

A hybrid approach to phishing email detection: leveraging machine learning and explainable artificial intelligence

Tarek Zidan¹, Fadi Abu-Amara², Ahmad Hasasneh¹, Muath Sawaftah¹, Seth Griner²

¹Department of Natural Engineering and Technology Sciences, Faculty of Graduate Studies, Arab American University, Ramallah, Palestine

²Division of Applied Technology, Cybersecurity Program, Shenandoah University, Virginia, United States

Article Info

Article history:

Received Jan 15, 2025

Revised Jun 19, 2025

Accepted Jun 30, 2025

Keywords:

ChatGPT hybrid model

Cybersecurity awareness

Machine learning

Natural language processing

Phishing detection

ABSTRACT

With the increasing use of emails in our daily lives, they have become a prime target of phishing attacks, posing a significant threat to users. Attackers pretend to be trusted sources and use email phishing attacks to trick people into clicking malicious links or opening attachments. The aim of these attacks is to obtain sensitive information, such as financial information, login credentials, and personally identifiable information. Emails have attributes including the URL, sender, subject, receiver(s), and body. This paper proposes a hybrid intelligence model that integrates machine learning algorithms (ML) and natural language processing (NLP) techniques for email phishing detection. Three ML algorithms are employed: logistic regression, decision tree, and random forest. In addition, a customized ChatGPT model has been developed to receive email classification results from the hybrid model. This model educates users on recognizing phishing emails by explaining email classifications, highlighting keywords, and offering security tips. The proposed approach to detecting phishing emails raises awareness and educates users on recognizing and reporting email phishing attacks.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Ahmad Hasasneh

Department of Natural Engineering and Technology Sciences, Faculty of Graduate Studies, Arab American University

Ramallah, West Bank, Palestine

Email: ahmad.hasasneh@aaup.edu

1. INTRODUCTION

Email phishing targets organizations and individuals. Cyber attackers trick victims into opening fake forms or links to enter their sensitive information. This technique allows attackers to steal the victim's financial data, login credentials, or personally identifiable information. Since email phishing attacks are a growing concern, they cause significant economic, legal, and reputational damage. While traditional detection methods, such as blacklists, signature-based techniques, blocking messages with specific phrases, and statistical methods, have been effective, they struggle to keep up with ever-evolving phishing tactics [1], [2]. This limitation has led to the development of more advanced email phishing detection techniques [3], [4]. It is worth mentioning that analyzing the email, specifically through sender address, subject line, and text body, is the most used method to identify phishing attempts [1].

Phishing and spam emails are a significant concern. In 2023, about 45.6% of the emails sent worldwide were identified as spam or phishing, down from 49% in 2022. Spam and phishing emails remain a large part of email traffic; however, their share has significantly decreased since 2011 [5], [6]. Even with the decrease in the percentage of total traffic that phishing emails make up; there is still a need to protect users

from email phishing fraud. The rise in phishing attempts, combined with the limitations of traditional phishing detection methods, calls for more advanced solutions, such as machine learning (ML) algorithms [2]. In this paper, we focus our work on three ML algorithms. However, there are other solutions to such a problem.

Logistic regression is one of the simplest and first algorithms used in phishing and spam email classification; while it was not the best, it had significant involvement and improved the classification accuracy [7]. Another ML algorithm is the random forest (RF) classifier, built from a set of decision trees. The final classification is reached through aggregating the outputs from all decision trees. It involves independent training data and randomly selected features to train the model [8], [9]. The third ML algorithm, the decision tree, is easy to interpret and visualize and can handle both categorical and numerical features. This method gives various features that can be used to detect phishing emails [10].

While multiple ML and deep learning algorithms were used in email phishing detection, our research work focuses on ML since they established effectiveness in detecting phishing emails [2], [6]. A study that used logistic regression to classify emails based on textual features achieved an accuracy of 92.5% [11]. Another study integrated natural language processing (NLP) into logistic regression to extract features from the email body. It achieved an accuracy of 90.8% [12]. In another study, a decision tree classifier was trained on a large dataset that included various features such as email headers, body content, and embedded links. The proposed model achieved a higher accuracy of 94.1%, outperforming earlier approaches that utilized logistic regression [13]. Another work used a decision tree classifier to detect phishing emails with an accuracy of 93.2% [14].

Previous studies achieved promising email phishing classification results. However, there is a need for further improvement to cope with the evolving email phishing tactics used by attackers. In [15], a random forest classifier was applied to a dataset of phishing and safe emails containing email bodies, titles, headers, and other extracted information. The RF classifier achieved an accuracy of 96.4%. Another study used a random forest classifier in addition to using feature selection methods, such as filter, wrapper, and embedded methods. It achieved 97.2% classification accuracy [16]. Another study investigated the effectiveness of large language models (LLMs) in detecting phishing emails. The study concluded that GPT 3.5, ChatGPT, and GPT 3.5 Turbo Instruct exhibited high classification accuracies [17]. Another study tested the effectiveness of using LLMs in detecting phishing websites. Results indicated that GPT-4V achieved 98.7% accuracy [18].

The previous two papers discussed the effectiveness of utilizing LLMs in email phishing detection, concentrating on threat identification. The existing models frequently need a user-centric approach and a model that explains to users the rationale behind classifying an email as phishing, which is essential for sustained user awareness. This work addresses these limitations by employing LLMs not only for phishing detection but also for delivering real-time feedback that aids users in recognizing the indicators of email phishing attacks. By integrating interactive feedback, our model provides users with the ability to recognize phishing attacks.

Recent academic studies have used complex deep-learning architectures, especially transformer and convolutional models, to enhance email phishing detection. For example, a study utilized BERT-based embeddings optimized through a hill-climbing hyperparameter strategy. The study achieved 95% accuracy on a well-known Kaggle dataset [19], [20]. Furthermore, a feature-selection-based model was developed that included 79 static header and body features. It performed textual analysis on 661,000 emails. It achieved 95.97% accuracy with 0.1% false positives [16]. Another study utilized multi-agent and LLM-driven systems. The MultiPhishGuard framework employed five special agents for text, URLs, metadata, adversarial testing, and explanation. The agents' work was coordinated through reinforcement learning, yielding 97.9% accuracy and a 0.2% false-negative rate [21]. A follow-up study introduced a debate-driven configuration where two LLM agents were used to verify the legitimacy of an email before a judge model decided. This method improved detection and interpretability across multiple phishing datasets [22].

In this paper, we employ machine learning techniques to classify user-supplied email texts as legitimate or phishing. Moreover, the classification results are sent to OpenAI's GPT 3.5 using a unique application programming interface (API) and private key to communicate with. The GPT supplements the email classification results with information, such as the primary keywords used for classification. The private key used is 32 characters long and consists of alphanumeric characters. The key is generated by OpenAI when we created the API key through its platform using a secure random generation process that ensures its uniqueness and security [17]. The API key does not change periodically. Therefore, it is the user's responsibility to generate new keys or rotate them as needed. The model reads the user's email content to provide real-time feedback. Furthermore, once the user uploads an email, the model warns the user if the email contains any potential security threats, such as suspicious web links or attachments. Besides, it provides real-time assistance in recognizing email phishing attacks, improving an individual's ability to identify and manage mail-related risks. This assistance and feedback include a list of words to notice for future emails, explaining to the user the reason behind email classification as phishing, and providing relevant security tips.

The rest of the paper is organized as follows. Section 2 explores the proposed phishing detection tool and data collection methods. Section 3 discusses the experimental results, while section 4 concludes the paper and discusses future directions.

2. PROPOSED PHISHING DETECTION TOOL

With the great dependency on the internet these days and the increase in cyberattacks, cybersecurity awareness is essential to train and educate employees [23]. Phishing attacks are prevalent and target organizations of different sizes. As a mitigation strategy, organizations implement email phishing detection measures, such as blocklists and statistical classification. However, these protective measures are ineffective against sophisticated phishing attacks [23]. Due to the limitations of traditional email phishing detection methods, novel solutions are needed. This paper proposes an email phishing detection and classification method that utilizes artificial intelligence (AI) and ML. Our proposed method also uses a ChatGPT model to provide a real-time explanation so the user understands why an email is classified as phishing. Therefore, the proposed method spreads cybersecurity awareness and educates employees about phishing email attacks.

In study [24], the technology acceptance model (TAM) was utilized to investigate the effectiveness of using AI in spreading cybersecurity awareness. Another study developed an AI-based cybersecurity framework focused on threat forecasting and threat tree analysis [25]. In [26], the Delphi technique was utilized in developing cybersecurity awareness. It applied the Delphi technique to develop effective cybersecurity awareness strategies to employees.

Our proposed model uses the Python programming language, with the 'sklearn' library for ML algorithms. 80% of the dataset is used for training and 20% for testing. In addition, to evaluate effectiveness of the email classification model, we used the F1 score, precision, accuracy, and confusion matrix. Finally, we have used an optical character recognition (OCR) tool called pytesseract, provided by Google's OCR engine. We used the OCR tool to extract information from emails uploaded by the user as images and PDF files. Our phishing detection tool also accepts emails as text. Furthermore, the OpenAI API connects our code to OpenAI's LLM GPT 3.5. This functionality enables our model to transmit behavioral instructions, a prompt, and input parameters from the code to OpenAI's LLM GPT 3.5 [27]. In this case, the user's email and the ML prediction use the API to communicate with GPT 3.5 to return real-time feedback that provides a prediction and the reason behind email classification.

The three machine learning models are trained using a labeled dataset comprising phishing and legitimate emails. The dataset used in training this model is called Phishing_Email.csv, containing email texts labeled "Phishing Email" or "Safe Email." Before the training phase, all entries with missing values are eliminated, and the labels are transformed into binary values, where 0 indicates phishing and 1 indicates a safe email. A desktop application is developed using Python and the Tkinter framework to run the three machine learning models and offer the user an enjoyable experience via a graphical user interface. The training and testing processes are conducted concurrently through Python's ThreadPoolExecutor to enhance efficiency, where multiple threads run simultaneously. After testing, the model with the highest accuracy is selected to make predictions within the application. Our interface allows users to upload an email in three formats: raw text, PDF, or image. The PyPDF2 library is used to extract information from PDF files, while the Tesseract OCR engine, via the pytesseract library, is used to extract information from emails uploaded as images. Once the email content is retrieved, it is processed through the same TF-IDF vectorizer and forwarded to the chosen classifier for prediction. The output consists of a straightforward message informing the user whether the email is likely to be phishing or safe. Figure 1 shows the user interface for our model with two test cases. The first case is classified as phishing, while the second case is classified as a safe email.

The proposed model is trained on a large email dataset containing 28,748 entries. The prediction model receives a new email as text, a screenshot, or a PDF file. Next, the prediction model preprocesses the uploaded email to extract specific features, such as subject, sender information, and email body. Thereafter, the prediction model runs logistic regression, random forest, and decision tree classifiers concurrently. Next, the prediction model identifies the classifier with the best result by comparing the classifiers' accuracy, F1 score, recall, and precision. It then transmits the chosen classifier's output to the GPT model via API. Then, GPT is used to provide the user with personalized feedback that includes justification for the email classification result. It also highlights keywords that aid in determining the classification result. Finally, it gives advice that aids the user in identifying future phishing emails. Figure 2 shows the operational cycle of the proposed prediction model.

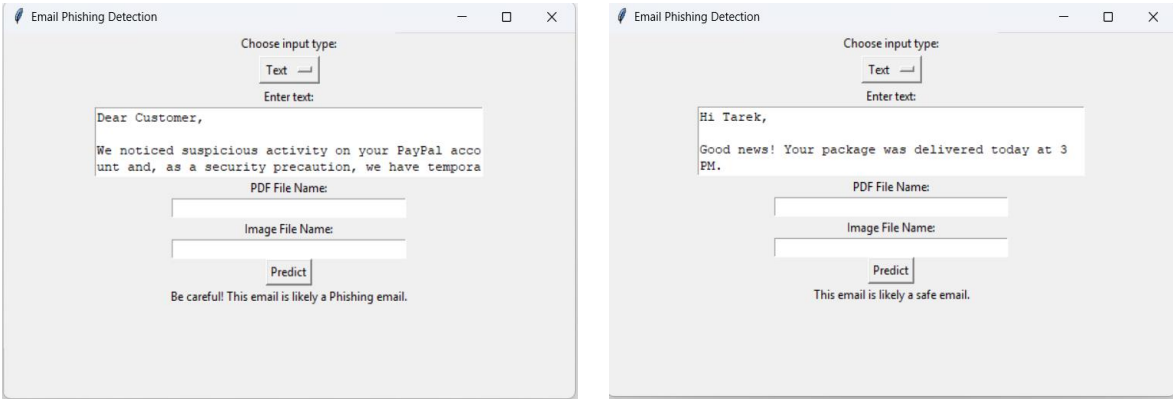


Figure 1. Model user interface

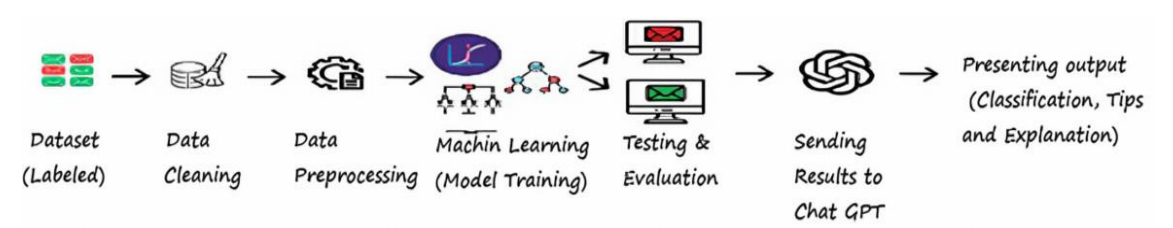


Figure 2. Prediction model cycle

After the training data is analyzed, the most frequent words are expressed using a “word cloud” after cleaning the dataset. The word cloud images are generated using a Python library called WordCloud, which visualizes the frequency of occurrence for each word after removing stop words, since they are useless. The higher the word frequency, the bigger the size drawn on the cloud. The word cloud images are split into two, one for legitimate and the other for phishing emails, as shown in Figures 3 and 4. Figure 4 shows that phishing emails are mainly related to financial incentives, well-known companies, popular products, and different offers to lure victims. Multithreading is used to implement the prediction model. To optimize processor performance, three machine algorithms, logistic regression, random forest, and decision tree, run concurrently on three different processor cores. Table 1 shows the experimental results for the three machine learning algorithms during training and testing. We applied data cleaning and preprocessing to the dataset, such as removing the empty records. Then, the data is trained using the three previously mentioned ML algorithms; each ML algorithm runs on a separate thread. After the training phase, the model is ready to receive new emails for classification. The email classification result and uploaded email are fed into GPT, which provides the user more information and feedback about the classification result and recommendations for dealing with similar future emails.

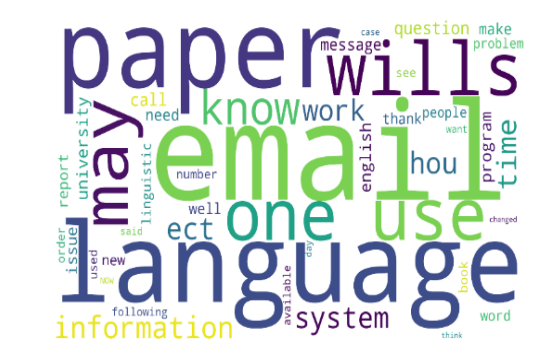


Figure 3. Safe emails word cloud



Figure 4. Phishing emails word cloud learning

The dataset consists of 18,634 emails. Each email is represented by three features: ID, email text, and email type. The dataset is preprocessed by removing duplicates and empty entries. The dataset used in training and testing consists of 11,322 legitimate and 7,312 phishing emails. Figure 5 shows the dataset email distribution. Furthermore, a test case was conducted by uploading a legitimate email to the email phishing detection model. The model's classification is correct, and the query is sent to the GPT using our API. The GPT responded with an explanation of why the email is classified as safe, in addition to some tips for identifying future similar emails. Figure 6 shows the screenshot of this interaction.

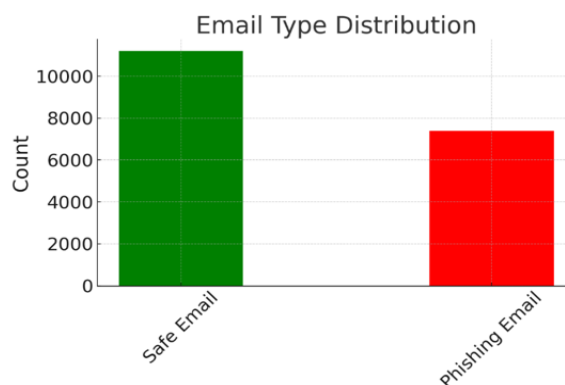


Figure 5. Dataset email distribution

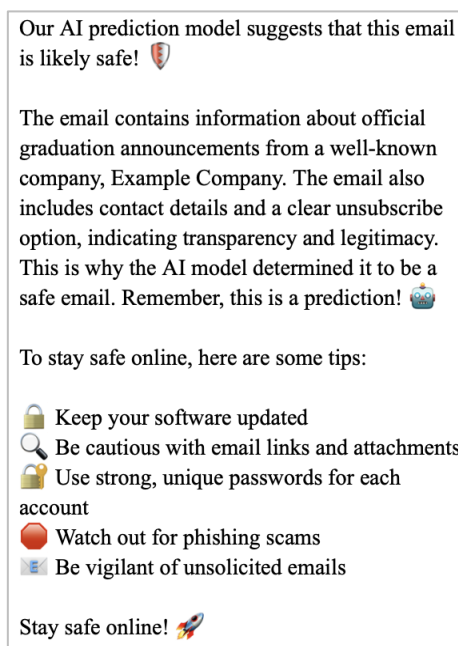


Figure 6. Model final output

Table 1. Classifiers' training and testing results after splitting our data into 80% training and 20% testing

Classifier	Training	Testing
Logistic regression	98.32%	97.20%
Decision tree classifier	98.87%	98.87%
Random forest classifier	98.87%	96.69%

We used the GridSearchCV with cross-validation to tune the hyperparameters of the three machine learning algorithms. The GridSearchCV is used to find the optimal combination of hyperparameters for each machine learning algorithm. We used the following parameters for the Random Forest classifier to balance

model complexity with training time. These parameters allow the random forest classifier to build trees with enough depth and a suitable number.

max_depth: can take the values {none, 10, 20, 30, 40, 50}. We set it to none to allow maximum tree depth.

criterion: Can take the values {'gini', 'entropy'}. We set it to the default value of 'gini' (Gini impurity), which allows the algorithm to determine the best split.

min_samples_split: can take a value from {2, 5, 10}. We set it to the default value of 2.

n_estimators: can take the values {50, 100, 200, 300}. We set it to the optimal value of 50.

For the decision tree classifier, the optimal settings are set as follows. These settings prevent overfitting of the model and maintain a dense tree to reduce underfitting:

max_depth: can take the values {none, 10, 20, 30, 40, 50}. We set it to none to allow maximum tree depth.

criterion: We set it to 'gini.'

min_samples_split: can take the values {2, 5, 10}. We set it to a default value of 2, indicating the minimum number of samples to split a node.

min_samples_leaf: Can take the values {1, 2, 4, 6}. We use the default value of 1, indicating the minimum number of samples for a leaf node.

3. EXPERIMENTAL RESULTS AND DISCUSSION

This section discusses the ML techniques employed to build the email phishing detection model. The used models are logistic regression, decision tree, and random forest, all trained on TF-IDF-transformed email data. The integration of the email phishing detection model with OpenAI's GPT-3.5 further enhances the system by providing user-facing interpretability and email phishing awareness feedback.

3.1. Logistic regression classifier

For the logistic regression classifier, optimal settings help control the training time and the convergence criterion, decrease the number of iterations, balance model complexity, and avoid overfitting the model by reducing the variance. The hyperparameter tuning process for the three classifiers optimizes each classifier for the email classification task. Logistic regression was the first machine learning algorithm tested, as it is a linear, simple, and effective binary classifier. It works by estimating the probability that a given input belongs to a specific class using the logistic (sigmoid) function.

In this work, we utilized the term frequency-inverse document frequency (TF-IDF) function to transform an email text into input features. The logistic regression algorithm produced 97.2% classification accuracy on the test set. The ease of implementation and interpretability make this algorithm a suitable model for email phishing detection. Figure 7 shows the evaluation confusion matrix of the logistic regression model. From this figure, we notice the effectiveness of the classifier with 1458 true negatives (TN), 2,165 true positives (TP), 43 false positives (FP), and 61 false negatives (FN). The precision and recall values are also high, at 98.1% and 97.2%, respectively, which indicates its high sensitivity in email phishing identification. It also exhibits a low false positive rate. The F1 score of 97.6% suggests an equal balance between precision and recall, making this model robust enough for classification purposes. The result demonstrates the robustness of this classifier in handling variations in data.

3.2. Decision tree classifier

The decision tree classifier is widely used across various applications due to its strong classification performance [7], [11]. It operates by recursively splitting the dataset into subsets based on the most informative attributes at each node, forming a hierarchical tree structure of decisions. In our analysis, the decision tree is particularly useful because it helps identify the most indicative features—such as specific words or phrases—associated with phishing emails. Figure 8 presents the confusion matrix illustrating the evaluation results of the decision tree model. This confusion matrix shows 1458 TN, 2165 TP, 43 FP, and 61 FN. The model has a classification accuracy of 97%, 80% reusability, and 2% proficiency in making correct predictions. Furthermore, the precision stands at 98. The actual positive measures the model's capacity to determine the existence of actual positive cases, which is 1% in the experiment. Furthermore, the recall rate is 97.2%, which shows that the proposed model achieved high effectiveness in retrieving the target positive samples, as revealed by 2%. The F-measure, the harmonic mean of precision and recall, is 97.6%, proving the chosen model's validity and effectiveness in the email classification.

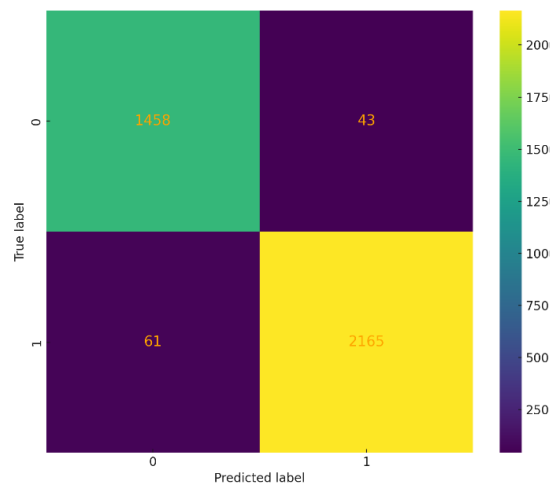


Figure 7. Logistic regression confusion matrix

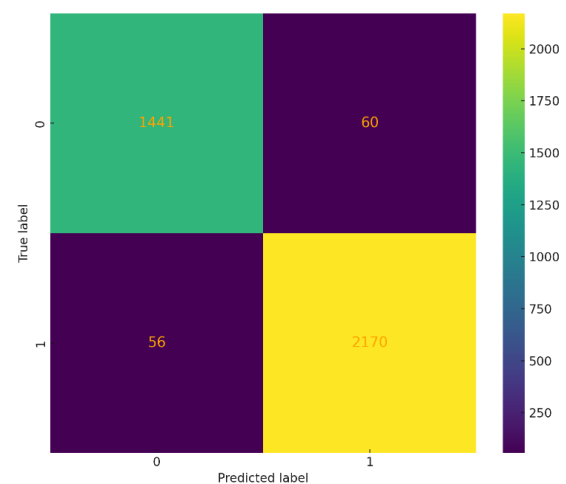


Figure 8. Decision tree confusion matrix

3.3. Random forest classifier

The random forest classifier utilizes multiple decision trees to reduce the overfitting risk and enhance its classification accuracy. In this classifier, each decision tree is trained on a random subset of the training data. To minimize overfitting risk and handle noise, we combine the prediction results of multiple decision trees. As a preprocessing step, emails to be classified are transformed into numerical features using the TF-IDF vectorization, which assigns weights to extracted words based on their frequency. Figure 9 shows the evaluation confusion matrix of the random forest. We have 1441 TN, 2170 TP, 60 FP, and 56 FN from the confusion matrix. This indicates a high accuracy of the model, which means a high rate at which emails are correctly classified. The test's response shows high accuracy, which means many optimistic predictions are accurate. It also shows high recall, which indicates the model's ability to identify the most positive instances. The F1 score, which averages both F scores, measures precision and recall, rightly emphasizing the model's strength and efficiency.

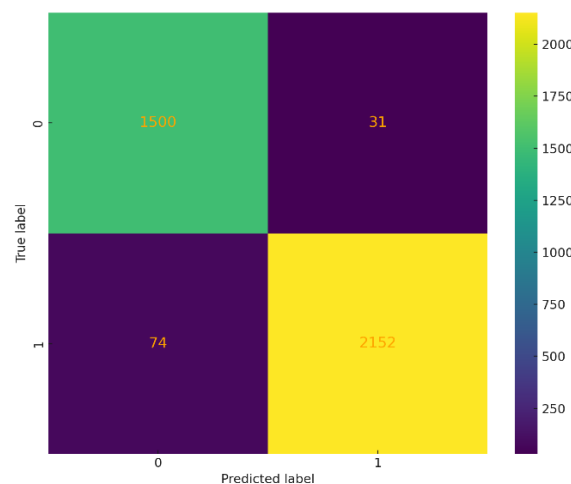


Figure 9. Random forest confusion matrix

3.4. Integration with GPT-3.5 for user feedback

The proposed ML model is integrated, via the OpenAI API, with GPT 3.5 LLM to raise awareness and educate users on phishing emails. We carefully crafted a prompt with specific instructions and constraints and then submitted it to the GPT. The GPT analyzes email content and classification results. It then generates customized feedback that explains the email classification result and educates users in recognizing similar future emails.

Once an email is classified as a potential phishing email, the following prompt is submitted to the GPT 3.5 LLM. It instructs the GPT to provide personalized feedback to raise cybersecurity awareness: “Create a response that uses around 160 completion tokens. Include emojis. You are an assistant who has just received the prediction that the user’s email is a phishing email. Under all circumstances: Don’t make a prediction yourself; we have an AI model responsible for forecasts. Only relay our model’s prediction and then give some advice. You’ll receive the email below. Give some reasoning on why it could be; a keyword could remind the user it’s a prediction, and could be phishing. Quote specific parts of the email in each part of your reasoning. Then, give some general tips on how to stay safe. Make the tips listed in an aesthetically pleasing manner.”

Once an email is classified as a potential safe email, the following prompt is submitted to GPT 3.5: “Create a response that uses 160 completion tokens. Include emojis. You are an assistant who has just received the prediction that the user’s email is safe. Under all circumstances: Don’t make a prediction yourself; we have an AI model responsible for predictions. Only relay our model’s prediction and then give some advice. You’ll receive the email below. Tell them it’s keyword likely; remind the user it’s a prediction, a safe email! You should quote parts of the email to show why this email could have been safe. Explain why the AI model decided the email is safe. Then, give some general tips on how to stay safe. Make the tips listed in an aesthetically pleasing manner.”

The experimental result findings indicate that users who interacted with the proposed system are 40% more inclined to accurately identify phishing emails in subsequent encounters than those who received merely static warnings using traditional systems. We developed Python code to integrate the ML email prediction algorithms with GPT-3.5 LLM. The GPT receives the email prediction result and the corresponding email and then provides personalized feedback.

4. CONCLUSION

This work explored the classification accuracy of three widely used ML algorithms for email phishing detection. The investigated algorithms were decision tree, logistic regression, and random forest. The three algorithms were trained using a large public dataset of English-language emails labeled safe or phishing. All three classifiers achieved high classification performance. The best email phishing detection performance was achieved by the decision tree classifier, correctly classifying 98.87% of emails. Furthermore, we integrated the proposed email detection model with OpenAI’s GPT API, which allows the transmission of email prediction results to the language model. By integrating ML and GPT, the proposed system successfully classified emails as phishing or legitimate. It also provided users with real-time feedback that included identified keywords used in the classification decision. These steps contribute to educating users about identifying future phishing attacks.

Future work includes training our prediction model to identify spear phishing attacks. These attacks are customized for known victims. Furthermore, the dataset will be enhanced to train the prediction model better. Also, more features will be extracted from emails. Additionally, LLMs will be integrated into the proposed prediction model to improve classification accuracy. Finally, we will use hybrid ML and deep learning models to train our model and reach higher accuracies effectively.

FUNDING INFORMATION

The publication fees for this manuscript were supported by the Barzinji Institute for Global Virtual Learning at Shenandoah University.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Tarek Zidan	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Fadi Abu-Amara	✓					✓				✓		✓	✓	✓
Ahmad Hasasneh	✓					✓				✓		✓	✓	
Muath Sawaftah	✓	✓	✓		✓				✓					
Seth Griner	✓	✓			✓				✓					

C : Conceptualization	I : Investigation	Vi : Visualization
M : Methodology	R : Resources	Su : Supervision
So : Software	D : Data Curation	P : Project administration
Va : Validation	O : Writing - Original Draft	Fu : Funding acquisition
Fo : Formal analysis	E : Writing - Review & Editing	

CONFLICT OF INTEREST STATEMENT




The authors declare that they have no financial interests or personal relationships that could have influenced the work reported in this paper. They report no conflicts of interest.

REFERENCES




- [1] K. Thakur, M. L. Ali, M. A. Obaidat, and A. Kamruzzaman, "A systematic review on deep-learning-based phishing email detection," *Electronics*, vol. 12, no. 21, p. 4545, Nov. 2023, doi: 10.3390/electronics12214545.
- [2] N. Ahmed, R. Amin, H. Aldabbas, D. Koundal, B. Alouffi, and T. Shah, "Machine learning techniques for spam detection in email and IoT platforms: analysis and research challenges," *Security and Communication Networks*, vol. 2022, pp. 1–19, Feb. 2022, doi: 10.1155/2022/1862888.
- [3] P. N. Wosah, Q. Ali Mirza, and W. Sayers, "Analysing the email data using stylometric method and deep learning to mitigate phishing attack," *International Journal of Information Technology*, May 2024, doi: 10.1007/s41870-024-01839-5.
- [4] P. Wanda, "GRUSpam: robust e-mail spam detection using gated recurrent unit (GRU) algorithm," *International Journal of Information Technology*, vol. 15, no. 8, pp. 4315–4322, Dec. 2023, doi: 10.1007/s41870-023-01516-z.
- [5] I. Moutafis, A. Andreatos, and P. Stefanias, "Spam email detection using machine learning techniques," *European Conference on Information Warfare and Security, ECCWS*, vol. 2023-June, pp. 303–310, 2023, doi: 10.34190/eccws.22.1.1208.
- [6] A. A. Akinyelu and A. O. Adewumi, "Classification of phishing email using random forest machine learning technique," *Journal of Applied Mathematics*, vol. 2014, pp. 1–6, 2014, doi: 10.1155/2014/425731.
- [7] A. O. Salau, T. A. Assegie, E. D. Markus, J. N. Eneh, and T. I. Ozue, "Prediction of the risk of developing heart disease using logistic regression," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 2, pp. 1809–1815, 2024, doi: 10.11591/ijece.v14i2.pp1809-1815.
- [8] R. Ageng, R. Faisal, and S. Ihsan, "Random forest machine learning for spam email classification," *Journal of Dinda: Data Science, Information Technology, and Data Analytics*, vol. 4, no. 1, pp. 8–13, 2024, doi: 10.20895/dinda.v4i1.1363.
- [9] S. T. T. Ibrahim, O. B. S. Adjunct Lecturer, and O. H. I. Part Time Lecturer, "Spam email detection scheme based on random forest algorithm," *LAUTECH Journal of Computing and Informatics*, vol. 3, no. 1, 2023.
- [10] A. A. Fazal and M. Daud, "Detecting phishing websites using decision trees: a machine learning approach," *International Journal for Electronic Crime Investigation*, vol. 7, no. 2, 2023, doi: 10.54692/ijeci.2023.0702155.
- [11] M. H. Alsuwit, M. A. Haq, and M. A. Aleisa, "Advancing email spam classification using machine learning and deep learning techniques," *Engineering, Technology & Applied Science Research*, vol. 14, no. 4, pp. 14994–15001, Aug. 2024, doi: 10.48084/etasr.7631.
- [12] Y. S. Navya, K. Pranathi, G. Srija, and S. H. Naaz, "Spam detection using machine learning: a logistic regression approach," *Advancement of Computer Technology and its Applications*, vol. 8, no. 3, pp. 1–10, 2025, doi: 10.5281/zenodo.15093547.
- [13] A. Yasin and A. Abuhasan, "An intelligent classification model for phishing email detection," *International Journal of Network Security & Its Applications*, vol. 8, no. 4, pp. 55–72, Jul. 2016, doi: 10.5121/ijnsa.2016.8405.
- [14] K. Omari, "Comparative study of machine learning algorithms for phishing email detection," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 9, pp. 417–425, 2023, doi: 10.14569/IJACSA.2023.0140945.
- [15] A. Alhaj, M. Abu-Faraj, and B. J. A. Ali, "A predictive technique using random forest classifier for phishing malicious attack," *Applied Mathematics & Information Sciences*, vol. 17, no. 6, pp. 1177–1187, Nov. 2023, doi: 10.18576/amis/170622.
- [16] H. Zhang, Y. Shi, M. Liu, L. Chen, S. Wu, and Z. Xue, "A combined feature selection approach for malicious email detection based on a comprehensive email dataset," *Cybersecurity*, vol. 8, no. 1, p. 14, Feb. 2025, doi: 10.1186/s42400-024-00309-6.
- [17] H. Patel, U. Rehman, and F. Iqbal, "Large language models spot phishing emails with surprising accuracy: a comparative analysis of performance," *arXiv:2404.15485*, 2024.
- [18] T. Koide, N. Fukushi, H. Nakano, and D. Chiba, "Detecting phishing sites using ChatGPT," *arXiv:2306.05816*, 2023.
- [19] N. Altwaijry, I. Al-Turaiki, R. Alotaibi, and F. Alakeel, "Advancing phishing email detection: A comparative study of deep learning models," *Sensors*, vol. 24, no. 7, 2024, doi: 10.3390/s24072077.
- [20] A. Gaurav, B. B. Gupta, A. Castiglione, S. Bansal, and K. T. Chui, "Optimized deep learning based phishing email detection using BERT and Hill climbing algorithm," *Lecture Notes in Computer Science*, vol. 15417 LNCS, pp. 258–269, 2025, doi: 10.1007/978-981-96-6389-7_23.
- [21] Y. Xue, E. Spero, Y. S. Koh, and G. Russello, "MultiPhishGuard: An LLM-based multi-agent system for phishing email detection," *arXiv:2505.23803*, 2025.
- [22] N. T. Vy Nguyen, F. D. Childress, and Y. Yin, "Debate-driven multi-agent LLMs for phishing email detection," *ISDFS 2025 - 13th International Symposium on Digital Forensics and Security*, 2025, doi: 10.1109/ISDFS65363.2025.11012014.
- [23] A. S. Alqahtani, S. A.-D. Qawasmeh, and M. K. Khan, "Navigating cybersecurity training: A comprehensive review," *Computers & Security*, vol. 135, p. 103013, 2023, doi: 10.1016/j.cose.2023.103013.
- [24] M. Farheen Ansari, "An effective cybersecurity awareness training model: first defense of an organizational security strategy," *International Research Journal of Engineering and Technology (IRJET)*, vol. 9, no. 4, pp. 1–6, 2022.
- [25] J. M. Camacho, A. Couce-Vieira, D. Arroyo, and D. R. Insua, "A cybersecurity risk analysis framework for systems with artificial intelligence components," *International Transactions in Operational Research*, 2025, doi: 10.1111/itor.70049.
- [26] A. Kaewsa-Ard and N. Utakrit, "Enhancing cybersecurity awareness strategies in organization using Delphi technique," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 15, no. 3, pp. 2986–2997, 2025, doi: 10.11591/ijece.v15i3.pp2986-2997.
- [27] G. Desolda, F. Greco, and L. Viganò, "APOLLO: A GPT-based tool to detect phishing emails and generate explanations that warn users," in *Proceedings of the ACM on Human-Computer Interaction*, 2024.

BIOGRAPHIES OF AUTHORS






Tarek Zidan    holds a bachelor's degree in computer engineering from Birzeit University, Palestine. Currently pursuing a master's degree in artificial intelligence at the Arab American University, Palestine. He is a teaching assistant at Birzeit University, where he teaches different labs such as Linux operating system, Python programming, and Android mobile applications. With various technical interests such as AI, machine learning, NLP, and mobile development. He can be contacted at: t.zidan@student.aaup.edu.






Fadi Abu-Amara    holds a Ph.D. and MSE in computer engineering from Western Michigan University, and a B.S. in computer engineering from Al-Balqa Applied University. He joined Shenandoah University in Fall 2022, after serving as an assistant professor in both Jordan and the United Arab Emirates. He brings over 24 years of experience in teaching, research, industry, and academic management. Dr. Abuamara has received multiple awards for exceptional performance, letters of appreciation from management, and consistently positive feedback from students. His teaching experience spans cybersecurity, computer science, and computer engineering. His research interests include cryptography, gamification for cybersecurity awareness, blockchain-based solutions for electricity and water management, and the use of robots to enhance academic skills in autistic children. He can be contacted at: fadi.abuamara@su.edu.






Ahmad Hasasneh    earned his B.Sc. in computer systems engineering from Palestine Polytechnic University in 2005 and his M.Sc. in computer graphics and programming from the University of Hull in 2006. He taught at Hebron University before receiving a Ph.D. in artificial intelligence and machine learning from Paris University in 2012. He has since held academic and leadership roles at Hebron University, PTUK, and Palestine Ahliya University, where he helped establish programs in multimedia and smart systems engineering. Since 2024, he has been an associate professor and department head at the Arab American University. His research focuses on machine learning applications in medical diagnostics, with active projects under the Palestinian German and Palestinian-Quebec Science Bridges, and collaborations with institutions in Germany, Canada, UAE, and Portugal. He has published widely in international journals and conferences. He can be contacted at: ahmad.hasasneh@aaup.edu.



Muath Sawaftah    holds a bachelor's degree in computer engineering from Al-Quds University, Palestine. Master's degree researcher in computer science and artificial intelligence at the Arab American/Al-Quds University, Palestine. He is an ICT specialist at the International Committee of the Red Cross ICRC, research interests include AI, machine learning, AI applications for Computer Architecture, AI for Healthcare applications, AI for humanitarian Aid and can be contacted at: LinkedIn <https://www.linkedin.com/in/mouath-sawaftah-393363148/> and he can be contacted at email: muath.sawafta@students.alquds.edu, engsawaftah@gmail.com.



Seth Griner    holds a bachelor's degree in information technology from Shenandoah University. During his undergraduate studies, he contributed to various IT based projects. He is currently working as a First-Year Admissions Counselor at Shenandoah University. He uses his IT background where possible to further admissions analytics. He plans to continue his education in either cybersecurity or data science. He can be contacted at: sgriner19@su.edu.