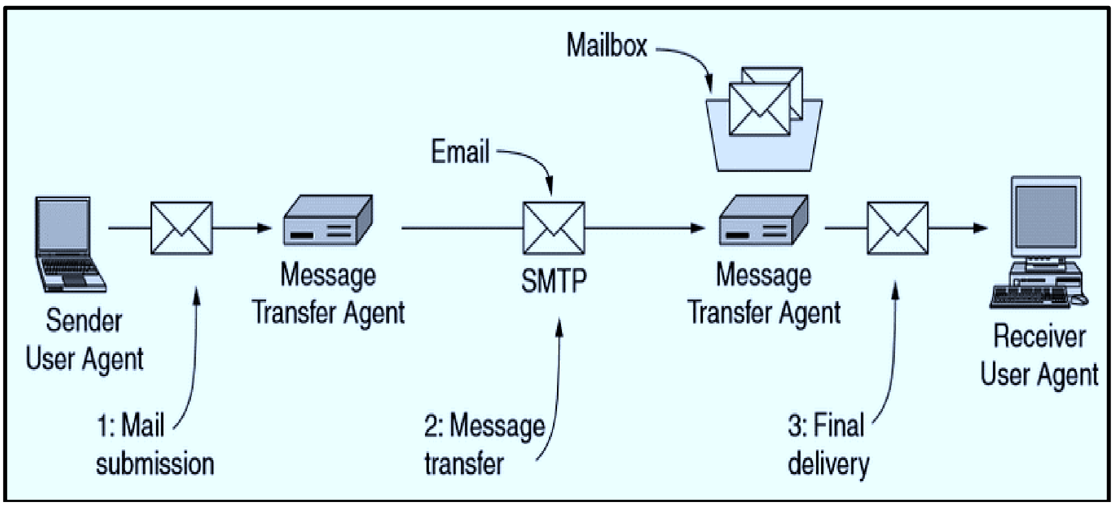

Figure 1. Email Architecture (Booker, 2024)

POP3 is appropriate for users who would rather keep emails locally and view them offline. As this usually downloads messages into a single device (Aleksic, 2022). However, because messages might not be synchronized between platforms, this method might make it harder to retrieve emails from numerous devices. By retaining emails on the server, IMAP, on the other hand, gives users more flexibility and permits them to view and manage their inbox from several devices while maintaining constant synchronization. To ensure smooth communication and configure email clients efficiently, it is necessary to comprehend these protocols.

2.2. Mechanism of Phishing Attacks

Phishing attacks are carried out by following a set of intentional steps meant to trick people into disclosing private information. The attacker usually starts the process by creating a persuasive message that seems to be from a reliable source, such a bank or respectable company. After that, this message is shared with a large audience or targeted via email, social media, or other avenues of communication. To compel quick action, the text frequently uses urgent language or alluring offers (Alkhalil, Hewage, Nawaf, & Khan, 2021). The victim is urged to download an attachment or click on a link in the message, which opens a malicious website or starts the installation of malware.

The counterfeit website mirrors a real one and takes in the victim's financial information, login passwords, or other personal information. The information that the victim enters is sent straight to the attacker, who can use it for identity theft, financial gain, or illegal system access. This systematic approach emphasizes how crucial user awareness and strong security measures are to identifying and stopping phishing attempts (Hewage, Khan, Nawaf, & Alkhalil, 2021).

2.3 History of Phishing

With advancement of technology, phishing attacks have equally evolved over the years. Starting off with basic deceptive emails to highly targeted method. As is known, earlier phishing attempts often involved generic messages which are sent to a large group of people, aiming to deceive users into revealing their personal information. However, as more awareness is created and with the defenses improving, the attackers have equally adapted to developing a more refined approach into tricking users.

There are many types of phishing attacks. Spear-phishing is a targeted approach where attackers will personalize the emails by having personal information about the victim. This method tends to lure the victim into trusting the emails before engaging with them. Meanwhile, whaling has another method where the victim is first targeted. Whaling is usually focused on high-profile individuals to exploit their access into giving sensitive information. Clone-phishing involves duplication legitimate mails, but links or attachments are attached or replaced with the original content, these links and attachments tend to have malicious context within.

These advanced phishing attempts are becoming more and more common, according to recent studies. According to a global poll of IT security experts conducted in 2021, 51% of the attacks accumulate to phishing (Barracuda, 2022), and within industries 13% reports more than 50 such occurrences (Petrosyan, 2023). Additionally, businesses have increased their cybersecurity expenses because of the surge in ransomware and social engineering attacks (Colback, 2024). These figures highlight the necessity for advanced detection and prevention techniques by highlighting how easily attackers can get beyond conventional security measures.

2.4 Introduction to Machine Learning in Cybersecurity

Machine learning (ML) is part of artificial intelligence (AI), that enables computers to learn their tasks and make necessary decisions based on data, without the need for programming to be done for each specific task (IBM, 2024). Machine learning usually trains algorithms on a dataset. When training, this enables the algorithm to recognize patterns within the dataset allowing them to make predictions and improve their performance over time.

With its advanced methods for identifying, averting, and responding to cyber threats, machine learning (ML) is among the important component within the cybersecurity industry. Traditional cybersecurity systems often rely on predefined signatures and rules. ML addresses this limitation by learning from historical data to detect new and unknown threats. For instance, supervised learning algorithms can classify network traffic as benign or malicious based on labelled datasets, while unsupervised learning can identify unusual patterns without prior knowledge of threats (Chugani, 2024). ML has also been a huge help by detecting high rate of false positives within a security team. It can analyze contextual information, and it can correlate pieces of information to reduce the amount of fault positive (Ballejos, 2024). Within the cybersecurity industry, ML has been a huge favor by improving accuracy over time.

Despite its advantages, integrating machine learning into cybersecurity is hard. By using adversarial tactics, adversaries can exploit machine learning models with false input. Furthermore, accurate, high-quality data is essential to machine learning's efficacy; without it, predictions may not hold up. Therefore, maintaining ML reliability requires constant monitoring and frequent updates. ML is essential since it greatly improves cybersecurity

2.5 Comparative Analysis of Machine Learning Algorithms

Type of ML	Description	Applications	Suitability for Phishing Detection
Supervised Learning	Trains of labelled datasets where they input are paired with outputs, allowing mapping between them. Effective for classification tasks.	Spam filtering, fraud detection, image recognition	It is best suited for phishing detection as it can benefit from labelled data. Email dataset classified as “phishing” or “legitimate”, helps it to be precise.

Unsupervised Learning	Trains on unlabeled data, identifying hidden patterns and structures within data.	Market segmentation, anomaly detection	Not ideal as it deals with non-labelled data. This would be less precise where classes are essential (phishing, legitimate).
Semi-Supervised Learning	Combines small, labelled datasets and a larger set of unlabeled datasets, leveraging both for accuracy.	Image and document classification	It might work but depending on the dataset. If large unlabeled data is available or may not match.
Reinforcement Learning	Train an agent by awarding and penalizing in an environment. Primarily used for goal-oriented tasks.	Robotics, gaming	Less suited for email classification, as RL is goal-oriented and does not fit well for classification tasks or pattern recognition.

2.5 Comparative Between Machine Learning Algorithms

Algorithm	**Characteristic**	**Strengths**	**Weaknesses**
K-Nearest Neighbors (KNN) (Found in the year 1967)	Classifies based on similarity to 'k' nearest neighbors using distance metrics.	Simple, effective within a smaller dataset.	Computationally intensive with large dataset. Sensitive to irrelevant features.
Support Vector Machine (SVM) (Found in the year 1992)	Finds an optimal hyperplane for maximum separation between classes, effective in high-dimensional space.	High accuracy, for binary classification and linearly separable data.	Sensitive to outliers and requires careful parameter.
Decision Tree (Found in the year 1963)	Recursively split data based on feature values to form a tree structure.	Easy to interpret, able to handle categorical and numerical data.	Prone to overfitting and become overly complex before pruning.

Logistic Regression (Found in the year 1958)	Statistical method for binary classification, predicts probability of a class using a logistic function.	Useful for binary classification. Works well with linear separability.	Limited to binary or multiclass. Can underperform with complex data.
Random Forest (Found in the year 1995)	Ensemble method that builds multiple decision trees and aggregates their predictions.	Reduces overfitting, handles high-dimensional data. Effective feature selection. Suitable for parallel processing.	May be less interpretable, can require significant computational resources for large dataset.

2.6 Traditional Methods

As is known, phishing emails has been an issue over the years ever since its first emergence back in the 1990s. According to the study, "Advancing Phishing Email Detection: A Comparative Study of Deep Learning Models" published in 2024, which explored 1D-CNN based models for phishing detection. It extended the baseline model with recurrence layers like LSTM and GRU and has attained very high detection accuracy and low false positives by using hybrid architectures along with advanced feature engineering (Altwaijry, Al-Turaiki, Alotaibi, & Alakeel, 2024). This paper demonstrates the potential of deep learning models to accurately detect phishing attempts.

Another approach was using the Bidirectional LSTMs with word embedding to allow the model to learn contextual relationships in the e-mails. This improved the performance of detection by understanding email dependencies more effectively compared to unidirectional models (Wolert & Rawski, 2023). Figures for the research entitled "Enhancement of Phishing Detection Using Machine Learning: A Comparative Study of Classifiers and Conventions," issued in December 2023, were checked and addressed (Altwaijry , Al-Turaiki , Alotaibi, & Alakeel, 2024). On how to fight phishing the matter in question was dealt with through the employment of numerous machine learning classifiers and feature sets. Serving the purpose, 500 phishing websites and 500 not-phishing websites were selected from 25 different features such as URL, html and http header attributes. Also, classifiers using decision tree and attributes like URL and HTTP of the HTMLs gave a very high F1 score of 0.99 to be superior to other models in terms of efficiency and speed of analysis. This method is effective for Phishing, and they aid in using machine learning models for detection of Phishing attacks, but the major problem with such techniques is the fact that they consider characteristics already realized, hence not effective enough because they will not be of much help in dealing with more complicated and new Phishing attacks.

PhishGuard differs from the existing phishing detection systems because of its unique features and the novelty of the implementation method. PhishGuard has an upper edge by embedding graph database technology, Neo4j, to model relationships between email entities and analyze them for phishing pattern detection. This relational insight allows it to provide a bigger picture for phishing campaigns that no other current system may provide. First, PhishGuard adds an extra layer of protection, an email quarantine system. Such a feature has not been emphasized in any of the reviewed studies. This approach will help in isolating suspicious emails before they reach the user to prevent human error in case there are false negatives. Moreover, by deploying PhishGuard as a cloud-based service, it provides scalability and accessibility for organizations without the burden of deep cloud integration. Such a deployment strategy contrasts with many of the deployment strategies proposed in current research, which offer fewer practical cloud-hosted solutions.

In this regard, PhishGuard surpasses this framework with the use of graph database technology such as Neo4j in the relational analysis of email entities. The purpose of this is to utilize PhishGuard to analyze advanced phishing techniques which would otherwise be challenging for the established machine learning approaches. Furthermore, among other features, PhishGuard has an email quarantine where it takes matters into its own hands, and it isolates any emails which pose a risk to the email users from reaching them – an issue that has not been discussed in the study. Moreover, PhishGuard uses the cloud in its operation which ensures balancing the workload of the system and in addition offering good performance and management to any customer without having to possess cloud expertise, which is cool and very current. The traditional decision tree-based methods are effective when the features are fed to the system, multiple in case of duties i.e. first-generation users, however, when you look at the solutions that PhishGuard offers such as graph-based detection, proactive email quarantining and adaptability to the cloud, it is evident that the above are more potent against trends in phishing

2.7 Email Pipeline

The visualization depicts the structural breakdown of email domains into granular components that are critical for phishing detection. Each "Domain" node, representing a full email domain, is decomposed into smaller parts such as Subdomain, Secondary Subdomain, Primary Domain, Top-Level Domain (TLD), and Secondary TLD (STLD). The Primary Domain is totally related to both TLD and STLD, which defines the organizational or ownership structure of the addresses on the network (like waikato.ac.nz will reside within .ac.nz). On the other hand, there are Subdomain nodes that contain all the links leading to the previous elements (for example, list.scms) whereby; these could probably show some patterns which might be of some interest to the attackers. Additionally, "SENDS" within the domains is intended to model sender-receiver norms that are critical in ascertaining patterns of trust in electronic communication. The structuring of email searchable data in this manner will assist in seeking exclusive resemblances of emails, but the architecture of miscommunication and hidden patterns which will aid them is even more useful. Such decomposition opens the road towards following stages of feature extraction and graph-based models within the phishing detection system.

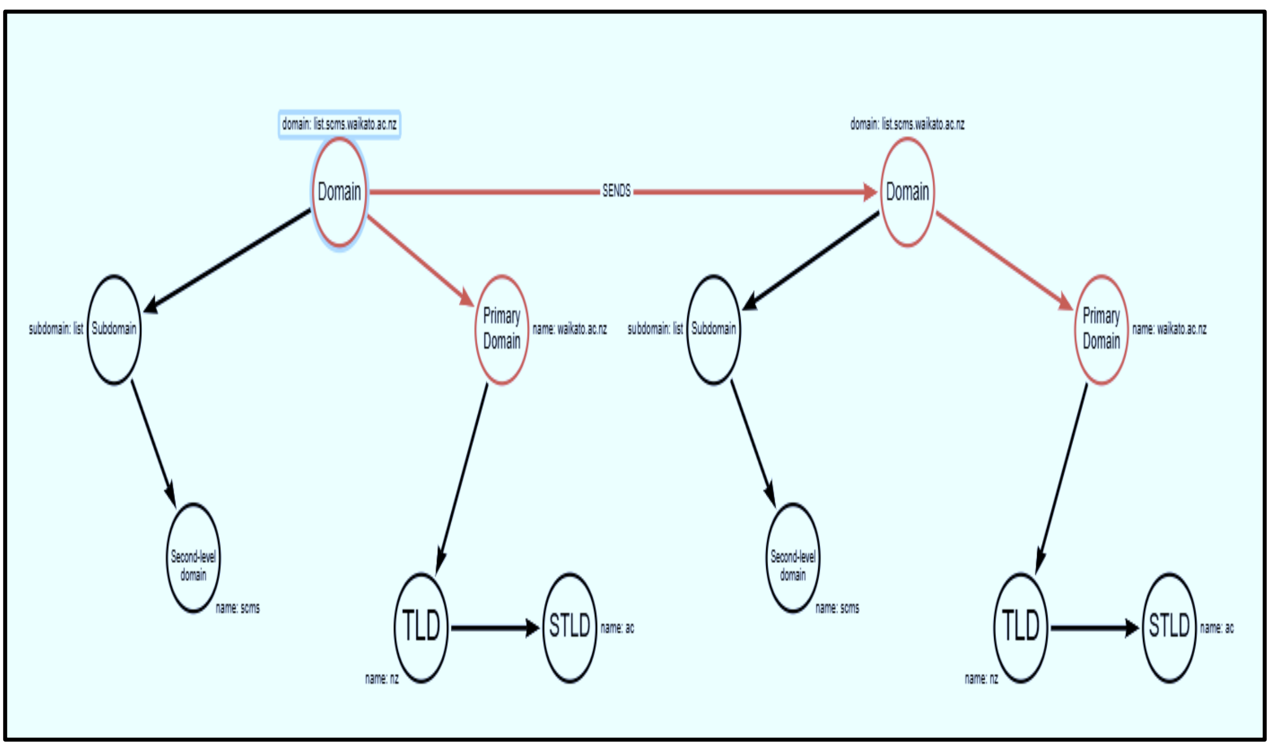

Figure 2. Email Pipeline

3. Methodology

A questionnaire is a data collection tool consisting of a series of questions designed to gather information from participants on specific topics. Questionnaires can include both open-ended and closed-ended questions. They are widely used in research to efficiently collect data from a large audience. They are particularly effective in gathering quantitative data that can be statistically analyzed and essential tool for cybersecurity research context of phishing detection research. A questionnaire allows researchers to gather insights on user awareness, experiences, and perceptions regarding phishing emails and security threats. Questionnaires enable researchers to understand common patterns in user behavior and experiences, such as the frequency with which they encounter phishing emails or their awareness of potential threats. By understanding the typical experiences of users, researchers can design phishing detection models that address actual user needs and vulnerabilities. Moreover, questionnaires are cheap to conduct a variable sample size. Data collection across different demographics might give the researchers an opportunity to check whether age, industry, or even levels of technical knowledge are relevant factors that relate an individual's vulnerability to phishing attacks. This is useful in cybersecurity research, as user feedback may uncover concerns and unconsidered vulnerabilities. User input enables the final solution will fit better into the actual needs and expectations.

3.1 Findings from User Survey on Phishing Awareness and Preferences

A sample of 50 people were surveyed using a questionnaire on phishing awareness in a bid to evaluate the extent of users' understanding, habits, and opinions on email security. Many thanks to the feedback provided by the participants in the form of their answers on how they perceive the existing and/or possible future threats of phishing and whether they have in place any remedial measures, as well as how such feedback can be utilised in constructing a phishing AI detection system. Most participants fell into the 18–24 age group and held academic backgrounds in cybersecurity, computer science, or related fields, indicating a tech-aware sample. A strong majority rated themselves as “familiar” or “very familiar” with phishing threats.

However, even this fact didn’t affect the results. While over 60% of people investigated the ad in the fear that it was fraud, a lot also confessed to not checking the email all the time. Almost two under thirds had firsthand contact with the phishing problem or have friends and family who have, which again provides evidence to how potent the danger is in electronic misuse. Most felt they could detect phishing attacks, but also many were not very sure, meaning training on the same issues as well as practical lessons were still needed.

The survey also explored safety rituals like use of Multi-Factor Authentication (MFA) along with Anti-Phishing programs. Knowledge on what MFA is, was quite high, but the protective software itself had varying usage levels. Only in 30 % of the cases participants were given phishing trainings or awareness and prevention updates, which shows that there is a disconnect in user training and continuous update of security.

In addition, survey participants’ level of awareness towards graph-based machine learning techniques has been assessed. Most participants had little prior knowledge in this area of interest. Despite this, most of the target population (over 60%) were in favour of applying relation-based features such as the sender-receiver domain behaviour for identifying fraudulent activities. Most respondents trusted machine learning’s effectiveness in spam filtering and recognized its potential to outperform traditional rule-based methods. This reflects a growing openness to adopting advanced AI and graph-based techniques in cybersecurity.

Finally, user preferences were clear: participants prioritized system accuracy, followed closely by performance and ease of use. Many indicated a willingness to sacrifice a complex interface in favour of higher detection accuracy. Open-ended feedback included suggestions for real-time phishing alerts, domain/IP tracing, ensemble learning, and even national blacklists for repeat phishing sources. Overall, the results underscore that users are ready for intelligent, scalable, and user-conscious security solutions like PhishGuard.

4. Development

4.1 Dataset

This development begins with the selection and preparation of the CEAS-08 email dataset, then advances to develop the Neo4j-graph-based model, then continues further to create and complete the architectural part of the system and then ends up by implementing machine-learning pipelines, API services, and the user interface. That is closely linked to the aims and objectives of the project. These decisions secure the system's performance at two different levels of protection: robustness and scalability of the PhishGuard.

4.2 Data Cleaning

The CEAS-08 dataset, comprising over 70,000 rows of email records, required extensive preprocessing to support both graph-based and text-based phishing detection. The first step in the data cleaning process involved isolating the relevant columns primarily sender, receiver, and body and discarding incomplete or malformed entries. Custom Python scripts were used to parse and validate each email address, ensuring proper format and removing rows without `@` symbols. The dataset was further processed to remove duplicates, normalize spacing and encoding issues, and convert the textual content into lowercase for uniformity. Specific attention was given to trimming metadata, such as email headers or non-informative tokens, from the message body to retain only meaningful textual content for Roberta classification.

Further cleaning functions were applied to extract domain-level information from both sender and receiver addresses. It involved breaking an e-mail into subcomponents such as subdomain, primary domain, and top-level domain (TLD). These components were used to engineer graph-based features that reflect behavioural patterns in communication. Regex was used to handle variations in email formatting, while partition_email() functions parsed and classified domain elements. A final structured dataset was produced that included sender-receiver pairs, domain hierarchy, and engineered features like Sender_Primary_Domain, which were later fed into Neo4j and used for Random Forest training. This dual-purpose cleaning ensured that both textual and structural aspects of phishing behaviour could be effectively analysed.

4.3 Dataset Loaded into Neo4j & Exporting Graph Metrics

An instance of the Neo4j Database Management System-based database, version 5.26.0, the enterprise edition, has been created and attached to the project "FYP". Up to date, the working database has been assigned to localhost with communications taking place through Bolt port 7687 and HTTP port 7474, which are the default settings useful when working with Neo4j Browser or Python scripts that use Neo4j. This form of implementation has the critical advantage of preventing any intrusion attempts on the documentation over the system developer's end machine. Most importantly, it affords a highly preferable setup for the uploading of vast amounts of the sender-receiver domain information that is obtained after all the irrelevant entities have been purged from the email. It also allows the logical representation of the flow of emails in graph theory using levels and lines as per the two states.

This code snippet demonstrates batch-inserting structured email communication data into a Neo4j graph database using Python and Cypher queries. The dataset (cleaned_email_pairs_with_count_v1.1.csv) is read into a panda DataFrame and converted to a dictionary for processing. Within the insert_batch function, for each email row, it constructs the receiver domain hierarchy—including components like subdomain, second-level subdomain, primary domain, top-level domain (TLD), and secondary TLD—using MERGE statements to avoid duplication. These domain components are then linked together with [:HAS] relationships to reflect domain structure. Finally, it creates a [:SENDS] relationship from the sender domain to the receiver domain, with a communication count property. This structure allows Neo4j to represent real-world domain-to-domain interactions, which can be analyzed further by means of graph analytics such as PageRank and spam pattern discovery.

Once the cleaned email data had been uploaded into Neo4j, four graph algorithms PageRank, ArticleRank, Degree Centrality, and Louvain community detection were executed on the domain graph to draw behavioural insights for each email-domain node. These algorithms essentially measured the sender and receiver domains' influence, connectedness, and community structure with each other within the communication network. PageRank and ArticleRank measured domain importance based on link structures, Degree Centrality captured the volume of connections a domain had, while Louvain identified clusters of domains frequently interacting with each other. Once computed, these metrics were exported into a CSV file and later merged into the main dataset. This enriched domain-level feature set served as input for training the Random Forest classifier, enabling it to detect phishing behaviour based on domain reputation and network positioning.

4.4 Text Classification

In both text classification and particularly, for the purpose of classifying spam and ham emails, RoBERTa was chosen as the main model for this project. Compared to the initial BERT model created by Facebook AI, RoBERTa is considered as a superior model by fine tuning the optimization techniques adopted during training which in turn enhances its variance on different NLU challenges.

This is due to the method's elusiveness for the Next Sentence Prediction (NSP) task in preprocessing and the general tout learning rate, which is much larger in BERT, it also does not require NSP in the conte of learning. This is named RoBERTa-base and similarly outperforms numerous BERT core variants in many such tasks such as text classification works (Chessa, et al., 2021).

These other objectives that they were looking to achieve in this phase were the development of a one sentence sample prevention system, in the sense that each incoming email body is a single input sentence. This method is suitable when light and real-life junk mail screening processes are required, cases where fast decisions per email must be taken without involving long history of many sentences (Zhou, et al., 2022). Classification of single sentences employing transformer models such as RoBERTa has been shown to work very well in detection of spam after such fine-tuning with respect to domain-specific data has been carried out (Zhou, et al., 2022).

The RoBERTa model and tokenizer were accessed using the Hugging Face Model Hub, which is an online platform that boasts of the finest selection of pre-trained models for straightforward import and fine-tuning (Hugging Face, 2025). Just as Hugging Face Transformers likewise provided a user-friendly API that is capable of loading, fine-tuning, and sizing down the model for export purposes also making it an apt pick for speedy model development as well as in deployment on cloud devices such as Google Collab Pro.

To sum up, RoBERTa turned out to be highly effective in carrying out this work even with the relatively noisy email text data since it showed very high-performance metrics. The ability to learn subtle nuances, and manual integration with Hugging

Face for streamlined fine-tuning, has allowed us to achieve very good performance rapid development and accurate spam forensics.

```
Epoch 5 - Avg Loss: 0.0072
Validation Metrics:
  - Accuracy:  0.9916
  - Precision: 0.9858
  - Recall:    0.9993
  - F1 Score:  0.9925

Model saved to: /content/d
```

Figure 3. Text Classification Model Accuracy

The RoBERTa-based text classification model was trained to distinguish between phishing and legitimate. The training process involved feeding pre-processed email body text into the model, optimizing it over several epochs using a binary cross-entropy loss function. Throughout training, a validation set was used to monitor performance and ensure generalization. The training was executed in a Google Collab environment, with the model being saved automatically at the end of the final epoch for future inference or deployment.

By epoch 5, training strong convergence was achieved, with the average training loss being 0.0072, which implies minimal error in prediction. Validation set evaluation produced good results, with 99.16% accuracy, 98.58% precision, 99.93% recall, and 99.25% F1-score. This shows that the model has a great capability to detect phishing emails while keeping false positives to a minimum. The high recall ensured that practically all phishing messages were flagged, which is of great importance in cybersecurity applications, whereby missing attacks can have dire consequences.

Overall, this RoBERTa model showed excellent performance, balancing high precision with near-perfect recall. It was successfully integrated into the PhishGuard system as the core content-based classifier, complementing the graph-based domain analysis to form a hybrid phishing detection pipeline.

4.5 Random Forest

The ceas08_cleaned_final.csv file is imported with the help of the panda's library. This process made a pipeline Full of SimpleImputer and StandardScaler models, the purpose is to refine and give the quantitative figures a particular form. This makes it possible to avoid the lack of necessary numbers and tunings of feature units, which is a necessity before any machine learning methods, for instance Random Forest are applied.

It entails the most customary way of fitting a model to the dataset. Primarily, the initial stage of Random Forest creation is accomplished without any parameter tuning, serves for a quick appraisal of how the model explains the data prior to improvement of hyperparameters. It paves the way for a benchmark in the following screencast for comparing performance after calibration and modification of the model parameters. The GridSearchCV is used and at the different configurations of the parameters, including n_estimators(maximum number of trees), max_depth(maximum tree size) and min_samples_split(minimum samples required to split a node), should be noticed at an attempt to develop the best models. This feature remains beyond the scope of this type of model as most of them evaluate parameters for a certain number of trials without a regional shift over longer or shorter periods.

```
=== Initial Graph Model ===
Confusion Matrix:
[[500   3]
 [ 12  20]]

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.99       503
           1       0.87      0.62      0.73        32

    accuracy                           0.97       535
   macro avg       0.92      0.81      0.86       535
weighted avg       0.97      0.97      0.97       535
```

Figure 1. Random Forest before Modelling

	precision	recall	f1-score	support
0	0.99	0.99	0.99	503
1	0.88	0.88	0.88	32
accuracy			0.99	535
macro avg	0.93	0.93	0.93	535

Figure 2. Random Forest Model Accuracy

macro avg	0.93	0.93	0.93	535
accuracy			0.99	535
macro avg	0.93	0.93	0.93	535
macro avg	0.93	0.93	0.93	535
weighted avg	0.99	0.99	0.99	535
weighted avg	0.99	0.99	0.99	535

Saved tuned graph model

Figure 6. Graph model

The graph theory model aimed at phishing detection was put to work using graph features derived from the Neo4j email communication network. These graph features included PageRank, ArticleRank, Degree Centrality, and Louvain community scores. On the first phase, the initial Random Forest model produced an all-encompassing accuracy of 97%. Nevertheless, the model had a bigger pitfall on detecting phishing emails than the legitimate emails, (label 1), where it could only achieve a recall rate of 0.73, a precision rate of 1.00, and an F1-score of 0.62. It indicated that the model is making errors on many phishing emails, which are most of the time underrepresented in any data.

However, it still had models where the precision-recall curve lop-sided more towards one class than the other. Imbalance position improvement simply meant that it must once again apply hyperparameter. Such adjustments helped the model classifier to better explain sampling was the balance which was introduced to the because the initial dataset had serious drawbacks of skewness which prevented them from building trees of sufficient depth and at the same time allowed to regress very idle features in the nodes. Finally, this model was re-trained at the verses with the optimal parameters which were found.

After tuning, the re-evaluated graph model demonstrated a significant performance boost. The model achieved an overall accuracy of 99%, with phishing detection (label 1) now improved to a precision of 0.88, recall of 0.88, and F1-score of 0.88. The model also maintained perfect scores of 0.99 across all metrics for ham detection (label 0). Macro average and weighted average metrics both rose to 0.93 and 0.99 respectively, confirming consistent performance across both classes. These results indicate that the graph-based Random Forest model, after proper tuning, is highly capable of identifying phishing domains based solely on their structural and relational features within the email graph.

4.6 Models Uploaded to WANDB

WandB has been designed in such a way that it will be responsible for dynamically fetching the critical models, ensuring that the spam identification activity takes place, load_models_from_wandb() is responsible for initiation of the latest Weights & Biases run, tagged with "specific job type" load-models. This begins to tie between the local script and the WandB cloud repository, where earlier training sessions had sampled the final models.

This will dynamically load all models sequentially. First, the RoBERTa was retrieved, a text classification model, as an ONNX binary, which is stored in the artifact roberta-email-classifier: v0. The download of artifact directory will lead to the generation of the model file (roberta_email_classifier.onnx) to be subsequently fed into ONNX Runtime's inference session ort.InferenceSession for speedy email text body inference. Then the random forest model is fetched (spread over graphs) from artifact spam-graph-model:v2. The joblib file loads itself into memory with the help of function - joblib.load.

Upon successfully loading all models and finishing with the above function output; RoBERTa ONNX inference session and Graph Random Forest model model-the mentioned parts are the regions to continue with in the testing pipeline to numerate ensemble predictions with each new email sample.

4.7 Test Mode

In test mode, the PhishGuard system performs an end-to-end evaluation of incoming emails by leveraging both graph-based and text-based machine learning models. The process begins by establishing a secure connection to the Neo4j Aura cloud database, where domain-level graph metrics have been precomputed and stored. When a test email is received, its sender and receiver addresses are extracted and parsed to obtain components like the primary domain and top-level domain. These domain values are then used to query Neo4j for key metrics including PageRank, ArticleRank, Degree Centrality, and Louvain cluster scores, which are essential inputs for the graph-based classifier.

Concurrently, the system logs into Weights & Biases (WandB) and accesses the two pre-trained models as described, RoBERTa model for inferencing content-based classifications and the Random Forest Graph model for inferring domain structural features. The RoBERTa model is applied to the content of the email, and probabilities are computed to determine whether the email message is a phishing message or not based on text features. On the other hand, Neo4j supplies the graph metrics, and the machine learning model (Random Forest) is run with these graph metrics to assess if the email is spam or not.

The final classification output is displayed in the Streamlit based graphical user interface, where detection results of the e-mail message are observed in conjunction with the probability scores from both models. Whenever an incoming e-mail is determined to be spam, the user is drawn to the risk alert section which offers the metrics and either move to inbox or leave it as it is. In addition, all this is achieved remotely via cloud hosted WandB machine learning models. No manual operations are performed.

4.8 Live Mode

The PhishGuard system's Live Mode, when active, is equipped with automated phishing alerts and warns users in real-time by connecting to their Gmail accounts and scanning all emails for malware and phishing. Live Mode operates in a fully automated manner and does not require any manual upload or test input. It is based on a Google Apps Script Web App that operates via the Gmail API to load messages from both the Inbox and Spam folders while maintaining the security of the information. The script is deployed as a web service, which provides a clean JSON response that includes elements such as the sender email, recipient email, thread ID and body of the email whether it is an Inbox or Spam message. As of now, the Apps Script is still utilized to support the backend layer of the application, specifically the FastAPI service. When the FastAPI endpoint is called, it sends a GET request to the deployed Apps Script URL, retrieving the most recent emails from the user's Gmail account (typically from both the Inbox and Spam folders). The response comes in JSON format and includes critical fields such as sender, receiver, subject, email body, and a label indicating the folder. FastAPI then parses the response and extracts the necessary details. The sender and receiver addresses are processed to derive their primary domains, which are essential for further querying. These domain components are passed to Neo4j Aura, a cloud-hosted graph database, to fetch key graph metrics such as PageRank, ArticleRank, Degree Centrality, and Louvain cluster IDs. These scores are used by the Random Forest classifier to evaluate phishing risk based on sender behaviour in the graph.

In addition, the email body is forwarded for analysis by the text classifier, which was previously trained, exported into ONNX, and stored in Weights & Biases (WandB). This model analyses the textual content of the email and generates a spam-probability score. Simultaneously, a Random Forest classifier, again recovered from WandB, ingests domain-based graph metrics and issues an alternative prediction on whether the sender's domain shows suspicious behaviour due to its interaction profiles within the email graph. Both predictions are kept separate and offer insights from different points of view: content on one side and behaviour on the other.

The cloud backend supporting the live detection workflow is hosted on DigitalOcean, thus utilizing its scalable compute infrastructure. The FastAPI service interspersing the communication with Apps Script, Neo4j Aura, and WandB is deployed on a DigitalOcean Droplet. This deployment allows fast inference response times, secured endpoints, and continuous uptime without having to depend on heavier cloud services that require intricate billing or configuration. The system has the overhead of running on DigitalOcean yet remains fully capable of carrying out graph queries, model inferences, and Gmail API actions, thus making it viable for individual as well as organizational use.

In live mode, a rule scoring system is used to make the final classification decision so that none of the meta-models are employed. If any email gets a spam score under RoBERTa or graph score exceeding a given threshold (e.g., 0.9 or 0.8 respectively), that email is declared as spam or phishing. Hence the method is my way of keeping the spam filtering system sensitive while ensuring interpretability and transparency. At the end of the classification, the label gets passed from the FastAPI backend to the Google Apps Script side, where Gmail-related actions execute themselves. If it turns fraudulent, the system calls GmailApp.getThreadById().moveToSpam() to move the thread right away to the Spam folder.

In real-time with interaction from Gmail, FastAPI, WandB, and Neo4j, this demonstrates PhishGuard as a cloud-integrated Phishing detection system. It offers a backend pipeline for the entire process wherein emails are scanned live, analyzed by two complementary models, and acted upon automatically. There is no separate UI users need to interface with in this mode; all classification and quarantine actions occur within their actual Gmail inbox, with a smooth flow and without interruption. This lightweight deployment model allows PhishGuard to work reliably, either for an individual or organization, without requiring deep infrastructure or client-side changes making it scalable, efficient, and effective against evolving phishing threats.

5. Conclusion

In conclusion, this investigation into PhishGuard demonstrates its potential as an innovative solution for combating phishing threats through the integration of graph-based machine learning, email quarantine mechanisms, and cloud deployment Through its use of a graph database like Neo4j, PhishGuard offers a more dynamic and contextual dimension to phishing detection, beyond usual models recognizing traditional feature sets. Analysing sender-receiver profiles and interconnected data could lead to the pinpointing of subtle phishing patterns more accurately. PhishGuard fits well with the Sustainable Development Goal 9 (Industry, Innovation, and Infrastructure), having alongside it the establishment of a secure and resilient digital infrastructure. With phishing remaining a major threat to industries all around the world, PhishGuard promotes innovative cybersecurity measures to protect email and ensure networks are reliable for communication. The project also concentrates on scalability and accessibility as a cloud service to allow organizations of every size to carry out cutting-edge and practical cybersecurity. This research reflects the symbolism of PhishGuard as not being merely technologically advanced but also an effort toward creating a safer and more innovative digital future. Addressing the urgent issue of phishing and green technology brings to focus the need to promote innovation alongside infrastructure for cybersecurity enhancement on a larger scale. A major disadvantage is that one of the most significant ways to improve varying technological methods of training is through synthetic data. This, in turn, very much limits the system's adaptation abilities so that it could be used in real-life cases, as there is a wide range of phishing activities. At the same time, constraints, including the use of computing resources of the local sites for machine learning as well as graph processing, may have limited both system scalability and comprehensive testing of individual blocks.

Now, a real-time detection module that integrates with major email vendors, such as Gmail or Outlook, has not yet been implemented. In fact, these limitations still stand in the way of an ultimately usable system, rendering its current implementation appropriate mostly for inspecting static datasets. Furthermore, web hosting would be expensive and would allow for fewer instances now. Thus, in the presence of major financial constraints, cloud deployment and load testing appear compromised, therefore deeming the implementation highly unreliable with great demand for traffic or resources. PhishGuard can be further improved by enhancing dataset quality through partnerships with organizations that have access to authentic phishing data, which would increase the model's robustness and generalization capabilities. Furthermore, another adjustment can be made in such a way that it may leverage cloud GPU resources during training and testing; this would remove the present computational bottlenecks and, in general, increase viability. Real-time email analysis and integration with platforms such as Gmail could see the adoption of such a system greatly enhanced. These upgrades would ideally be completed before the full implementation of the quarantine-related system and its notification features, so as to better appeal to organizational users. Also, detailed cost estimates and forecasting for cloud services should be undertaken so that resource management may also be simplified and the system ready for large-scale deployment.

Corresponding author

Aitizaz Ali
aitizaz.ali@apu.edu.my

Acknowledgements
NA

Funding
NA

Contributions

Conceptualization, H.R; N.I; N.A.A.R; A.A; Methodology,H.R; N.I; N.A.A.R; A.A; Software, H.R; N.I; N.A.A.R; A.A; Validation, H.R; N.I; N.A.A.R; A.A; Formal Analysis, H.R; N.I; N.A.A.R; A.A; Investigation, H.R; N.I; N.A.A.R; A.A; Resources; H.R; N.I; N.A.A.R; A.A; Data Curation, H.R; N.I; N.A.A.R; A.A; Writing (Original

Draft), H.R; N.I; N.A.A.R; A.A; Writing (Review and Editing), H.R; N.I; N.A.A.R; A.A; Visualization, H.R; Supervision; H.R; Project Administration, H.R; Funding Acquisition, H.R. All authors have read and agreed to the published version of the manuscript.

Ethics declarations
This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for publication
Not applicable.

Competing interests
All authors declare no competing interests

References

[1] Hewage, C., Khan, I. A., Nawaf, L., & Alkhalil, Z. (2021, February). *Phishing attacks: A recent comprehensive study and a new anatomy*. ResearchGate.https://www.researchgate.net/publication/349312504_Phishing_Attacks_A_Recent_Comprehensive_Study_and_a_New_Anatomy
[2] Adil, M., Farouk, A., Ali, A., Song, H., & Jin, Z. (2025). Securing tomorrow of next-generation technologies with biometrics, state-of-the-art techniques, open challenges, and future research directions. *Computer Science Review, 57*, 100750.
[3] Al-Maari, A. A., Abdulnabi, M., Nathan, Y., Ali, A., Ali, U., & Khan, M. (2025). Optimized credit card fraud detection leveraging ensemble machine learning methods. *Engineering, Technology & Applied Science Research, 15*(3), 22287–22294.
[4] Addula, S. R., & Ali, A. (2025). A novel permissioned blockchain approach for scalable and privacy-preserving IoT authentication. *Journal of Cyber Security and Risk Auditing, 2025*(4), 222–237.
[5] Frederick, N., & Ali, A. (2024). Enhancing DDoS attack detection and mitigation in SDN using advanced machine learning techniques. *Journal of Cyber Security and Risk Auditing, 2024*(1), 23–37.
[6] Nadeem, M., Zahra, S. W., Abbasi, M. N., Arshad, A., Riaz, S., & Ahmed, W. (2023, September). *Phishing attack, its detections and prevention techniques*. ResearchGate. https://www.researchgate.net/publication/374848676_Phishing_Attack_Its_Detections_and_Prevention_Techniques
[7] Putra, F. P., Ubaidi, U., Zulfikri, A., Arifin, G., & Ilhamsyah, R. M. (2024, August). *Analysis of phishing attack trends, impacts and prevention methods: Literature study*. ResearchGate. https://www.researchgate.net/publication/383193964_Analysis_of_Phishing_Attack_Trends_Impacts_and_Prevention_Methods_Literature_Study
[8] SiteGround. (2024). *What are email protocols (POP3, SMTP and IMAP) and their default ports?* https://www.siteground.com/tutorials/email/protocols-pop3-smtp-imap/
[9] Booker, E. Z. (2024). *How email systems are designed?* OpenGenus. https://iq.opengenus.org/how-email-systems-are-designed/
[10] Aleksic, M. (2022, April 14). *IMAP vs. POP3 vs. SMTP: What are the differences?* PhoenixNAP. https://phoenixnap.com/kb/imap-vs-pop3-vs-smtp
[11] Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021, February). *Phishing attacks: A recent comprehensive study and a new anatomy*. ResearchGate. https://www.researchgate.net/publication/349312504_Phishing_Attacks_A_Recent_Comprehensive_Study_and_a_New_Anatomy (duplicate of [1], included once)
[12] Barracuda. (2022, March). *Spear phishing: Top threats and trends*. https://www.barracudamsp.com/content/dam/barracuda-msp/docs/resources/pdf/reports/RP-Spear-Phishing-vol7.pdf
[13] Petrosyan, A. (2023, March 17). *Volume of spear phishing and whaling attacks on organizations worldwide in 2021*. Statista. https://www.statista.com/statistics/1147426/volume-phishing-attacks-organizations-face-it-professionals/
[14] Colback, L. (2024, July 3). *Technology and cyber crime: how to keep out the bad guys*. Financial Times. https://www.ft.com/content/8a79ab25-c902-4110-bcb8-be2fd422f6bf
[15] IBM. (2024). *What is machine learning (ML)?* https://www.ibm.com/topics/machine-learning
[16] Chugani, V. (2024, October 1). *Industries in focus: Machine learning for cybersecurity threat detection*. Machine Learning Mastery. https://machinelearningmastery.com/industries-in-focus-machine-learning-for-cybersecurity-threat-detection/
[17] Ballejos, L. (2024, October 16). *The role of machine learning in cybersecurity*. NinjaOne. https://www.ninjaone.com/blog/machine-learning-in-cybersecurity/
[18] Altwaijry, N., Al-Turaiki, I., Alotaibi, R., & Alakeel, F. (2024, March 24). *Advancing phishing email detection: A comparative study of deep learning models*. MDPI. https://www.mdpi.com/1424-8220/24/7/2077
[19] Wolert, R., & Rawski, M. (2023, June). *Email phishing detection with BLSTM and word embeddings*. ResearchGate. https://www.researchgate.net/publication/377592515_Email_Phishing_Detection_with_BLSTM_and_Word_Embeddings
[20] Chessa, M., Panebianco, M., Corbu, S., Lussu, M., Dessì, A., Pintus, R., ... & Fanos, V. (2021, July). *Urinary metabolomics study of patients with bicuspid aortic valve disease*. ResearchGate.

https://www.researchgate.net/publication/353215356_Urinary_Metabolomics_Study_of_Patients_with_Bicuspid_Aortic_Valve_Disease
[21] Zhou, H., Xiao, X., Ali, A., Ali, A., Han, D., Zheng, W., ... & Zhou, Q. (2022, March). *Integration of GWAS and transcriptome analyses to identify SNPs and candidate genes for aluminum tolerance in rapeseed (Brassica napus L.)*. ResearchGate. https://www.researchgate.net/publication/359389739_Integration_of_GWAS_and_transcriptome_analyses_to_identify_SNPs_and_candidate_genes_for_aluminum_tolerance_in_rapeseed_Brassica_napus_L
[22] Hugging Face. (2025). *Transformers documentation*. https://huggingface.co/docs/transformers/index
[23] Adil, M., Abulkasim, H., Ali, A., Song, H., Farouk, A., & Jin, Z. (2024). Role of 5G and 6G technologies in metaverse, quality of service challenges and future research directions. *IEEE Network*.
[24] Alkhdour, T. A. Y. S., Alrawashdeh, R. A. N. A., Almaiah, M. O., Alali, R. O., Salloum, S. A., & Aldahyani, T. H. (2024). A new technique for detecting email spam risks using LSTM-particle swarm optimization algorithms. *Journal of Theoretical and Applied Information Technology, 102*(14).
[25] Ali, A. (2025). The impact of AI-generated content on customer and patient service optimization with clinical decision support. *Babylonian Journal of Artificial Intelligence, 2025*, 107–116.
[26] Naveed, F., Masih, A., Mahmood, J., Ahmed, M., Ali, A., Saddiqa, A., ... & Agbozo, E. (2025). Sustainable AI for plant disease classification using ResNet18 in few-shot learning. *Array, 26*, 100395.
[27] Ullah, R., Sarwar, N., Alatawi, M. N., Alsadhan, A. A., Alwageed, H. S., Khan, M., & Ali, A. (2025). Advancing personalized diagnosis and treatment using deep learning architecture. *Frontiers in Medicine, 12*, 1545528.