Concept Paper

RealPhish: An Algorithm for Real-Time Email Phishing Detection

Devendra Chapagain *, Naresh Kshetri, Bishnu Bhusal, Pradip Subedi

Posted Date: 5 November 2025

doi: 10.20944/preprints202511.0218.v1

Keywords: phishing; email security; defense; cybersecurity; machine learning; real time

 doi:10.20944/preprints202511.0218.v1

Concept Paper

RealPhish: An Algorithm for Real-Time Email Phishing Detection

Devendra Chapagain [1,*], Naresh Kshetri [2], Bishnu Bhusal [3] and Pradip Subedi [4]

[1] Birendra Multiple Campus, Computer Department
[2] Department of Cybersecurity, Golisano College of Computing and Information Sciences
[3] University of Missouri, Columbia, Missouri, USA
[4] Huazhong University of Science and Technology (HUST), Wuahn, China
* Correspondence: devendra.chapagain@bimc.tu.edu.np

Abstract

In this era of technology, phishing emails remain a critical cybersecurity threat, exploiting human vulnerabilities to compromise sensitive data and systems. Traditional detection methods, such as blacklists and static heuristics, often fail to keep pace with the evolving sophistication of phishing tactics. We propose RealPhish, a real-time phishing detection algorithm combining machine learning, Natural Language Processing (NLP), and rule-based heuristics to identify malicious emails with high precision. RealPhish analyzes both static email features and simulated user interaction data to enhance detection accuracy. Using a publicly available phishing email dataset and synthetically generated behavioral data, the algorithm achieves a detection accuracy of **95%**, with a precision of **96%** and recall of **89%**, outperforming baseline models. The system also includes a rule-based override layer for known threats and provides interpretable outputs for transparency. RealPhish demonstrates strong potential for deployment in real-world email security platforms, offering a scalable and adaptive solution to combat phishing attacks in real time.

Keywords: phishing; email security; defense; cybersecurity; machine learning; real time

1. Introduction

The human element is the weakest link in cyber security, and that is what social engineering attacks target [1]. The universal threat of phishing attacks necessitates continuous advancements in detection methodologies, as traditional filtering techniques like heuristics and blacklisting have demonstrated limited effectiveness against increasingly sophisticated phishing campaigns [2] . Phishing, a deceptive practice of acquiring sensitive information through masquerading as a trustworthy entity in electronic communication, continues to be a significant cybersecurity concern for individuals and organizations.

Phishing attacks are a dominant form of cybercrime, where attackers deceive individuals into providing sensitive information by masquerading as trustworthy entities. Phishing attacks typically occur via email, deceiving users into clicking malicious links or disclosing sensitive information. The increasing sophistication of phishing techniques necessitates advanced detection mechanisms to protect users and organizations.

Numerous approaches have been proposed for phishing email detection, ranging from traditional rule-based systems to advanced machine learning models. Early methods relied on predefined rules and blacklists to identify phishing emails. While effective to some extent, these methods struggled to keep up with the evolving tactics of attackers.

Machine learning models, such as logistic regression, random forests, and neural networks, have shown promise in detecting phishing emails by learning patterns from labeled datasets. However, these models often require extensive training data and may not generalize well to new phishing

 doi:10.20944/preprints202511.0218.v1

techniques. RealPhish addresses these challenges by integrating machine learning with rule-based heuristics, providing a comprehensive approach to phishing detection.

To combat this ever-evolving threat, researchers have explored diverse approaches, including machine learning-based solutions that can adapt to new phishing tactics [3,4]. Machine learning has emerged as a potent tool for phishing detection, offering the ability to analyze patterns and anomalies in email content and user behavior [5]. These machine learning models require content samples for training to effectively identify threats [6]. Machine learning methods have proven effective at detecting patterns in data, making it possible to recognize common phishing traits and identify phishing websites [7].

2. Research Gap: Limitations of Current Anti-Phishing Systems

The detection of malicious URLs is crucial in mitigating phishing attacks, as many phishing attempts rely on directing victims to fraudulent websites [5]. Real-time phishing detection is particularly critical because it can prevent users from falling victim to attacks before they have a chance to be harmed [8]. Many organizations and researchers have adopted user training strategies to improve cybersecurity awareness and understanding; however, users frequently struggle to retain security awareness knowledge and information [9]. Therefore, it is advantageous to offer email users help in spotting phishing emails. One promising avenue for real-time detection involves analyzing user interactions with emails to identify anomalies that may indicate phishing attempts.

Despite significant advancements in phishing detection technologies, current anti-phishing systems face critical challenges in effectively responding to the rapidly evolving tactics used by cyber attackers. One of the primary limitations is the overreliance on static and reactive detection methods. Traditional approaches, such as blacklists, heuristic analysis, and even some machine learning techniques, often struggle to adapt to new and sophisticated phishing strategies. These methods, while useful, are not foolproof and tend to fall short in real-time scenarios, as highlighted by [10,11].

Another notable limitation lies in the narrow focus of many machine learning approaches. While ML-based systems have demonstrated superior accuracy compared to traditional strategies [12], they predominantly concentrate on analyzing content features—such as lexical properties of URLs, host data, or static email attributes. Although this has enabled detection of many phishing attempts [13,14], it does not fully account for the contextual and behavioral aspects of user interaction, which can be critical in identifying more subtle or adaptive attacks.

Furthermore, there is a clear lack of emphasis on real-time behavioral detection in existing systems. The dynamic and deceptive nature of phishing requires proactive, continuously learning models. However, most current techniques depend on periodically updated blacklists or manually crafted heuristics, which are easily bypassed. While numerous machine learning algorithms have been explored [15], they still rely heavily on static data and offer limited capabilities in detecting behavior-driven anomalies during user interaction. The performance of user interaction-based approaches was reported inconsistently in the literature[16]. While the study explored how intervention and incentive mechanisms influence user behavior during phishing attacks, it did not provide specific accuracy metrics for the predictive model.

To address these gaps, a more dynamic and adaptive approach is necessary. While machine learning continues to outperform many traditional methods, current system remains reactive and dependent on static indicators like URLs or email contents. To bridge this gap, we propose RealPhish, a novel system that integrates machine learning. This approach aims to add a real-time, enhancing the system's ability to respond to evolving phishing threats with greater precision and agility.

Table 1. Comparative Analysis of Related Studies.

Study	Study Focus	ML technique and Detection	Performance matrix

 doi:10.20944/preprints202511.0218.v1

[17]	Email summarization and phishing identification	Transformer-based models (T5, XL-Net, BERT) Content and emotion analysis	No mention found
[18]	Anomalous email detection	Interactive machine learning (iML) models Active learning, visual analysis	No mention found
[19]	Phishing Email Detection	Random Forest Classifier, Support Vector Machine Email content analysis	Accuracy, confusion matrix, classification report
[16]	User behavior analysis in phishing attacks	Decision Tree-J48, Naive Bayes, Support Vector Machine (SVM), Multilayer Perceptron User behavior analysis, email classification	Accuracy Score
[20]	Phishing email detection	Locally-deep SVM, SVM, Boosted decision tree, Logistic regression, Averaged perceptron, Neural network, Decision Forest, Email content analysis	Accuracy rates
[21]	Phishing susceptibility prediction	TransMLP Link and TransMLP Hybrid Eye-tracking data analysis	Accuracy rates

3. Methodology

This research focuses on the development and evaluation of the RealPhish algorithm for real-time email phishing detection. The methodology encompasses algorithm design and implementation, dataset preparation, data preprocessing, feature engineering, model training, ensemble classification, and rule-based overrides.

Dataset Preparation

A publicly available email dataset was sourced from Kaggle [22], one of the largest repositories for open datasets. The dataset contains approximately 17,000 emails, comprising both phishing and legitimate messages. Each entry includes fields such as subject, body, sender, receiver and URLs. It was used to train and evaluate the static analysis components of the RealPhish algorithm. The dataset comprises a diverse collection of both legitimate and phishing emails, enabling robust model development and testing.

Data Preprocessing

The preprocessing pipeline consists of three main stages: text cleaning, feature engineering, and vectorization, followed by ensemble classification with rule-based overrides.

Preprocessing included text cleaning (removal of HTML tags and special characters), lowercasing, tokenization, and normalization. Additional features were engineered to enrich the dataset, including:

Text Cleaning and Normalization

Raw email content underwent standard preprocessing to prepare it for analysis:

- HTML tag removal: Stripped all HTML markup and formatting

 doi:10.20944/preprints202511.0218.v1

- Special character filtering: Removed non-alphanumeric characters except basic punctuation
- Text normalization: Applied lowercasing, tokenization, and basic normalization

Feature Engineering

Four additional binary and numerical features were extracted to capture phishing indicators:

- URL Count (n_urls): Total number of hyperlinks present in the email content.
- IP-based URL Detection (ip_flag): Binary indicator for presence of IP-formatted URLs:

  ip_flag = { 1, if any URL in U contains IP address format
  { 0, otherwise
- Urgency Score: Binary indicator for presence of urgency-related keywords:

  UrgencyScore(T) = { 1, if $|W \cap K| > 0$
  { 0, otherwise

Where:

- $W = \{w_1, w_2, ..., w_n\}$ represents the tokenized words in email text T

- K = {urgent, immediate, verify, suspended, expire, deadline, ...} is the predefined urgency keyword set

- $|W \cap K|$ denotes the count of urgency keywords found in the email

Blacklist Check (blacklist_flag): Binary Indicator for Sender Domain Reputation

blacklist_flag = { 1, if sender_domain ∈ Blacklist
{ 0, otherwise

- URL count: Number of links in the email.
- IP-based URL detection: Binary indicator for presence of IP-formatted URLs.
- Urgency score: Presence of urgency-related keywords, defined as:

$$\boldsymbol{UrgencyScore(T) = \begin{cases} 1, if\{\omega_i\} \cap u \neq \emptyset \\ 0, otherwise \end{cases}}$$

Where $\{\omega_i\}$ is the set of words in the email body and u is the set of urgency-related keywords.

- Blacklist check: Binary indicator for whether the sender's domain appears in a predefined blacklist.

The cleaned email text was vectorized using Term Frequency–Inverse Document Frequency (TF-IDF), and the resulting vectors were concatenated with the engineered features to form the final feature set. RealPhish employs an ensemble of three machine learning models—Logistic Regression, Random Forest, and Multinomial Naive Bayes. The ensemble prediction is computed using hard voting:

$$\boldsymbol{\hat{y} = mode(\hat{y}LR, \hat{y}RF, \hat{y}NB)}$$

Where $\hat{y}LR, \hat{y}RF, \hat{y}NB$ are the predictions from each base model.

To enhance robustness, a rule-based override layer is applied. If an email meets specific threat criteria—such as being from a blacklisted domain or containing both urgency cues and IP-based URLs—the final prediction is overridden as phishing:

$$\boldsymbol{\hat{y}_{final} = \begin{cases} 1, if\ Blacklisted\ (s) = 1\ or\ (UrgencyScore(T) = 1 \wedge HasIP(u) = 1) \\ \hat{y}_{ensemblr,}\ Otherwise \end{cases}}$$

Where, s is the sender domain, T is the email text and U is the set of URLs.

Experimental Setup

The algorithm was implemented and executed in Google Colab, a cloud-based Python development environment. Key libraries used include scikit-learn for model training and evaluation, pandas for data manipulation, and NumPy for numerical operations.

Performance Evaluation and Analysis

 doi:10.20944/preprints202511.0218.v1

Model performance was assessed using standard classification metrics including Precision, Recall, F1-Score, and Accuracy. These metrics were computed on a held-out test set to evaluate the effectiveness of both the ensemble model and the rule-based override mechanism.

4. The RealPhish Algorithm

RealPhish is designed to classify emails as either "Phishing Email" or "Safe Email" in real-time. The algorithm leverages NLP, machine learning, and rule-based heuristics to achieve high accuracy and reliability.

Key Components

1. **Input Layer**

Email Content: Subject, body, and metadata (sender, links, attachments).
Real-Time Trigger: Activated upon email receipt or user interaction.

2. **Preprocessing Module**

Text Cleaning: Remove HTML tags, special characters, and stop words.
Tokenization: Break text into words or phrases.
Normalization: Lowercasing, stemming, or lemmatization.

3. **Feature Extraction**

TF-IDF Vectorization: Capture important terms.
URL Analysis: Count of URLs, presence of IP-based URLs, domain reputation check (via API).
Sender Analysis: Domain mismatch, SPF/DKIM validation.
Linguistic Features: Urgency cues (e.g., "verify now"), threat language (e.g., "account suspended").
Attachment Analysis: File type and extension, presence of macros or scripts.

4. **Classification Engine**

Logistic Regression: For linear patterns.
Random Forest: For non-linear relationships.
Naive Bayes: For probabilistic text classification.
Voting Classifier: Aggregates predictions from all models.

5. **Rule-Based Override Layer**

Overrides model output if:

- Email contains blacklisted domains or IPs.
- Known phishing patterns are matched (regex or heuristics).
- Fails SPF/DKIM and contains suspicious links.

6. **Output Layer**

Label: Phishing Email or Safe Email.
Confidence Score: Probability from ensemble.
Explanation: Key features that influenced the decision (for transparency).

7. **Real-Time Deployment Considerations**

Lightweight model for fast inference.
Caching of known safe/phishing domains.
API integration for domain reputation and threat intelligence.
Algorithm: RealPhish Pseudo code

```
BEGIN RealPhish
  INPUT: Incoming Email (subject, body, sender, links, attachments)
  Step 1: Preprocessing
      CLEAN email text (remove HTML, special characters)
```

 doi:10.20944/preprints202511.0218.v1

TOKENIZE text into words
NORMALIZE text (lowercase, stemming/lemmatization)
Step 2: Feature Extraction
EXTRACT TF-IDF features from email text
ANALYZE URLs:
- COUNT number of links
- CHECK for IP-based URLs
- QUERY domain reputation API
ANALYZE sender:
- CHECK domain mismatch
- VALIDATE SPF/DKIM
DETECT linguistic cues:
- SEARCH for urgency/threat keywords
ANALYZE attachments:
- CHECK file types and presence of macros
Step 3: Classification
PREDICT using ensemble of:
- Logistic Regression
- Random Forest
- Naive Bayes
COMBINE predictions using majority vote
Step 4: Rule-Based Override
IF email contains blacklisted domain OR
matches known phishing pattern OR
fails SPF/DKIM AND contains suspicious links THEN
SET label = “Phishing Email”
ELSE
USE ensemble prediction
Step 5: Output
RETURN label (“Phishing Email” or “Safe Email”)
RETURN confidence score
RETURN explanation of key features
END RealPhish

 doi:10.20944/preprints202511.0218.v1

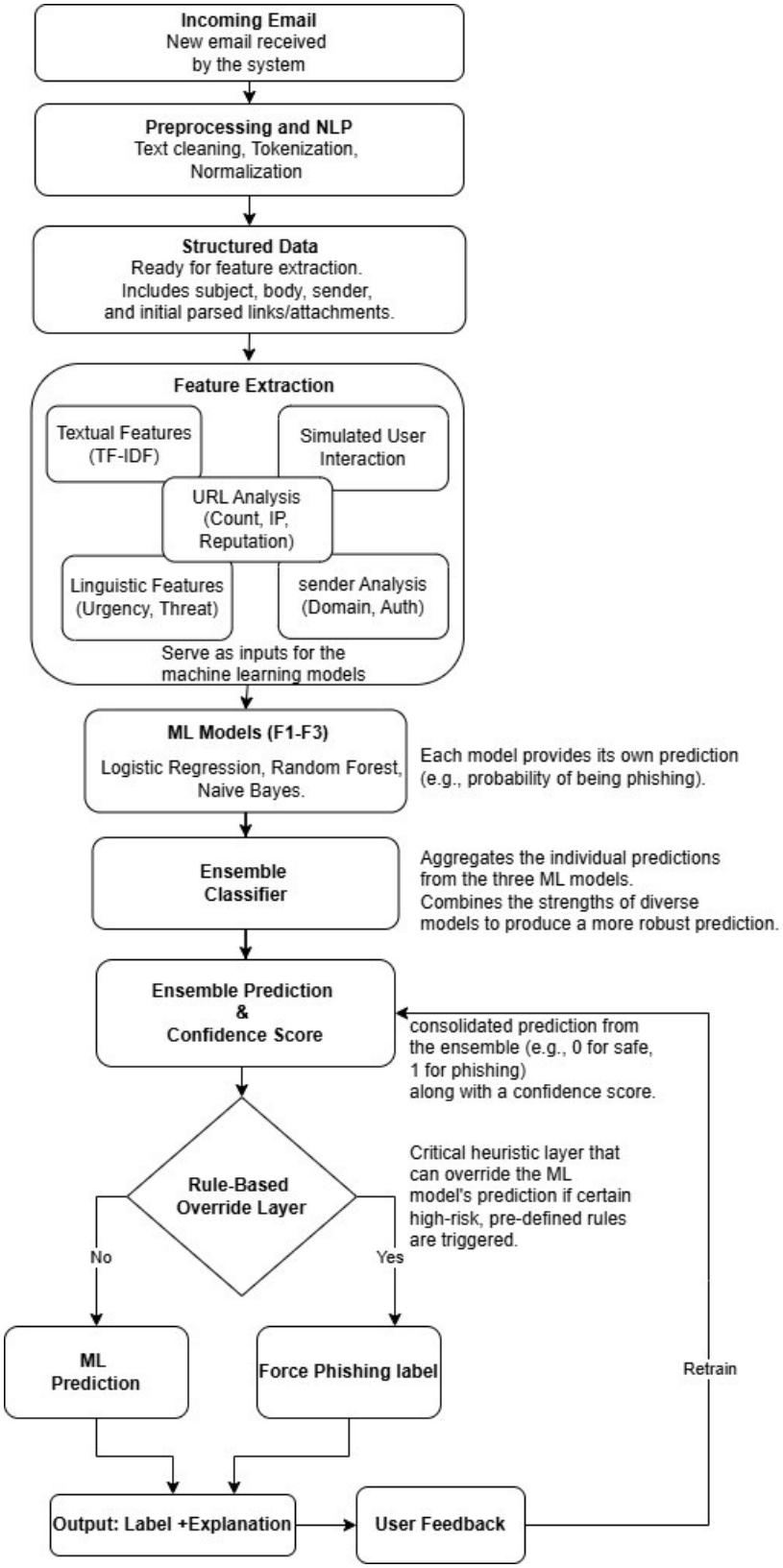

Figure 1. *RealPhish workflow (showing input → processing → output).*

5. Evaluation, Result and Discussion

 doi:10.20944/preprints202511.0218.v1

The performance of RealPhish was evaluated using a labeled dataset of 1,162 emails, consisting of 835 legitimate emails (class 0) and 327 phishing emails (class 1). The algorithm demonstrated strong overall performance across multiple evaluation metrics, as shown in Table 1.

Table 1. Performance Metrics for RealPhish.

Class	Precision	Recall	F1-Score	Support
Legitimate (0)	0.96	0.98	0.97	835
Phishing (1)	0.94	0.89	0.92	327
Overall Accuracy			**0.95**	1,162
Macro Average	0.95	0.93	0.94	1,162
Weighted Average	0.95	0.95	0.95	1,162

Performance Analysis

Accuracy: RealPhish achieved an overall accuracy of 95%, correctly classifying 1,104 out of 1,162 emails. This high accuracy demonstrates the effectiveness of combining machine learning models with rule-based heuristics for phishing detection.

Precision: The precision for legitimate emails (0.96) indicates that when RealPhish classifies an email as legitimate, it is correct 96% of the time. For phishing emails, the precision of 0.94 means that 94% of emails flagged as phishing were actually malicious. High precision is crucial in real-world applications, as it reduces false alarms that may cause alert fatigue or disrupt legitimate communications.

Recall: The recall for legitimate emails (0.98) shows that RealPhish correctly identified 98% of all legitimate emails in the dataset. For phishing emails, the recall of 0.89 indicates that the algorithm successfully detected 89% of all phishing attempts. While this recall rate is slightly lower than for legitimate emails, it still represents strong performance in identifying the majority of threats.

F1-Score: The F1-scores of 0.97 for legitimate emails and 0.92 for phishing emails represent the harmonic mean of precision and recall, providing a balanced measure of the algorithm's performance. These high F1-scores confirm that RealPhish achieves a good balance between minimizing false positives and detecting actual threats.

Interpretation of Results

The slightly lower recall for phishing emails (0.89) compared to legitimate emails (0.98) suggests that RealPhish is more conservative in flagging emails as phishing, prioritizing precision over recall. This design choice helps minimize false positives, which is often preferable in real-world email systems where incorrectly blocking legitimate communications can be disruptive.

The macro-average metrics (precision: 0.95, recall: 0.93, F1-score: 0.94) demonstrate that RealPhish performs well across both classes, without being overly biased toward the majority class (legitimate emails). This is particularly important given the class imbalance in the dataset (835 legitimate vs. 327 phishing emails).

The weighted averages, which take into account the relative frequency of each class, align closely with the overall accuracy (0.95), further confirming the robust performance of the algorithm across the entire dataset.

6. Limitation and Future Work

 doi:10.20944/preprints202511.0218.v1

Despite the promising results achieved by RealPhish, several limitations remain that warrant further attention. The effectiveness of algorithm is heavily dependent on the quality and diversity of its training data. The lack of representative examples, - particularly those reflecting newly emerging phishing techniques—can reduce its effectiveness in real-world scenarios. Additionally, while the rule-based override layer enhances detection for known threats, it may not fully capture the complexity of evolving phishing strategies, which often adapt to bypass static rules. To address these challenges, future work will focus on enhancing the system's adaptability through automated updates based on newly observed phishing patterns, exploring advanced deep learning models to capture more complex behavioral and linguistic signals, and optimizing the algorithm for faster inference to maintain real-time performance. Incorporating user feedback into the learning loop will also be explored to improve accuracy through real-world corrections. Furthermore, seamless integration with existing email security platforms will be prioritized to support practical deployment. The future research will include a comprehensive comparative evaluation of RealPhish against other state-of-the-art phishing detection systems to benchmark its performance and identify areas for further refinement.

7. Conclusion

RealPhish is a comprehensive and robust algorithm designed to detect phishing emails in real time. By combining machine learning models with rule-based heuristics, it delivers high accuracy and reliability in identifying phishing threats. Evaluation results highlight effectiveness, making it as a valuable tool for enhancing email security. Overall, RealPhish offers a scalable and transparent framework for strengthening defenses against phishing in today's rapidly growing dynamic threat environments.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Phishing dataset.xls

References

1. D. Chapagain, N. Kshetri, B. Aryal, and B. Dhakal, "SEAtech: Deception Techniques in Social Engineering Attacks: An Analysis of Emerging Trends and Countermeasures," *arXiv preprint arXiv:2408.02092,* 2024.
2. P. Bountakas, K. Koutroumpouchos, and C. Xenakis, "A Comparison of Natural Language Processing and Machine Learning Methods for Phishing Email Detection," presented at the Proceedings of the 16th International Conference on Availability, Reliability and Security, Vienna, Austria, 2021. [Online]. Available: https://doi.org/10.1145/3465481.3469205.
3. S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," *Decision Support Systems,* vol. 107, pp. 88-102, 2018/03/01/ 2018, doi: https://doi.org/10.1016/j.dss.2018.01.001.
4. N. S. Mudiraj, "Detecting Phishing using Machine Learning," *Published in International Journal of Trend in Scientific Research and Development (ijtsrd),* vol. 3, no. 4, pp. 488-490, 2019.
5. S. Sankhwar, D. Pandey, and R. A. Khan, "Email Phishing: An Enhanced Classification Model to Detect Malicious URLs," *EAI Endorsed Trans. Scalable Inf. Syst.,* vol. 6, no. 21, p. e5, 2019.
6. S. Palka and D. McCoy, "Dynamic phishing content using generative grammars," in *2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 13-17 April 2015 2015, pp. 1-8, doi: 10.1109/ICSTW.2015.7107458.
7. V. Shahrivari, M. M. Darabi, and M. Izadi, "Phishing detection using machine learning techniques," *arXiv preprint arXiv:2009.11116,* 2020.
8. V. Nguyen, "Attribution of spear phishing attacks: A literature survey," 2013.
9. W. P. Nmachi, "Phishing mitigation techniques: A literature survey," *Available at SSRN 3831721,* 2021.

 doi:10.20944/preprints202511.0218.v1

10. P. Dewan, A. Kashyap, and P. Kumaraguru, "Analyzing social and stylometric features to identify spear phishing emails," in *2014 apwg symposium on electronic crime research (ecrime)*, 2014: IEEE, pp. 1-13.
11. S. Hamadouche, O. Boudraa, and M. Gasmi, "Combining Lexical, Host, and Content-based features for Phishing Websites detection using Machine Learning Models," *EAI Endorsed Trans. Scalable Inf. Syst*, vol. 11, pp. 1-15, 2024.
12. A. Arshad, A. U. Rehman, S. Javaid, T. M. Ali, J. A. Sheikh, and M. Azeem, "A systematic literature review on phishing and anti-phishing techniques," *arXiv preprint arXiv:2104.01255*, 2021.
13. R. B. Basnet and A. H. Sung, "Classifying phishing emails using confidence-weighted linear classifiers," in *International conference on information security and artificial intelligence (ISAI)*, 2010: Citeseer, pp. 108-112.
14. A. Cidon, L. Gavish, I. Bleier, N. Korshun, M. Schweighauser, and A. Tsitkin, "High precision detection of business email compromise," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 1291-1307.
15. G. Harinahalli Lokesh and G. BoreGowda, "Phishing website detection based on effective machine learning approach," *Journal of Cyber Security Technology*, vol. 5, no. 1, pp. 1-14, 2021.
16. Y. Li, K. Xiong, and X. Li, "Applying machine learning techniques to understand user behaviors when phishing attacks occur," *EAI Endorsed Transactions on Security and Safety*, vol. 6, no. 21, 2019.
17. A. Kashapov, T. Wu, S. Abuadbba, and C. Rudolph, "Email summarization to assist users in phishing identification," in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, 2022, pp. 1234-1236.
18. J. Kongmanee et al., "A human-AI interaction dashboard for detecting potentially malicious emails," in *2024 IEEE 4th International Conference on Human-Machine Systems (ICHMS)*, 2024: IEEE, pp. 1-6.
19. A. Kumar, "Phishing email detection using machine learning," *Int. J. Sci. Res. Eng. Manag*, vol. 8, pp. 1-5, 2024.
20. A. Mughaid, S. AlZu'bi, A. Hnaif, S. Taamneh, A. Alnajjar, and E. A. Elsoud, "An intelligent cyber security phishing detection system using deep learning techniques," *Cluster Computing*, vol. 25, no. 6, pp. 3819-3828, 2022.
21. N. Xu, J. Fan, and Z. Wen, "Email reading behavior-informed machine learning model to predict phishing susceptibility," in *International Conference on Artificial Intelligence Security and Privacy*, 2023: Springer, pp. 579-592.
22. A. Al-Subaiey, M. Al-Thani, N. A. Alam, K. F. Antora, A. Khandakar, and S. A. U. Zaman, "Novel interpretable and robust web-based AI platform for phishing email detection," *Computers and Electrical Engineering*, vol. 120, p. 109625, 2024.