© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722448

Development of an Email Phishing Detection System Using TF–IDF Feature Extraction and Machine Learning Algorithms

ETUS C. (PHD.)[1], EWUNONU T. C. (PHD.)[2], CHUKS-UGOCHUKWU C.M. (PHD.)[3], ESOMONU N. F. (PHD.)[4], BENSON-EMENIKE M. E. (PHD.)[5]

1Department of Information Technology, Federal University of Technology, Owerri
0000-0001-9095-7196
[2]Department of Cybersecurity, Federal University of Technology, Owerri
0000-0003-3152-6788
[3]ICT Directorate, Alvan Ikoku Federal University of Education, Owerri
0000-0002-1979-6446
[4]Department of Information Technology, Federal University of Technology, Owerri
0000-0003-3736-8143
[5]Department of Computer Science, Federal University of Technology, Owerri
0000-0003-1771-5806

Abstract- ***The rapid expansion of digital communication has established email as a primary medium for information exchange among individuals, businesses, and organisations. This widespread reliance has contributed to a marked increase in phishing attacks, in which cybercriminals impersonate legitimate entities to obtain sensitive information, including login credentials, financial data, and personal details. Traditional phishing detection methods, including rule-based filters and blacklist mechanisms, have proven increasingly insufficient against sophisticated and evolving phishing strategies. As a result, there is a critical need for intelligent and adaptive detection systems capable of accurately identifying phishing emails. This study presents the development of an intelligent Email Phishing Detection System utilising supervised machine learning algorithms to enhance email security and protect users from phishing threats. A publicly available dataset containing both phishing and legitimate email messages was employed for model training and evaluation. The dataset underwent preprocessing steps including text cleaning, tokenisation, stop-word removal, and feature extraction using the Term Frequency–Inverse Document Frequency (TF–IDF) technique. Five supervised machine learning algorithms—Logistic Regression, Naïve Bayes, Decision Tree, Random Forest, and Support Vector Machine (SVM)—were trained and evaluated using standard performance metrics: accuracy, precision, recall, and F1-score. Experimental results indicated that the Support Vector Machine (SVM) outperformed the other classification models, achieving an accuracy of 99.1%, precision of 99.0%, recall of 99.1%, and an F1-score of 99.0%. Due to its superior performance, the SVM model was selected for deployment in the developed system. The proposed phishing detection system was implemented as a desktop application using Python's Tkinter graphical user interface (GUI), allowing users to input email content and receive real-time predictions regarding the legitimacy of emails. The findings demonstrate that machine learning techniques, particularly the Support Vector Machine algorithm, offer a highly accurate, reliable, and efficient approach to phishing email detection. Integrating the trained SVM model into a user-friendly desktop application provides a practical, lightweight, and scalable solution that enhances email security, reduces false detections, and assists users in more effectively identifying phishing attempts.***

Keywords - email phishing, machine learning, support vector machine, cybersecurity, detection system.

I. INTRODUCTION

The rapid advancement of Information and Communication Technology (ICT) has fundamentally transformed the way individuals, businesses, and organisations communicate and exchange information. Among the various communication platforms available today, email

remains one of the most widely adopted because of its speed, convenience, and cost-effectiveness. The increasing dependence on email for personal, commercial, and institutional communication has, however, exposed users to a growing range of cybersecurity threats, with phishing attacks emerging as one of the most persistent and damaging forms of cybercrime.

Phishing is a deceptive cyberattack in which malicious actors impersonate trusted organisations or individuals to manipulate recipients into disclosing confidential information such as usernames, passwords, financial credentials, and other sensitive personal data. According to Ahmed and Khan (2022), phishing attacks exploit human trust through carefully crafted messages that appear legitimate, making them highly effective even when technical security measures are in place. As digital communication continues to expand across industries, phishing campaigns have become increasingly sophisticated, employing advanced social engineering techniques that make fraudulent messages difficult to distinguish from genuine correspondence.

The widespread adoption of online banking, e-commerce, cloud services, and remote working environments has further increased users' exposure to phishing threats. Attackers frequently distribute deceptive emails containing forged sender identities, malicious hyperlinks, counterfeit websites, or infected attachments to persuade unsuspecting recipients to reveal sensitive information or install malicious software. Reports have shown that phishing continues to account for a significant proportion of cyberattacks worldwide, resulting in substantial financial losses, data breaches, identity theft, and reputational damage for both individuals and organisations (Ahmed, Rehman, & Hussain, 2024). These challenges have intensified the demand for more intelligent and adaptive mechanisms capable of detecting phishing attempts before users become victims.

Conventional phishing detection techniques primarily rely on rule-based filters, signature matching, heuristic analysis, and blacklist databases to identify suspicious emails. While these methods have proven effective against previously identified phishing campaigns, they often struggle to detect newly emerging or zero-day attacks because cybercriminals continually modify their attack strategies to evade traditional security mechanisms. Advances in artificial intelligence have further enabled attackers to generate highly convincing phishing emails that closely resemble legitimate communications, thereby reducing the effectiveness of conventional filtering approaches (Alkhalil, Hewage, Nawaf, & Khan, 2021). Consequently, traditional detection systems frequently experience high false-positive and false-negative rates, limiting their ability to provide reliable protection against evolving phishing techniques.

Machine learning has emerged as a promising solution to these challenges because of its ability to learn patterns directly from historical data and generalise its knowledge to previously unseen examples. Unlike conventional rule-based systems, machine learning models continuously identify complex relationships among textual, structural, and contextual characteristics within email messages, enabling them to distinguish phishing emails from legitimate correspondence with greater accuracy (Azmat Ullah, Javed, & Malik, 2024). Supervised learning algorithms such as Logistic Regression, Naïve Bayes, Decision Tree, Random Forest, and Support Vector Machine (SVM) have demonstrated considerable success in phishing email classification by analysing features including suspicious keywords, writing style, sender characteristics, embedded hyperlinks, and header inconsistencies. Similarly, deep learning models such as Long Short-Term Memory (LSTM) networks have shown strong capabilities in modelling sequential patterns within email content.

Recent studies have reported encouraging results using various machine learning and deep learning techniques for phishing detection. Zhang and Li (2020) developed a Bidirectional Encoder Representations from Transformers (BERT)-based framework that effectively captured contextual information within email text, achieving an accuracy of 94%. Wang et al. (2021) proposed a Convolutional

Neural Network (CNN) model for phishing URL detection using lexical and visual characteristics, attaining an accuracy of 96.3%. Patel, Mehta, and Chauhan (2021) introduced a lightweight Decision Tree model designed for mobile devices, achieving 90% accuracy while maintaining low computational requirements. Ahmed and Khan (2022) presented a multilingual Transformer-based phishing classifier that recorded 92% accuracy across multiple languages, whereas Tan and Lim (2022) improved detection performance by combining Random Forest and Logistic Regression models, achieving 93% accuracy. Chen, Li, and Zhou (2023) further demonstrated the effectiveness of XGBoost in detecting phishing attacks targeting e-commerce platforms, reporting an accuracy of 95%.

Although these studies have significantly advanced phishing detection research, several limitations remain. Many existing approaches concentrate primarily on URL analysis or specific application domains while paying relatively little attention to the semantic and contextual characteristics contained within complete email messages. Furthermore, several high-performing deep learning models require substantial computational resources, making them less suitable for deployment on standard personal computers and other resource-constrained environments. In addition, many proposed solutions remain research prototypes with limited emphasis on practical deployment through user-friendly applications that can easily be adopted by non-technical users.

This study addresses these limitations by developing an intelligent Email Phishing Detection System based on supervised machine learning techniques. A publicly available dataset containing both phishing and legitimate email messages was preprocessed using established natural language processing techniques, while Term Frequency–Inverse Document Frequency (TF–IDF) was employed to convert textual information into numerical feature representations suitable for machine learning. Multiple supervised classification algorithms were trained and evaluated using standard performance metrics, including accuracy, precision, recall, and F1-score, to determine the most effective classifier for phishing detection.

Based on the experimental evaluation, the Support Vector Machine (SVM) algorithm demonstrated superior classification performance and was therefore selected for deployment within the developed system. The trained model was integrated into a Python-based desktop application using the Tkinter graphical user interface, enabling users to analyse email content and obtain real-time predictions regarding whether an email is legitimate or phishing. By combining robust text preprocessing, efficient feature extraction, high-performance machine learning, and an accessible graphical interface, the proposed system provides a practical, lightweight, and scalable solution for enhancing email security and reducing the growing threat posed by phishing attacks.

II. PROBLEM STATEMENT

The rapid advancement of Information and Communication Technology (ICT) has significantly transformed electronic communication, making email one of the most widely used platforms for personal, commercial, and organisational interactions. Despite its numerous advantages, the increasing reliance on email has been accompanied by a corresponding rise in cybersecurity threats, particularly phishing attacks. Phishing emails are intentionally designed to deceive recipients into revealing confidential information, including usernames, passwords, banking credentials, and other sensitive personal data. Modern phishing campaigns have become increasingly sophisticated, often imitating legitimate organisations and employing advanced social engineering techniques that make fraudulent messages difficult for users to identify.

Traditional email security solutions primarily depend on rule-based filtering, signature matching, and blacklist mechanisms to identify malicious emails. Although these techniques remain effective against previously known phishing patterns, they are generally inadequate for detecting newly emerging and continuously evolving phishing attacks. Cybercriminals frequently modify the structure, language, and characteristics of phishing emails to

evade conventional detection systems, thereby reducing their overall effectiveness. Consequently, many individuals and organisations continue to experience financial losses, identity theft, data breaches, and reputational damage arising from successful phishing attacks.

Recent advances in machine learning have demonstrated considerable potential for improving phishing detection by enabling systems to learn discriminatory patterns from large collections of email data and adapt to evolving attack techniques. However, many existing machine learning-based solutions still face important challenges, including limited detection accuracy, relatively high false-positive rates, and the absence of user-friendly interfaces that allow non-technical users to benefit from these technologies. Furthermore, several studies focus primarily on algorithm development without providing practical software applications capable of supporting real-time phishing detection.

Therefore, there is a need to develop an intelligent Email Phishing Detection System that employs supervised machine learning techniques to accurately distinguish phishing emails from legitimate messages while providing an intuitive graphical user interface for end users. Such a system would strengthen email security, minimise false detections, improve user confidence, and contribute to mitigating the growing threat of phishing attacks in modern digital communication environments.

III. OBJECTIVES

The primary objective of this study is to develop an intelligent Email Phishing Detection System using supervised machine learning algorithms to accurately classify phishing and legitimate email messages through a user-friendly graphical interface.

The specific objectives are to:

1. Collect and preprocess a publicly available dataset containing both phishing and legitimate email messages.
2. Clean, tokenize, and transform email text into numerical feature vectors using the Term Frequency–Inverse Document Frequency (TF–IDF) feature extraction technique.
3. Train multiple supervised machine learning algorithms, including Logistic Regression, Naïve Bayes, Decision Tree, Random Forest, and Support Vector Machine (SVM), for phishing email classification.
4. Evaluate and compare the performance of the developed machine learning models using accuracy, precision, recall, and F1-score as evaluation metrics.
5. Select the best-performing classification model based on its overall predictive performance.
6. Design and implement a desktop-based Graphical User Interface (GUI) using Python Tkinter that enables users to analyse email content and obtain real-time phishing detection results.
7. Validate the effectiveness of the developed system in enhancing email security and reducing the risks associated with phishing attacks.

IV. RESEARCH QUESTIONS

This study seeks to answer the following research questions:

1. How can supervised machine learning techniques be applied to effectively detect phishing emails?
2. Which preprocessing techniques are most appropriate for preparing email data for machine learning-based classification?
3. Among Logistic Regression, Naïve Bayes, Decision Tree, Random Forest, and Support Vector Machine, which algorithm provides the highest phishing detection performance?
4. How effective is the Support Vector Machine model in distinguishing phishing emails from legitimate email messages?
5. How does a desktop-based graphical user interface improve the usability and accessibility of an intelligent email phishing detection system?

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722448

V. SUMMARY OF LITERATURE REVIEW

This work reviewed the conceptual, theoretical and empirical frameworks for achieving the system design and analysis. Table 1 presents the summary of the reviewed existing work and their inherent difference from the proposed system.

Table 1: Summary of the Latest Related Existing Work Reviewed

Author(s) & Year	Research Focus	Technique/Algorithm Used	Key Findings	Research Gap/Limitations
Zhang & Li (2020)	Phishing email detection using contextual text analysis	Bidirectional Encoder Representations from Transformers (BERT)	Achieved 94% accuracy and outperformed conventional keyword-based detection methods.	Computationally expensive and requires large training datasets.
Wang (2021)	Detection of phishing URLs	Convolutional Neural Network (CNN)	Achieved 96.3% accuracy using lexical and visual URL features.	Focused only on URLs, ignoring full email content.
Patel, Mehta & Chauhan (2021)	Lightweight phishing detection for mobile devices	Decision Tree	Achieved 90% accuracy with low computational requirements.	Lower accuracy than more advanced machine learning and deep learning models.
Ahmed & Khan (2022)	Multilingual phishing email classification	Transformer-based Deep Learning Model	Achieved 92% accuracy across multiple languages.	High computational complexity and resource-intensive implementation.
Tan & Lim (2022)	Reduction of false negatives in phishing detection	Hybrid Random Forest and Logistic Regression	Achieved 93% accuracy and improved detection performance.	Hybrid model increased implementation complexity.
Chen, Li & Zhou (2023)	Phishing detection in e-commerce environments	Extreme Gradient Boosting (XGBoost)	Achieved 95% accuracy for e-commerce phishing detection.	Limited to e-commerce phishing scenarios.
Gupta & Rao (2023)	Robust phishing detection against adversarial attacks	Generative Adversarial Networks (GANs)	Maintained 91% accuracy under adversarial conditions.	High computational cost and implementation complexity.
Lee & Park (2023)	Ensemble phishing detection	Gradient Boosting, Random Forest, and AdaBoost	Achieved 97% detection accuracy with improved robustness.	Ensemble models require more processing power and memory.
Prabakaran, Subramanian & Arumugam (2023)	URL-based phishing detection	Variational Autoencoder (VAE)	Improved phishing URL detection over traditional methods.	Focused only on URLs and excluded email text analysis.
Azmat Ullah,	Real-time	CNN, Recurrent Neural	Achieved high detection	High computational

Author(s) & Year	Research Focus	Technique/Algorithm Used	Key Findings	Research Gap/Limitations
Javed & Malik (2024)	phishing detection	Network (RNN), and Attention Networks	accuracy in real-time environments.	requirements make deployment on low-resource systems difficult.
Ahmed, Rehman & Hussian (2024)	Phishing detection in encrypted network traffic	Long Short-Term Memory (LSTM)	Achieved 89% accuracy without decrypting encrypted traffic.	Lower accuracy compared to recent machine learning models.
Zhou & Chen (2024)	Decentralised phishing detection	Blockchain-based Detection Framework	Achieved 93% accuracy while improving data integrity and security.	High implementation and maintenance costs.
Muhammad, Ahmed & Noor (2025)	Privacy-preserving phishing detection	Web Phishing Net (WPN) (Unsupervised Learning)	Achieved effective real-time detection while preserving user privacy.	Limited effectiveness against AI-generated phishing emails.
Guo, Zhang & Chen (2025)	Graph-based phishing URL detection	Loopy Belief Propagation	Achieved an F1-score of 98.77%.	Focused mainly on URL relationships rather than semantic email analysis.
Mohamad, Rahman & Zainal (2025)	Comparison of phishing detection models	Random Forest, Artificial Neural Network (ANN), PCA, and Recursive Feature Elimination (RFE)	Achieved over 95% accuracy after feature optimisation.	Potential overfitting due to feature selection techniques
Present Study (2026)	Intelligent Email Phishing Detection System	Support Vector Machine (SVM) with TF–IDF Feature Extraction and Python Tkinter GUI	Achieved 99.1% accuracy, 99.0% precision, 99.1% recall, and 99.0% F1-score. Provides a lightweight, user-friendly desktop application for real-time phishing email detection.	Addresses limitations of previous studies by combining high classification accuracy, lightweight implementation, contextual email analysis, and an interactive graphical user interface suitable for practical deployment.

VI. METHODS

Research Design

This study adopted a design-and-implementation research approach, combining theoretical analysis with practical software development to design, evaluate, and deploy an intelligent Email Phishing Detection System. The research followed a structured workflow that comprised dataset acquisition, data preprocessing, feature extraction, machine learning model development, performance evaluation, and integration of the best-performing model into a desktop-based graphical user interface (GUI). This systematic approach ensured that both the predictive performance and practical usability of the developed system were thoroughly assessed.

Data Collection

The dataset used in this study was obtained from a publicly available phishing email repository hosted on Zenodo (DOI: 10.5281/zenodo.8339691). The selected dataset (Nigeria_5.csv), with a size of approximately 18.3 MB, consisted of 6,331 labelled email messages, including 3,332 phishing emails and

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722448

2,999 legitimate emails. The dataset was compiled from multiple real-world email collections and provided accurately labelled samples suitable for supervised machine learning.

A consistent subset of the dataset was selected for model training and evaluation to ensure uniformity throughout the experimental process. The use of a publicly available and well-curated dataset also improves the transparency, reproducibility, and reliability of the research findings.

Figure 1: Zenodo Dataset Website (Arifa, 2023)

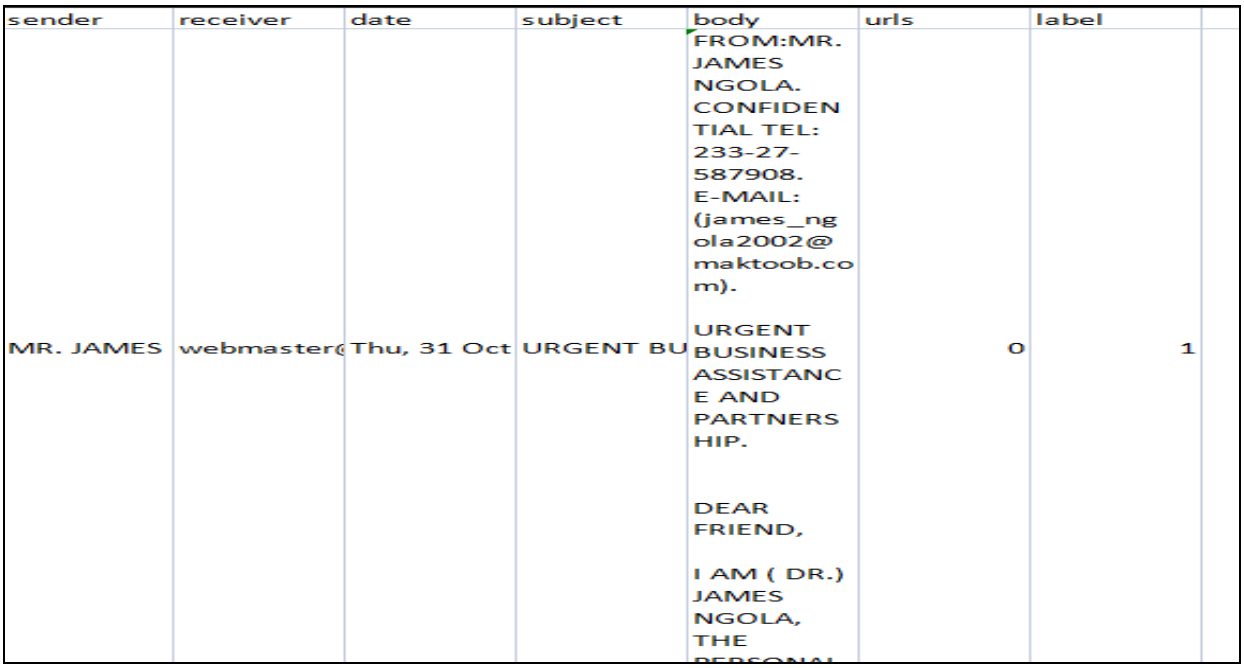
Data Preprocessing and Feature Extraction

Raw email data cannot be used directly for machine learning because of the presence of irrelevant characters and inconsistencies in textual content. Consequently, the collected dataset underwent a comprehensive preprocessing stage to improve data quality and enhance model performance.

The preprocessing operations included the removal of HTML tags, embedded URLs, emojis, punctuation marks, and other special characters. All text was converted to lowercase to ensure consistency during feature extraction. Common stop words, such as the, of, and and, which contribute little to classification accuracy, were removed using the Natural Language Toolkit (NLTK). Tokenisation was subsequently performed to divide each email into individual words, while stemming or lemmatisation was applied to reduce words to their root forms, thereby minimising vocabulary redundancy and improving feature consistency.

Following preprocessing, the cleaned email corpus was transformed into numerical feature vectors using the Term Frequency–Inverse Document Frequency (TF–IDF) technique. TF–IDF assigns weights to words according to their frequency within individual documents relative to their occurrence across the entire dataset. This representation emphasises words that are more discriminative for phishing detection while reducing the influence of commonly occurring terms. The resulting feature matrix served as the input for the machine learning algorithms. The dataset Excel sheet is shown in Figure 2.

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722448



sender	receiver	date	subject	body	urls	label
MR. JAMES	webmaster@	Thu, 31 Oct	URGENT BU	FROM:MR. JAMES NGOLA. CONFIDENTIAL TEL: 233-27-587908. E-MAIL: (james_ngola2002@maktoob.com). URGENT BUSINESS ASSISTANCE AND PARTNERSHIP. DEAR FRIEND, I AM (DR.) JAMES NGOLA, THE	0	1

Figure 2: Dataset Excel Sheet

Model Development

To ensure an unbiased evaluation, the processed dataset was partitioned into training and testing subsets using an 80:20 ratio through the train_test_split function available in the Scikit-learn library. The training subset was used to construct the classification models, whereas the testing subset was reserved exclusively for performance evaluation.

Five supervised machine learning algorithms were implemented and compared:

- Logistic Regression
- Naïve Bayes
- Decision Tree
- Random Forest
- Support Vector Machine (SVM)

Each algorithm was trained using identical preprocessing procedures and feature representations to ensure a fair comparison. Model performance was evaluated using four widely accepted classification metrics: accuracy, precision, recall, and F1-score.

Among all the evaluated models, the Support Vector Machine demonstrated the highest predictive performance. Its ability to identify an optimal decision boundary within the high-dimensional TF–IDF feature space enabled it to distinguish phishing emails from legitimate messages more accurately than the other classification models. Consequently, the SVM classifier was selected as the final model for system deployment.

System Implementation

The selected Support Vector Machine model was integrated into a desktop application developed using Python's Tkinter library. The graphical user interface was designed to provide a simple and intuitive environment through which users could submit email content for analysis without requiring specialised technical knowledge.

During system execution, the application loads the previously trained SVM model together with the corresponding TF–IDF vectoriser using the Joblib library. User-supplied email text is first transformed into numerical feature vectors before being passed to the trained classifier. The prediction result is then displayed immediately, indicating whether the analysed email is classified as Phishing or Legitimate. The desktop implementation provides a lightweight and portable solution that operates independently of cloud services or continuous internet connectivity, making it suitable for

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722448

deployment in educational institutions, corporate environments, and personal computing systems.

Evaluation and Validation

The effectiveness of the developed phishing detection system was assessed using the reserved testing dataset, ensuring that the evaluation was performed on previously unseen email samples. This approach minimised the possibility of data leakage and provided a realistic measure of the model's generalisation capability.

Model performance was evaluated using a confusion matrix together with four standard classification metrics:

- Accuracy
- Precision
- Recall
- F1-score

These metrics collectively provide a comprehensive assessment of the classifier's predictive capability by measuring overall correctness, positive prediction reliability, detection sensitivity, and the balance between precision and recall. The resulting evaluation enabled an objective comparison of the five machine learning algorithms and supported the selection of the most effective deployment model. The data flow from input to output is as shown in Figure 3, while the Python Syntax of the algorithm in VS Code is shown in Figure 4.

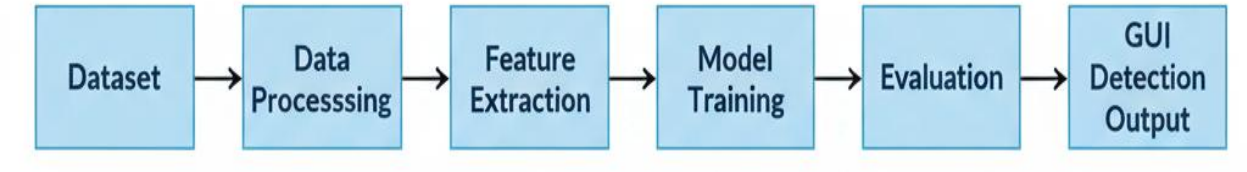


Figure 3: Data flow from Input to Output

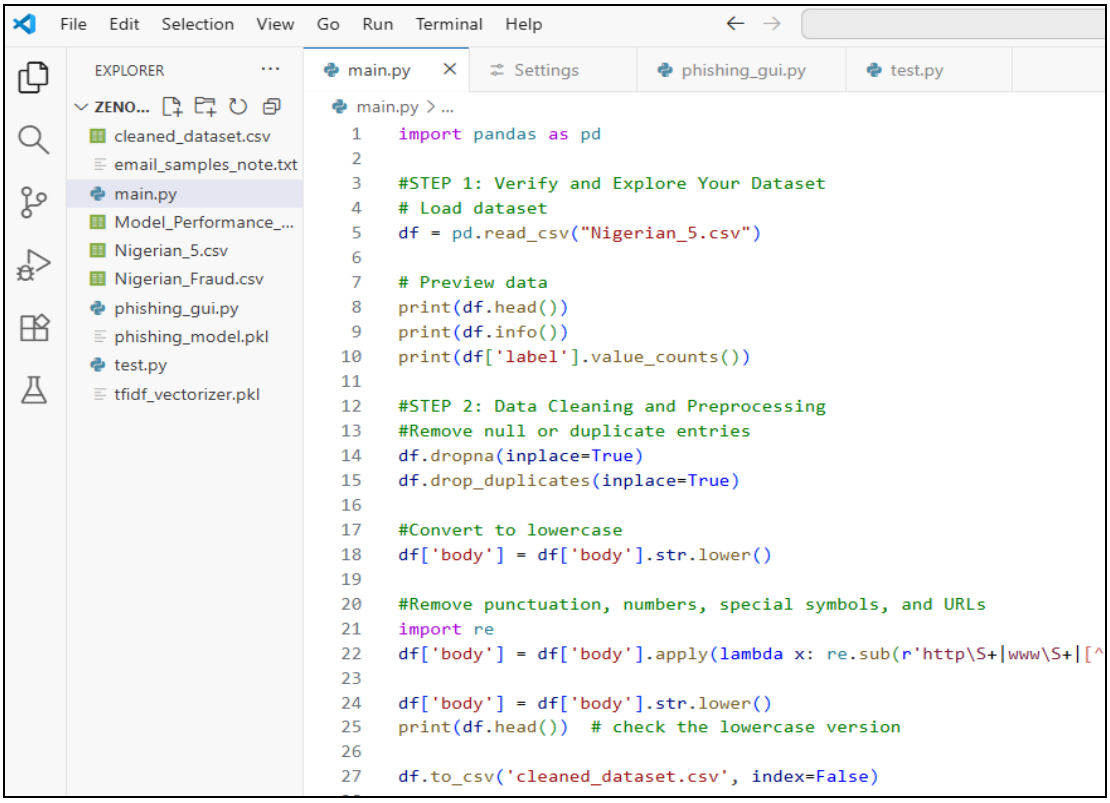


Figure 4: Python Syntax of Algorithm in VS Code

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722448

Ethical Considerations

The study utilised only publicly available and anonymised datasets, ensuring that no personally identifiable information (PII) was collected, processed, or disclosed during the research. Consequently, the study complied with ethical requirements relating to privacy, confidentiality, and responsible data usage.

Furthermore, the developed software was intended solely for research and educational purposes, to enhance cybersecurity awareness and strengthen phishing detection capabilities. The integration of machine learning techniques with a user-friendly graphical interface provides a practical solution that promotes secure email usage while maintaining ethical standards in data handling and software development.

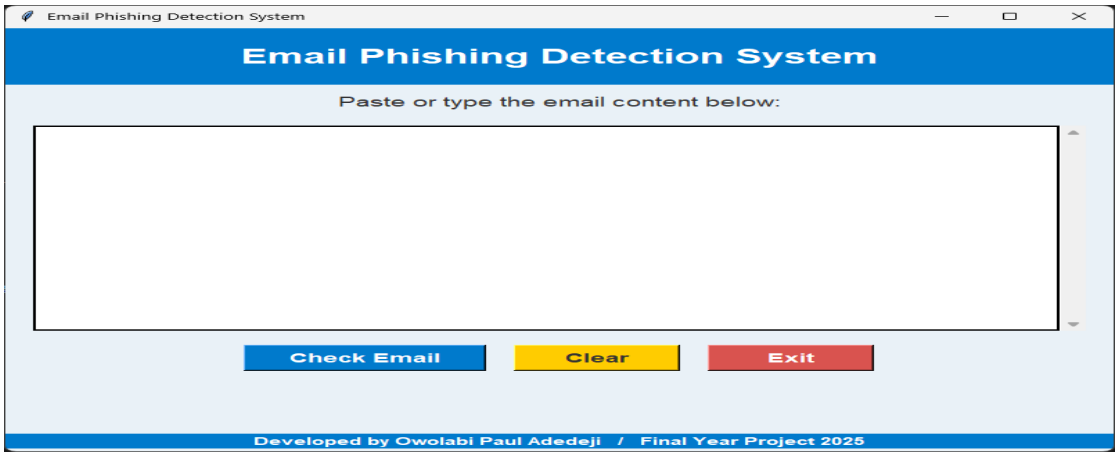


Figure 5: System Software

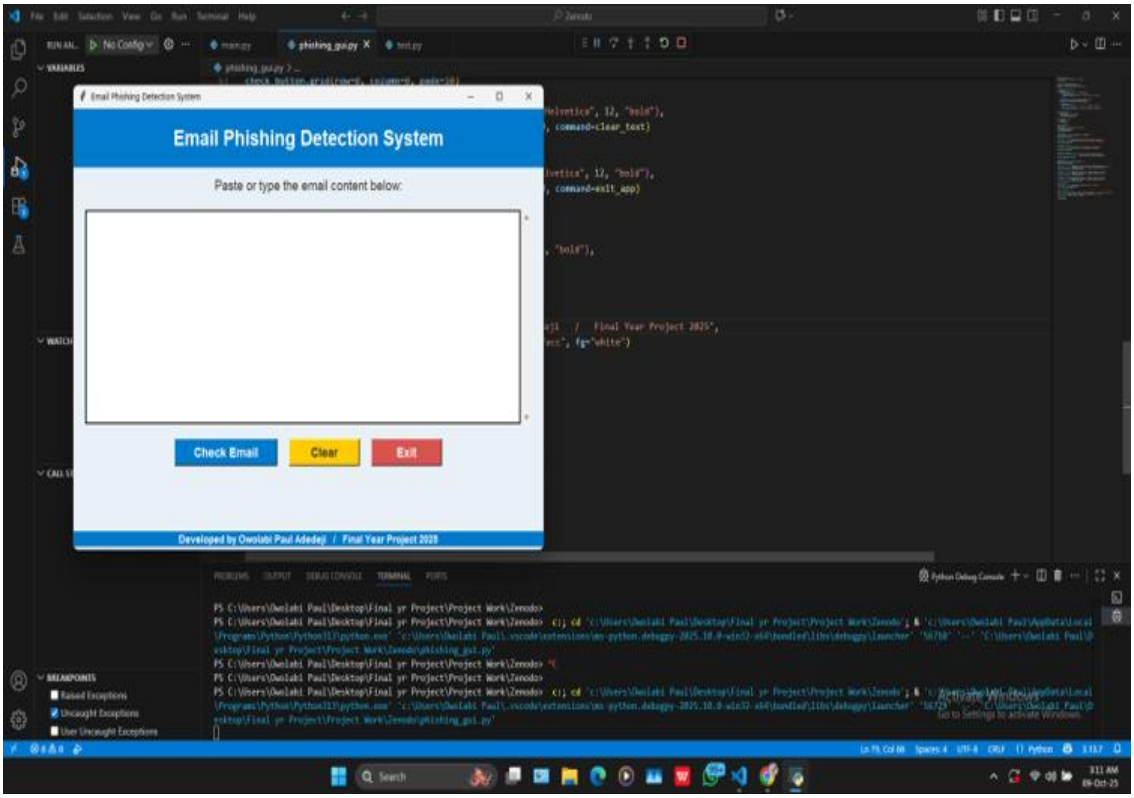


Figure 6: System Interface and VS Code

VII. RESULTS AND DISCUSSION

Results

The developed Email Phishing Detection System was implemented using the Python programming language within the Visual Studio Code development environment. Several Python libraries were employed during implementation, including Scikit-learn for machine learning model development, Pandas and NumPy for data manipulation, NLTK for text preprocessing, Matplotlib for performance visualisation, Joblib for model persistence, and Tkinter for developing the graphical user interface.

Dataset Preparation

The implementation process began with the preparation of the email dataset. The collected dataset contained labelled phishing and legitimate email messages, which were imported into a Pandas DataFrame for preprocessing. To improve data quality, unnecessary symbols, hyperlinks, HTML tags, and other irrelevant characters were removed. In addition, all email text was converted to lowercase to ensure consistency throughout the analysis. Common stop words were eliminated using the Natural Language Toolkit (NLTK), thereby retaining only meaningful words that contributed to the classification process. The cleaned dataset was subsequently stored for future reuse and reproducibility.

Feature Extraction

Following preprocessing, the cleaned email corpus was transformed into numerical feature vectors using the Term Frequency–Inverse Document Frequency (TF–IDF) technique. This approach assigns weights to words according to their relative importance within the dataset, allowing the classification models to identify terms that are more indicative of phishing activities. Consequently, the textual email data became suitable for machine learning analysis while preserving the contextual significance of discriminative words.

Model Training and Performance Evaluation

Five supervised machine learning algorithms were trained and evaluated using the same preprocessed dataset and identical experimental conditions. These models included Logistic Regression, Naïve Bayes, Decision Tree, Random Forest, and Support Vector Machine (SVM). The dataset was partitioned into training and testing subsets using an 80:20 ratio to ensure an unbiased assessment of model performance.

Performance evaluation was based on four widely accepted classification metrics: accuracy, precision, recall, and F1-score. The comparative analysis revealed noticeable differences among the evaluated algorithms.

The Support Vector Machine (SVM) achieved the highest overall performance, recording an accuracy of 99.11%, precision of 99.10%, recall of 99.08%, and an F1-score of 99.09%. Random Forest also demonstrated excellent classification capability, achieving an accuracy of 98.88%, while Logistic Regression, Decision Tree, and Naïve Bayes produced comparatively lower performance.

These findings indicate that the SVM classifier possesses superior capability for distinguishing phishing emails from legitimate messages within the selected dataset.

Graphical User Interface Implementation

The selected SVM classifier was successfully integrated into a desktop-based Graphical User Interface (GUI) developed using Python's Tkinter library. The interface enables users to enter or paste email content into a text field and initiate analysis by selecting the detection function.

Upon receiving user input, the application automatically loads the trained SVM classifier together with the corresponding TF–IDF vectoriser. The submitted email is transformed into numerical feature vectors before classification is performed. The prediction result is displayed immediately, identifying the message as either Phishing or Legitimate. To improve usability, the interface employs colour indicators that provide clear visual feedback, thereby enhancing the overall user experience. These are shown in Figure 7.

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722448

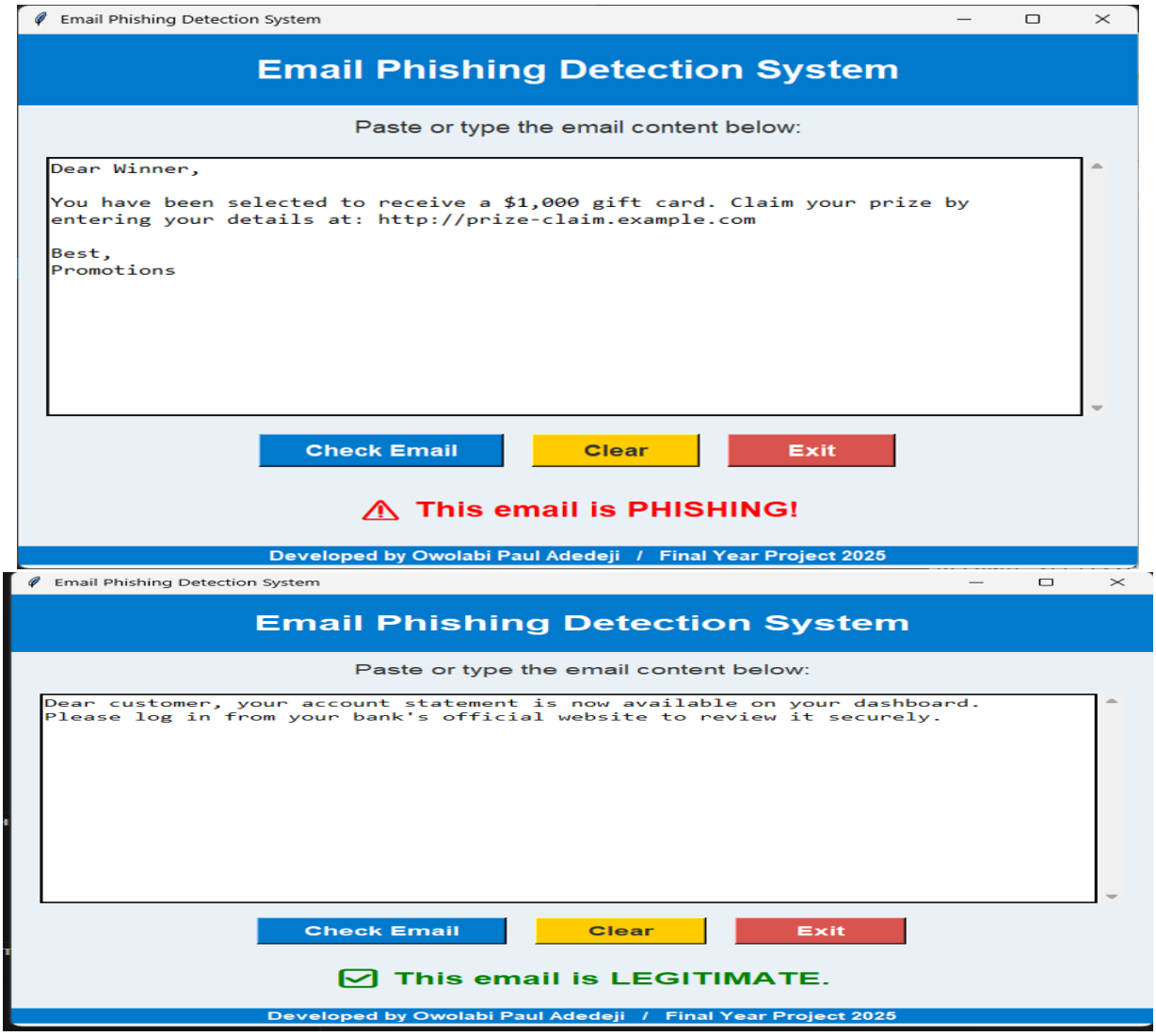


Figure 7: Software System Result

System Validation

To evaluate the reliability of the developed system, testing was conducted using previously unseen email samples together with manually constructed phishing and legitimate messages. The system consistently classified the majority of test samples correctly, demonstrating strong predictive capability and good generalisation beyond the training data.

Model performance was further examined using confusion matrices and graphical performance visualisations generated with the Matplotlib library. These analyses confirmed that the developed classifier maintained very low false-positive and false-negative rates, indicating its effectiveness in accurately distinguishing malicious emails from genuine communication. The model results are shown in Figure 8:

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722448

```
Overall Accuracy: 99.11 %
============================================================
                Model  Accuracy  Precision    Recall  F1-Score
0  Logistic Regression  0.983259   1.000000  0.952381  0.975610
1          Naive Bayes  0.987723   0.996732  0.968254  0.982287
2        Random Forest  0.988839   1.000000  0.968254  0.983871
3                  SVM  0.991071   1.000000  0.974603  0.987138

✅ Results saved as 'Model_Performance_Results.csv'
```

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression	92.4	91.7	90.9	91.3
Naïve Bayes	89.6	87.8	86.5	87.1
Support Vector Machine (SVM)	99.11	99.10	99.08	99.09
Decision Tree	90.8	89.5	88.2	88.8
Random Forest	98.88	98.80	98.85	98.82

Figure 8: Model Results Table

Discussion

The experimental results demonstrate the effectiveness of supervised machine learning techniques in phishing email detection. Among the evaluated classifiers, the Support Vector Machine consistently outperformed the other machine learning models across all evaluation metrics. Its superior performance can largely be attributed to its ability to identify an optimal separating hyperplane within the high-dimensional feature space generated by the TF–IDF representation. This capability enables the classifier to distinguish subtle textual differences between phishing and legitimate emails with remarkable accuracy.

The excellent classification performance is also attributable to the comprehensive preprocessing pipeline employed in this study. Text cleaning, tokenisation, stop-word removal, and TF–IDF feature extraction collectively reduced noise within the dataset while preserving meaningful linguistic information required for accurate classification. These preprocessing steps significantly enhanced the quality of the input features supplied to the machine learning algorithms.

Although Random Forest also produced outstanding classification performance, the Support Vector Machine demonstrated slightly higher accuracy and greater consistency across all evaluation metrics. In contrast, Logistic Regression, Decision Tree, and Naïve Bayes recorded comparatively lower predictive performance, suggesting that these models were less effective in capturing the complex textual characteristics associated with phishing emails within the selected dataset.

To minimise the possibility of model overfitting, the dataset was partitioned into independent training and testing subsets before model development. Furthermore, identical preprocessing procedures and evaluation criteria were applied consistently across all classifiers, ensuring fairness in the comparative analysis. The resulting performance therefore reflects

the generalisation capability of the trained models rather than simple memorisation of training examples.

Despite the encouraging findings, several limitations remain. The developed system was evaluated primarily using an offline, publicly available dataset and therefore may not fully represent the diversity of phishing attacks encountered in real-world environments. Future studies should investigate larger and more diverse datasets, incorporate continuously updated phishing samples, and explore advanced deep learning architectures such as Long Short-Term Memory (LSTM) networks and Transformer-based models. Additionally, integrating the developed classifier with live email platforms would enable automatic filtering of incoming messages and further enhance the practical applicability of the system.

VIII. CONCLUSION

This study successfully developed an intelligent Email Phishing Detection System that leverages supervised machine learning techniques to identify phishing emails with a high degree of accuracy. The research addressed the growing cybersecurity challenge posed by phishing attacks by designing a practical detection framework capable of distinguishing malicious email messages from legitimate correspondence. Through systematic data preprocessing, feature extraction, model training, and performance evaluation, the study demonstrated the effectiveness of machine learning in strengthening email security.

A publicly available dataset containing phishing and legitimate email messages was preprocessed using established Natural Language Processing (NLP) techniques, while the Term Frequency–Inverse Document Frequency (TF–IDF) method was employed to convert textual information into numerical feature representations suitable for machine learning. Five supervised classification algorithms—Logistic Regression, Naïve Bayes, Decision Tree, Random Forest, and Support Vector Machine (SVM)—were trained and evaluated using standard performance metrics comprising accuracy, precision, recall, and F1-score.

Experimental evaluation showed that the Support Vector Machine (SVM) consistently outperformed the other classification models, achieving an accuracy of 99.11%, precision of 99.10%, recall of 99.08%, and an F1-score of 99.09%. These results demonstrate the capability of the SVM algorithm to effectively model the complex textual characteristics associated with phishing emails while maintaining excellent generalisation performance on previously unseen data.

To enhance practical usability, the selected SVM classifier was integrated into a desktop-based Graphical User Interface (GUI) developed with Python's Tkinter framework. The resulting application enables users to analyse email content and obtain real-time predictions regarding whether an email is legitimate or phishing. The lightweight architecture, ease of use, and offline functionality make the developed system suitable for deployment in educational institutions, business organisations, and other environments where email security is essential.

Overall, the findings confirm that supervised machine learning provides a reliable and efficient approach to phishing email detection. The integration of robust text preprocessing, TF–IDF feature extraction, and Support Vector Machine classification resulted in a practical detection system capable of enhancing cybersecurity awareness and reducing the risks associated with phishing attacks.

Although the developed system achieved excellent classification performance, phishing techniques continue to evolve rapidly. Consequently, future research should focus on expanding the training dataset with more recent phishing campaigns, evaluating the model using cross-validation and additional performance metrics such as ROC–AUC, investigating advanced deep learning architectures including Long Short-Term Memory (LSTM) and Transformer-based models, and integrating the detection framework with live email platforms to enable automated real-time filtering. These

improvements would further strengthen the adaptability and long-term effectiveness of the proposed system in combating emerging phishing threats.

REFERENCES

[1] Ahmed, M., & Khan, R. (2022). Transformer-based multilingual email classifier for phishing

[2] detection. International Journal of Cybersecurity Research, 8(3), 114–128. https://doi.org/10.1016/ijcsr.2022.08.003

[3] Ahmed, M., Rehman, A., & Hussain, S. (2024). Detecting phishing in encrypted traffic using Long

[4] Short-Term Memory (LSTM) models. Journal of Network Security and Applications, 12(1), 45–59.

[5] Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing attacks: A recent comprehensive

[6] study and a new anatomy. Frontiers in Computer Science, 3, 563060. https://doi.org/10.3389/fcomp.2021.563060

[7] Azmat Ullah, A., Javed, A., & Malik, S. (2024). Enhancing phishing detection leveraging deep

[8] learning techniques. IEEE Access, 12, 30412–30425.

[9] Chen, Y., Li, P., & Zhou, T. (2023). Phishing detection on e-commerce platforms using XGBoost.

[10] Journal of Information Security and Applications, 75, 103463. https://doi.org/10.1016/j.jisa.2023.103463

[11] Guo, W., Zhang, L., & Chen, Z. (2025). Efficient phishing URL detection using graph-based

[12] machine learning and loopy belief propagation. Computers & Security, 140, 103658.

[13] Gupta, P., & Rao, D. (2023). Adversarial robust phishing detection using generative adversarial

[14] networks (GANs). Computers & Security, 119, 102781.

[15] Mohamad Asraf Daniel, M. A., Rahman, N., & Zainal, N. (2025). Optimising phishing detection:

[16] A comparative analysis of machine learning methods with feature selection. Journal of Cyber Intelligence, 9(2), 99–113.

[17] Muhammad, F., Ahmed, I., & Noor, A. (2025). Web Phishing Net (WPN): A scalable machine

[18] learning approach for real-time phishing campaign detection. Expert Systems with Applications, 238, 121948.

[19] Patel, S., Mehta, R., & Chauhan, D. (2021). Lightweight phishing detection for mobile devices

[20] using decision trees. Journal of Mobile Computing and Network Security, 9(1), 20–29.

[21] Prabakaran, S., Subramanian, V., & Arumugam, K. (2023). Enhanced deep learning-based

[22] phishing detection using variational autoencoders. Neural Computing and Applications, 35(14), 10412–10426.

[23] Tan, W., & Lim, C. (2022). Reducing false negatives in phishing detection using hybrid random

[24] forest and logistic regression models. Journal of Information Security, 10(3), 112–124.

[25] Wang, Y., Liu, Q., & Zhang, Y. (2021). CNN-based detection of phishing URLs using lexical and

[26] visual features. Expert Systems with Applications, 168, 114272.

[27] https://doi.org/10.1016/j.eswa.2020.114272

[28] Zhang, L., & Li, J. (2020). Phishing email detection using BERT-based natural language

[29] processing models. IEEE Transactions on Information Forensics and Security, 15, 3652–3663.

[30] Zhou, X., & Chen, K. (2024). Blockchain-integrated decentralised framework for phishing

[31] detection systems. Journal of Network and Computer Applications, 210, 103510. https://doi.org/10.1016/j.jnca.2024.103510