STM JOURNALS

Journal of
Computer Technology & Applications

ISSN: 2229-6964 (Online)
ISSN: 2347-7229 (Print)
Volume 16, Issue 2, 2025
May–August
DOI (Journal): 10.37591/JoCTA

https://journals.stmjournals.com/jocta

Review JoCTA

Phisherman: A Phishing Email Detection Browser Extension

Vaibhav Valmik Shermale[1,*], Sarang Gopal More[1], Darshan Mahesh Jadhav[1], Om Devidas Chaudhari[1]

Abstract

Phishing attacks continue to pose significant security risks, exploiting email as a primary vector to deceive users and compromise sensitive information. To counter these threats, Phisherman presents a sophisticated, real-time phishing detection system that integrates both rule-based methods and deep learning for heightened accuracy. Built as a cross-browser extension, compatible with Chrome, Firefox, and Edge through the WebExtension API, Phisherman combines traditional verification checks, such as DNS blacklisting, SPF, DKIM, and DMARC, with an advanced 1D-CNN and Bi-GRU deep learning model. This hybrid approach allows Phisherman to identify a wide range of phishing tactics, from well-known techniques to emerging, more subtle patterns that evade conventional filters. Upon detection, the system automatically moves flagged emails to the spam folder and promptly alerts the user, thereby minimizing the risk of interaction with malicious content. By combining multiple verification layers with a user-friendly interface, Phisherman offers high detection accuracy, low false-positive rates, and seamless integration, establishing itself as a robust, accessible solution for enhanced email security in both individual and organizational contexts.

Keywords: Phishing, cyber security, browser extension, Bi-GRU, LSTM, 1D-CNNPD

INTRODUCTION

Phishing attacks have emerged as one of the most persistent and damaging threats in the realm of cybersecurity [1, 2], with attackers using deceptive emails to trick users into disclosing sensitive information, such as passwords, financial details, and personal data. As phishing techniques become increasingly sophisticated, traditional email filtering and detection systems are often inadequate, struggling to differentiate between legitimate messages and cleverly disguised phishing attempts. This challenge underscores a significant gap in current email security measures, which can expose individuals and organizations to financial loss, identity theft, and reputational harm.

The objective of this study is to develop a more comprehensive phishing detection solution that addresses the limitations of existing systems by combining multiple detection techniques. The goal is to detect both straightforward and advanced phishing attempts in real-time, minimizing user exposure to potentially malicious emails. Key objectives include *high detection accuracy, minimal false-positive rates, cross-browser compatibility,* and a *user-friendly alert system* that empowers users to make safer decisions regarding suspicious emails. Designed for accessibility, this system empowers even *non-technical users* to identify and mitigate phishing threats effortlessly.

***Author for Correspondence**
Vaibhav Valmik Shermale
E-mail: vaibhavs.forwork@gmail.com

[1]Student, Department of Computer Engineering, Mumbai Educational Trust (MET) Institute of Engineering, Nashik, Maharashtra, India.

Received Date: March 03, 2025
Accepted Date: April 21, 2025
Published Date: May 03, 2025

Citation: Vaibhav Valmik Shermale, Sarang Gopal More, Darshan Mahesh Jadhav, Om Devidas Chaudhari. Phisherman: A Phishing Email Detection Browser Extension. Journal of Computer Technology & Applications. 2025; 16(2): 99–105p.

The motivation behind this project is rooted in the rapidly increasing volume and sophistication of

phishing attacks, which target millions of users worldwide, exploiting email as a primary attack vector. Despite ongoing improvements in cybersecurity, phishing continues to be a leading cause of data breaches, requiring a more adaptive, multi-layered approach to effectively protect users. By addressing the challenges of detecting diverse phishing methods with an efficient and accessible tool, this work aims to contribute to a safer digital environment for both individual and organizational email users.

RELATED WORK

Atawneh and Aljehani [3] and Altwaijry *et al.* [4]

These papers investigate the use of various deep learning models for phishing email detection, focusing on one-dimensional convolutional neural networks (1D-CNNPD). The authors propose enhanced versions of the base model by adding layers like LSTM, Bi-LSTM, GRU, and Bi-GRU. Two benchmark datasets (Phishing Corpus and Spam Assassin) were used for training and testing. Among the models, the 1D-CNNPD with Bi-GRU achieved the highest performance, with a precision of 100%, an accuracy of 99.68%, and a recall of 99.32%.

Rawal *et al.* [5]

This paper explores machine learning techniques for detecting phishing emails by extracting nine key features from email content. These features include link-based (e.g., number of links), tag-based (e.g., presence of JavaScript, form tag), and word-based features (e.g., action words and keywords like "paypal" or "account"). Several machine learning classifiers were evaluated, including Support Vector Machine (SVM), Random Forest, Naive Bayes, Logistic Regression, and Voted Perceptron. The results show the effectiveness of machine learning models in classifying phishing emails, but the authors highlight limitations related to the dataset, which may not fully replicate real-world scenarios.

Thakur *et al.* [6]

This paper systematically reviews the application of deep learning techniques in phishing email detection. The authors focus on models such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and attention mechanisms. The study critically examines the strengths and limitations of these models, particularly noting their success in detecting phishing emails written in English. One of the key challenges highlighted is the limitation of existing models in handling multilingual phishing emails, reducing their overall applicability.

Chen and Hossain [7]

This study details the development of a Google Chrome extension for detecting phishing emails, leveraging the Term Frequency-Inverse Document Frequency (TF-IDF) technique to identify suspicious words. The extension analyzes email samples and uses a high-frequency term filtering mechanism to detect phishing attempts. However, limitations include the extension's inability to directly access encrypted emails, requiring users to manually access the email's source to perform phishing detection. Another shortfall is the need for manual activation by users, as the extension does not automatically run the detection algorithm.

Sahingoz *et al.* [8]

This paper focuses on detecting phishing websites using machine learning techniques to analyze URLs. Seven different classifiers, including Random Forest, Naive Bayes, and k-NN, are used to categorize URLs as phishing or legitimate based on feature sets such as word vectors and Natural Language Processing (NLP) features. The Random Forest classifier, combined with NLP-based features, achieved the highest accuracy of 97.98%. The paper highlights the effectiveness of machine learning models but notes limitations related to dataset size, particularly for legitimate URLs.

Bagui *et al.* [9]

This paper compares machine learning (ML) and deep learning (DL) techniques for classifying phishing and legitimate emails. Using a dataset of 18,366 labeled emails (3,416 phishing and 14,950

legitimate), the study evaluates the performance of various classifiers, including Naïve Bayes, Support Vector Machines (SVM), Decision Trees, LSTM, CNN, and Word Embedding. Deep learning models, particularly CNN and LSTM, demonstrated higher accuracy compared to traditional ML models. However, SVM exhibited inconsistent results, especially with word phrasing features.

Shalini *et al.* [10]

This paper presents a phishing email detection system that uses both machine learning (ML) and deep learning (DL) techniques. The study focuses on data analysis, tokenization, lemmatization, and oversampling for balancing the dataset. The authors utilize an Artificial Neural Network (ANN) for detecting phishing attempts, comparing its performance to three other ML models. One limitation of the study is its narrow focus on just three ML models and one DL model, potentially overlooking other effective approaches.

PROPOSED APPROACH

The architecture for *Phisherman* is designed as a cross-browser, real-time phishing detection system, implemented as a browser extension [7] to operate seamlessly across Chrome, Firefox, and Edge using the *WebExtension API.* The system integrates within the user's email client to provide layered protection against phishing attacks, combining rule-based verification methods with a deep learning model for advanced detection capabilities. Below is a detailed structure and process flow of the system:

Browser Integration and Compatibility

- Utilizes the *WebExtension API* to ensure cross-browser compatibility, enabling deployment on Chrome, Firefox, and Edge without extensive platform-specific adjustments.
- Interacts directly with web-based email clients to intercept incoming emails, making this system accessible directly from the browser without additional software installation.

Email Interception and Data Extraction

- *Real time email monitoring:* Continuously monitors the inbox for new incoming emails to initiate the detection process immediately upon receipt.
- *Data extraction:* Extracts key elements from the email, including the subject, body, sender's metadata, and any attachments or embedded links. This data is essential for both rule-based and deep learning analysis.

Data Preprocessing

- *Content cleaning:* Strips out unnecessary HTML tags, encoded files, scripts, and extraneous characters, ensuring that the data passed to detection models is clean and standardized.
- *Normalization:* Standardizes text data, such as converting URLs to a placeholder (e.g., http), which helps in pattern recognition and reduces noise during analysis.

Rule-Based Verification

- *DNS blacklisting:* Checks the sender's domain and any linked domains against a database of known malicious addresses, identifying emails from flagged sources [8].
- *SPF, DKIM, and DMARC verification:* Authenticates the sender's domain and verifies email integrity by performing SPF (Sender Policy Framework), DKIM (Domain Keys Identified Mail), and DMARC (Domain-based Message Authentication, Reporting and Conformance) checks. These protocols confirm that the email is coming from an authorized server for the sender's domain, reducing the risk of spoofing.
- *File extension blacklisting:* Scans attached files for high-risk extensions (e.g., .exe, .scr) commonly used for malware, blocking emails with dangerous attachments.
- *Email address analysis:* Validates that the sender's email domain aligns with the claimed domain and checks for subtle domain variations that may indicate phishing.

Deep Learning Model: 1D Convolutional Neural Network (1D-CNN) with Leaky ReLU activation and Bidirectional Gated Recurrent Unit (Bi-GRU)

- *1D-CNN layer:* Extracts local patterns from the email text, identifying features that could indicate phishing, such as suspicious phrases or structural anomalies [4].
- *Bi-GRU layer:* Captures sequential dependencies within the email text, analyzing the content from forward and backward directions to understand context more effectively [3, 4].
- *Classification:* After feature extraction, the model classifies each email as either phishing or legitimate based on learned patterns and behaviors commonly found in phishing emails.

Detection and Decision-Making Pipeline

- *Layered detection pipeline:* The system combines the results of rule-based checks and the deep learning model to make a final classification. Each detection layer operates independently, but their results are aggregated to form a comprehensive decision.
- *Decision thresholds:* Thresholds are implemented to flag an email as phishing based on the combined outputs of rule-based and machine learning checks, optimizing for high accuracy with minimal false positives.

User Notification and Email Handling

- *Automated email handling:* Identified phishing emails are automatically moved to the spam folder, reducing the risk of accidental interaction with malicious content.
- *User notification:* The user is immediately alerted with a concise message about the detected phishing attempt, including details on the specific checks or model patterns that triggered the alert.
- *User preferences and customization:* Stores user settings locally, allowing customization of detection sensitivity and notification preferences for a personalized experience.

Storage and Data Management

- *Local storage via WebExtension API:* Stores user preferences, detection logs, and blacklists securely in local storage, ensuring data privacy and accessibility without an external database.

Modular and Scalable Architecture

- *Component modularity:* Each component, such as email preprocessing, rule-based checks, and deep learning analysis, operates independently but integrates cohesively within the system.
- *Scalability:* The architecture allows for future expansion, such as adding additional detection layers or extending to mobile platforms, without significant structural changes.

User Accessibility and Ease of Use

- Designed to be intuitive and user-friendly, this system requires no technical knowledge to operate. The system runs automatically, with minimal setup and streamlined notifications, making it suitable for both individual users and enterprises focused on enhancing email security Figure 1.

ALGORITHM USED

- *Input layer:* The model accepts preprocessed email text data, which is tokenized and converted into numerical sequences (using Word2Vec or GloVe embeddings) to create vector representations of words. To ensure uniformity, sequences are padded or truncated to a fixed length.
- *1D convolutional layer (1D-CNN):* This layer extracts local patterns and features within the email, focusing on specific phishing-related phrases and structures. Multiple 1D convolution filters scan the input sequence, each detecting unique patterns across different word or character windows. The Leaky ReLU activation introduces non-linearity, enhancing the layer's ability to capture subtle phishing indicators by allowing gradients to pass even for small negative inputs. The result is a feature map, where each feature represents a detected pattern in the text [4].

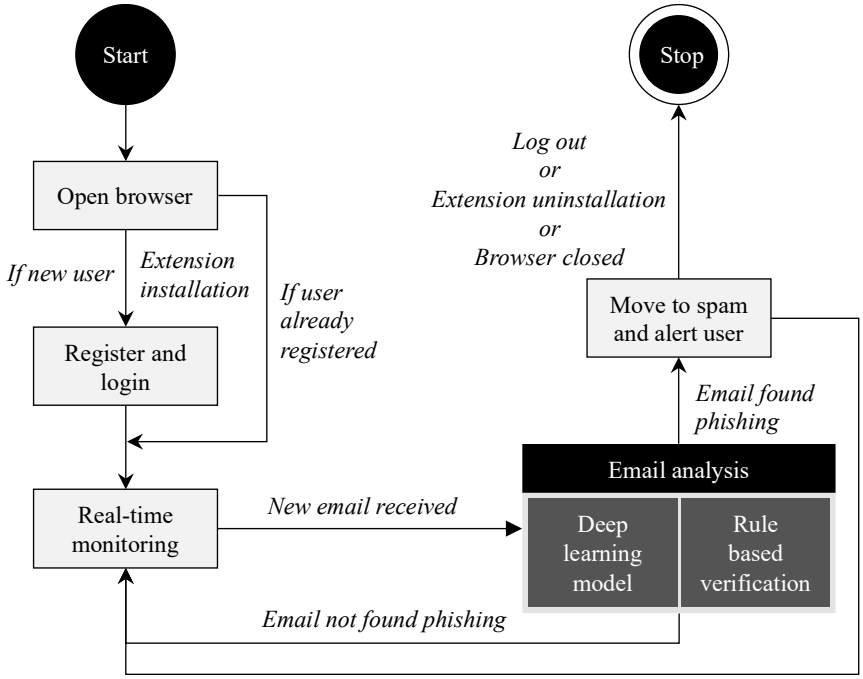

Figure 1. State machine diagram.

- *Max pooling layer:* By applying a max-pooling operation, this layer reduces the dimensionality of the feature map, retaining only the most relevant features. This process enhances computational efficiency, minimizes overfitting, and isolates the key features most indicative of phishing.
- *Bidirectional gated recurrent unit (Bi-GRU) layer:* The Bi-GRU layer captures sequential dependencies and contextual information within the email, analyzing the text in both forward and backward directions. This dual analysis allows the model to recognize relationships across the entire sequence, essential for detecting phishing emails that rely on contextual cues. The result is a hidden state vector, which represents the email's sequential and contextual information [4].
- *Fully connected (dense) layer:* After flattening the output from the Bi-GRU layer, the dense layer performs non-linear transformations, preparing the data for final classification. Leaky ReLU is applied here to maintain non-linearity, improving feature representation for classification.
- *Output layer:* This layer performs binary classification using a sigmoid activation function, producing a probability score between 0 and 1. Emails with scores above a specific threshold are classified as phishing, while others are classified as legitimate.
- *Training and optimization:* The model uses Binary Cross-Entropy Loss to minimize misclassifications by measuring the difference between predicted probabilities and actual labels. The Adam optimizer is used to adjust model weights based on gradients during backpropagation, accelerating convergence and enhancing model performance.

TECHNICAL CONCEPTS

DNS Blacklisting [8]

The system performs DNS blacklisting by comparing the sender's domain and linked URLs against a whitelist of known legitimate domains. When an email arrives, the system checks if the sender's domain is present in the whitelist. This method minimizes false positives by ensuring that only domains not listed in the whitelist undergo further inspection. If the sender's domain is absent from the whitelist, system flags the email as suspicious, triggering additional verification steps. By using a whitelist-based approach, the system offers more accurate initial filtering, especially for domains not previously associated with malicious activity.

- *SPF (Sender policy framework) verification:* SPF is a protocol that confirms whether an email is sent from an authorized server for the sender's domain. The system extracts the SPF record from

the sender's DNS settings and verifies if the originating IP address is listed as authorized. If the sender's IP does not match the SPF policy, it flags the email as potentially spoofed. This verification helps prevent unauthorized use of the domain by ensuring that only legitimate servers can send emails from it.

- *DKIM (DomainKeys identified mail) verification:* DKIM allows verification of the email's integrity and authenticity by attaching a digital signature to the message header. The system extracts the DKIM-Signature header and uses the sender's public key in the DNS record to verify the signature against the original email content. If verification fails, this indicates potential tampering or spoofing, prompting further inspection. This ensures that the email was not altered during transmission, which is critical for authenticity.
- *DMARC (Domain-based message authentication, reporting, and conformance) verification:* DMARC combines SPF and DKIM checks to provide additional protection against email spoofing. The system queries the DMARC record from the sender's DNS and verifies that the "From" header aligns with the domains used in SPF and DKIM checks. If the email lacks alignment or fails both SPF and DKIM checks, this system follows the DMARC policy (reject, quarantine, or monitor) to determine the next steps. This adds a policy-based layer of security, enhancing domain-level email authentication.
- *Certificate verification and validation:* The system performs certificate verification to authenticate the sender's identity and check the email's cryptographic certificate chain. This involves retrieving the sender's certificate, validating its chain against trusted root certificates, and verifying that the certificate is not expired or revoked. Additionally, the system checks if the certificate's registered email address matches the sender's email in the message header. If discrepancies exist, or if the certificate fails verification, the email is flagged as suspicious. This step ensures that emails originate from trusted, verified sources, adding an additional layer of security for emails with cryptographic signing.
- *File extension blacklisting:* The system scans attachments for high-risk file extensions frequently used in malware distribution (e.g., .exe, .scr, .js). This process involves analyzing the extensions of attached files and cross-referencing them with a list of disallowed file types. If an attachment is found to have a risky extension, the email is flagged as potentially malicious. This feature helps prevent users from unknowingly downloading harmful files.

Future Scope

The system provides a strong and reliable method for detecting phishing attempts in email messages. By leveraging DNS blacklisting, file extension blacklisting, certificate verification, and DKIM, SPF, and DMARC checks, the system effectively validates the legitimacy of incoming emails. These integrated security measures work together to provide comprehensive protection against phishing attacks, ensuring a higher level of email security. The system's design is both reliable and efficient, making it a valuable tool for safeguarding users from evolving phishing threats in today's digital communication landscape.

CONCLUSION

The system provides a strong and reliable method for detecting phishing attempts in email messages. By leveraging DNS blacklisting, file extension blacklisting, certificate verification, and DKIM, SPF, and DMARC checks, the system effectively validates the legitimacy of incoming emails. These integrated security measures work together to provide comprehensive protection against phishing attacks, ensuring a higher level of email security. The system's design is both reliable and efficient, making it a valuable tool for safeguarding users from evolving phishing threats in today's digital communication landscape.

REFERENCES

1. Chavan D, Yawalkar PM. Privacy and Owner autherisation framework to manage keys. Int J Adv Res Innov Ideas Educ. 2017; 3(4): 785–795.

2. Dabhade V, Alvi AS. An Energy efficient approach for Secure data communication using Pairwise key encryption in WSN. Neuro Quantology. 2022 Nov; 20(15): 6311–6321.
3. Atawneh S, Aljehani H. Phishing email detection model using deep learning. Electronics. 2023 Oct 15; 12(20): 4261.
4. Altwaijry N, Al-Turaiki I, Alotaibi R, Alakeel F. Advancing phishing email detection: A comparative study of deep learning models. Sensors. 2024 Mar 24; 24(7): 2077.
5. Rawal S, Rawal B, Shaheen A, Malik S. Phishing detection in e-mails using machine learning. Int J Appl Inf Syst. 2017 Oct; 12(7): 21–4.
6. Thakur K, Ali ML, Obaidat MA, Kamruzzaman A. A systematic review on deep-learning-based phishing email detection. Electronics. 2023 Nov 5; 12(21): 4545.
7. Chen H, Hossain M. Developing a Google Chrome Extension for Detecting Phishing Emails. EPiC Ser Comput. 2021; 77: 13–22.
8. Sahingoz OK, Buber E, Demir O, Diri B. Machine learning based phishing detection from URLs. Expert Syst Appl. 2019 Mar 1; 117: 345–57.
9. Bagui S, Nandi D, Bagui S, White RJ. Classifying phishing email using machine learning and deep learning. In 2019 IEEE International Conference on Cyber Security and Protection of Digital Services (Cyber Security). 2019 Jun 3; 1–2.
10. Shalini L, Manvi SS, Gowda NC, Manasa KN. Detection of phishing emails using machine learning and deep learning. In 2022 IEEE 7th International conference on communication and electronics systems (ICCES). 2022 Jun 22; 1237–1243.

