International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Comprehensive Survey of Phishing Detection and URL Validation Techniques

Anuja Phapale[1], Shreya Kadam[2], Sarthak Shah[3], Suhani Shinde[4], Sarthak Patil[5]

[1]Anuja Phapale, Assistant Professor, Department of Information Technology, AISSMS IOIT, Pune
[2]Shreya Kadam, Student, Department of Information Technology, AISSMS IOIT, Pune
[3]Shreya Kadam, Student, Department of Information Technology, AISSMS IOIT, Pune
[4]Shreya Kadam, Student, Department of Information Technology, AISSMS IOIT, Pune
[5]Shreya Kadam, Student, Department of Information Technology, AISSMS IOIT, Pune

ABSTRACT:

Phishing attacks are a form of online scam where cybercriminals trick individuals into revealing sensitive information, such as passwords, credit card numbers, or other personal details, by pretending to be trustworthy organizations or individuals. These attacks are typically carried out through fake emails, messages, or websites that appear legitimate, but are designed to steal information or install harmful software on the victim's device. Phishing has become one of the most common and dangerous types of cybercrime, causing significant financial and personal damage to users worldwide.

To counter these threats, various phishing detection techniques have been developed, with a particular focus on machine learning (ML), deep learning (DL), and Natural Language Processing (NLP). This paper reviews and compares the different methods used to detect phishing, such as identifying fake websites or analysing phishing emails. It highlights the effectiveness of advanced deep learning algorithms like Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) in accurately detecting phishing websites. It also looks at how NLP is used to analyse the content of phishing emails. Furthermore, it explores the role of traditional machine learning algorithms, such as Random Forest, Decision Trees, and Support Vector Machines (SVM), in identifying phishing attacks. Despite progress, challenges such as false alarms and the constantly evolving tactics of attackers remain a concern.

Keywords: Cybercrime, Phishing, Machine Learning (ML), Deep Learning (DL), Convolutional Neural Network (CNN), Natural Language Processing (NLP), Recurring Neural Network (RNN), Support Vector Machine (SVM), Feature Extraction etc.

1. Introduction

Phishing Phishing is a type of cyberattack that relies on social engineering tactics to trick individuals into revealing sensitive information such as personal identities and financial details. Cybercriminals often disguise themselves as trusted entities and deliver deceptive messages through various channels, including email services (like Gmail or Outlook) and social media platforms (such as Twitter and Facebook). Victims are typically compromised when they unknowingly provide personal data or download infected attachments.

Recently, the frequency of phishing attacks on social media has escalated. These platforms allow attackers to reach vast numbers of users worldwide by sharing just one malicious post. As reported by the Anti-Phishing Working Group (APWG), phishing incidents surged by 250,000 in January 2021 alone. Furthermore, business-related compromises jumped by 56% from Q4 2020 to Q1 2021. Figure 1 highlights that industries most targeted in 2021 include financial institutions, webmail services, and social networking platforms, emphasizing that attackers primarily aim to steal financial data or user identities. Some phishing attempts also distribute malware or ransomware, expanding their damage.

Given this growing threat landscape, phishing detection has become a crucial cybersecurity challenge . Despite widespread awareness, many organizations still depend heavily on human expertise to identify such threats. However, distinguishing between legitimate and fake content is increasingly difficult—even for experienced professionals—because phishing messages are becoming highly convincing. To aid detection, security analysts often inspect elements like URLs and email addresses. But attackers are constantly refining their techniques, crafting fake URLs that closely resemble real ones (e.g., https://www.faceb00k.com/ instead of https://www.facebook.com/), making detection even harder.

These include:

- **Blacklists**, which maintain databases of known malicious URLs and IPs. While simple and fast, they fail to detect new, previously unknown threats.
- **Traditional machine learning**, which can learn from phishing patterns, but requires extensive manual feature extraction. This process is time-consuming and may struggle with newly crafted URLs or rapidly evolving attack strategies.
- **Deep Learning (DL)**, which automates feature extraction from both text and images. Although more effective in many cases, DL requires large datasets and significant computing power. Moreover, some models may not detect short URLs or cleverly designed phishing pages.

Detection strategies generally rely on three data types: URL-based, content-based, and hybrid approaches.

- **URL-based detection** analyzes the structure of URLs alone without needing to visit the site. It's safer but may miss critical visual or contextual clues from the webpage.
- **Content-based approaches** examine the page's content (HTML, JavaScript, text, images), offering deeper insight but requiring access to the site, which can introduce risk.
- **Hybrid methods** combine both, aiming to balance safety and detection accuracy.

Numerous survey papers have been published on phishing detection. Some focus on feature extraction methods, analyzing how machine learning models perform using different features (e.g., HTML, CSS, URLs). Others investigate learning techniques, comparing traditional ML, DL, and hybrid approaches. However, many of these works lack a detailed exploration of deep learning methods, especially regarding data preprocessing (like tokenization and feature extraction) and model architecture.

Table 1 outlines the limitations of existing studies, such as their failure to address how DL models handle input data or their minimal coverage of unsupervised learning techniques. Although some papers mention models like CNNs or RNNs, they often don't explore how these models process data or extract features.

To address these gaps, this paper presents a comprehensive review of phishing detection techniques using deep learning. The key contributions of this paper include:

- A detailed review of current DL models for phishing detection, with focus on data preprocessing (cleaning, tokenization, embedding).
- A categorization of detection methods based on data type (URL, content, hybrid) and learning approach (supervised, unsupervised), including end-to-end analysis from raw input to model output.
- A comparative study highlighting the strengths and weaknesses of different methods in terms of data handling, feature extraction, and overall performance.

2. Motivation

The rapid growth of digital communication and online services has significantly increased the attack surface for phishing scams. With attackers continuously evolving their tactics—crafting deceptive URLs and webpages that closely mimic legitimate ones—traditional defense mechanisms often fall short. Existing blacklist and rule-based systems struggle to keep pace with new phishing techniques, while manual feature extraction in classical machine learning is both time-intensive and insufficient for large-scale real-time detection. The complexity and subtlety of modern phishing attacks call for more robust, scalable, and intelligent solutions. This motivates the need for deep learning-based systems that can automatically learn patterns from vast amounts of URL and webpage data, adapt to new attack variants, and offer improved accuracy with minimal human intervention. By focusing on URL-based and HTML content features, our goal is to explore advanced models capable of detecting phishing threats more effectively, even in their most camouflaged forms.

3. Background

Phishing remains one of the most prevalent and disruptive cybersecurity threats faced by organizations today. These attacks often serve as entry points for more severe breaches, where attackers deceive employees into revealing sensitive login credentials, enabling further compromise of critical systems. Despite the use of preventive strategies like email scanning and filtering, some phishing emails manage to bypass defences due to attackers' continuously evolving tactics. As a result, organizations not only rely on preventive mechanisms but also adopt reactive measures to minimize damage when attacks penetrate their defences. Among the most valuable sources for identifying phishing campaigns are user-reported incidents, as these often reveal subtle variations in phishing attempts that evade automated detection systems. However, managing such reports can be time-consuming and labour-intensive, especially when IT teams are required to manually review and correlate multiple emails to identify patterns across phishing campaigns. Campaigns often include multiple variations of a single phishing email, altering attributes such as the sender ID, subject line, or content to bypass filters and prolong the attack. Due to the volume and complexity of such reports, help desk and security teams face challenges in collecting representative samples, which can lead to incomplete mitigation. This highlights the need for automated systems capable of grouping related phishing emails to enable quicker understanding and more effective response to phishing campaigns.

4. Literature Review

Author Name	Applied Approach	Algorithm Used	Dataset	Main Findings	Limitations/Challenges
Shirazi et al.	Machine Learning & Visual Similarity	SVM, Naive Bayes, KNN, Gradient Boosting, Decision Tree	PhishTank (1000), Alexa (1000), OpenPhish (2013)	Gradient boosting achieved 97.00% accuracy	Limited features based only on domain name; small and biased dataset for training/testing

Hannousse and Yahiouche	ML, Visual Similarity & Heuristic	SVM, Decision Tree, Logistic Regression, Random Forest, Naive Bayes	PhishTank, Alexa, OpenPhish, Yandex API (87 features)	Random Forest with hybrid features achieved 96.61% accuracy	Content-based features not suited for runtime; no feature selection apart from manual 87 features; no train-test split ratio mentioned
Rashid et al.	Machine Learning	SVM	Alexa, Common Crawl (5000 URLs)	Achieved 95.66% accuracy in phishing detection	Only one classifier (SVM) and 5 features used; small dataset; only Accuracy metric used
Basit et al.	Machine Learning	Random Forest, KNN, Decision Tree, ANN	UCI repository (11,055 instances, 30 features)	KNN + Random Forest achieved 97.33% accuracy	Did not evaluate on multiple datasets; used open-source UCI dataset with normalized features and no original URL; no feature selection
Stobbs et al.	ML, Heuristic & List Based	Random Forest, Linear Regression, Neural Network, SVM	PhishTank, Alexa	RF + PSO (feature selection) + TPE (hyperparameter optimization) achieved 99.33% accuracy	No split ratio mentioned; some performance metrics lacking; only Recall and Accuracy are better than others
Sahingoz et al.	ML & Heuristic	Naive Bayes, Random Forest, KNN, AdaBoost, K-star, SMO, Decision Tree	Custom dataset (Ebbu 2017) – 73,575 URLs	Random Forest with NLP-based features achieved 97.98% accuracy	Dataset created with custom script; NLP features less effective for short domains
Abedin et al.	ML & Heuristic	KNN, Logistic Regression, Random Forest	Kaggle – 11,504 URLs, 32 attributes	Random Forest: Accuracy 97.0%, Recall 99.0%, F1 Score 97.0%	Limited ML algorithms used; no hyperparameter info; all features used; no feature reduction or comparison with existing studies
Saha et al.	Machine Learning	Random Forest, Decision Tree	Kaggle – 11,504 URLs, 32 attributes	Random Forest achieved highest accuracy of 97.00%	Only two ML algorithms and one dataset used; PCA applied; no comparison with other studies; shallow analysis
Mao	ML & Visual Similarity	SVM, Decision Tree	PhishTank – 2923 instances	Both classifiers achieved over 93% accuracy	Small dataset; only two classifiers tested; lower performance compared to other studies
Sindhu et al.	ML & Heuristic	Random Forest, SVM, Neural Network	UCI – 11,055 URLs (6157 phishing, 4898 legitimate)	Accuracies: 97.36% (RF), 97.45% (SVM), 97.25% (NN)	Single dataset used; UCI is open source with normalized features; lacks original URLs and feature selection
Kasim	ML & Heuristic	SVM, LightGBM, MLP, CNN	ISCXURL-2016 – 2978 instances, 77 features	LightGBM + SAE-PCA achieved 99.60% accuracy	Dataset limited to 2978 instances; features reduced from 77 to 20 via PCA, which may limit generalization

Sánchez-Paniagua et al.	ML & Heuristic	Random Forest, KNN, SVM, Naive Bayes, Logistic Regression	60,000 custom URLs; also used PWD2016 and Ebbu2017 datasets	Random Forest achieved best accuracy of 94.59%	Accuracy lower than other literature; index/login pages in dataset may lower performance
Butnaru et al.	ML & Heuristic	Naive Bayes, Decision Tree, Random Forest, SVM, MLP	PhishTank – 100,315 instances, 12 features (2 newly proposed)	Optimized Random Forest achieved highest accuracy:99%	Compared against Google Safe Browsing; 3 out of 5 classifiers performed well
Munir Prince et al.	Machine Learning	Naive Bayes, C4.5, JRip, PART, KNN, Random Forest, SVM	Mendeley – 10,000 website instances, 48 attributes	Random Forest achieved highest accuracy: 98.36%	Limited dataset; no feature reduction used; overlapping features may affect model
Geyik et al.	Machine Learning	Decision Tree, Logistic Regression, Naive Bayes, Random Forest	PhishTank, Alexa, Common-Crawl	Random Forest achieved highest accuracy: 83.0%	Accuracy significantly lower compared to similar studies using same classifiers and datasets
Anupam and Kar	ML & Heuristic	SVM, Grey Wolf Optimizer, Bat Algorithm, Whale Optimization Algorithm, Firefly Algorithm	PhishTank, Yahoo, UCI	Grey Wolf Optimizer achieved highest accuracy of 90.38%	UCI dataset lacks original URL; normalized features; accuracy lower than other studies using same dataset
Suleman and Awan	ML & Heuristic	Naive Bayes, ID3, KNN, Decision Tree, Random Forest, Genetic Algorithms	UCI	ID3 + YAGGA achieved best accuracy of 94.99%	UCI dataset lacks original URL; normalized features
Jain et al.	ML & Heuristic	SVM, Naive Bayes	PhishTank – 33,000 instances, 14 features	SVM classifier achieved 91.28% accuracy	Accuracy lower compared to other studies with similar dataset and classifier
Zuhair and Selamat	ML, Visual Similarity & Heuristic	Hybrid Classifier (Naive Bayes + Decision Tree)	PhishTank, Chinese eBusiness, DMOZ	TPR of 0.984 obtained through hybrid classifier	Accuracy not calculated; algorithm choices not justified; only TPR reported
Ortiz Garces et al.	ML, List Based & Heuristic	Logistic Regression, Neural Network	Kaggle – 420,464 instances	Analyzed anomalous behavior in phishing detection using ML techniques	Only two algorithms used; limited features; no clear feature selection procedure; randomly selected URL characteristics
Korkmaz et al.	ML & Heuristic	Logistic Regression, KNN, Decision Tree, SVM, Naive Bayes, XGBoost, RF, ANN	PhishTank, Alexa, Common-Crawl	Random Forest achieved highest accuracy of 94.59%	Performance is low compared to similar studies with same classifiers and datasets
Patil et al.	ML, List Based, Visual Similarity & Heuristic	Decision Tree, Logistic Regression, Random Forest	Alexa – 9076 websites	Random Forest achieved highest accuracy of 96.58%	Limited algorithms and dataset used; robustness not proved; small dataset; minimal false positives/negatives reported

Yadollahi et al.	ML & Heuristic	Decision Tree, AdaBoost, Kstar, Random Forest, SMO, Naive Bayes, XCS	3983 phishing, 4021 legitimate URLs (total 8004)	XCS achieved highest accuracy of 98.39%	Dataset size limited to 8004 URLs
Palaniappan et al.	ML, List Based & Heuristic	Logistic Regression	PhishTank, Alexa, ICANN, DNS-BH – 20,000 domain names	Accuracy of 60.00% with Logistic Regression	Significantly lower accuracy than other studies; only one algorithm used
Ozker and Sahingoz	ML & Visual Similarity	Naive Bayes, Random Forest, SVM, Logistic Regression, KNN, Decision Tree, MLP, XGBoost	PhishTank – 13,791 samples, 58 features	Random Forest achieved highest accuracy of 97.91%	Dataset not public; multiple algorithms used without justification; dataset insufficient; no feature selection applied
Shirazi et al.	ML, Visual Similarity & Heuristic	SVM, Decision Tree, Gradient Boosting, KNN, Random Forest	UCI, Mendeley	Gradient Boosting achieved highest accuracy of 95.47%	UCI dataset lacks original URLs and has normalized features
Chiew et al.	ML, Visual Similarity & Heuristic	Random Forest, SVM, Naive Bayes, C4.5, JRip, PART	PhishTank, OpenPhish, Alexa, Common Crawl – 50,000 phishing and 50,000 valid URLs	Random Forest achieved highest accuracy of 96.17%	Accuracy comparatively low given similar datasets and classifiers used
Parekh et al.	ML & Heuristic	Random Forest	PhishTank – 31 different URL features	Random Forest achieved around 95% accuracy	Accuracy lower than similar studies; only 31 URL features used for evaluation

5. Literature Review

a. Heuristic-Based Techniques - Heuristic methods rely on extracting specific features from websites that are commonly associated with phishing behavior. These include:

- Disabled right-click functionality
- Presence of the '@' symbol in URLs
- Use of pop-up windows for credentials
- IP addresses in place of domain names

Heuristic-based methods have achieved high accuracy, with some models reaching up to **99.57%**, especially when using ensemble-based classifiers like Random Forest.

b. Visual Similarity-Based Techniques - This approach involves comparing phishing websites with legitimate ones based on their visual elements. These include:

- Page layout and formatting
- Cascading Style Sheets (CSS) and HTML structure
- Logos, screenshots, and other visual components

Some visual similarity techniques have achieved up to **99.77% accuracy**, demonstrating their effectiveness in detecting visually deceptive sites. Studies using large datasets have also reported strong performance, exceeding **98% accuracy** using multiple algorithms.

c. List-Based Techniques- Web browsers commonly utilize list-based techniques by maintaining:

- **Blacklists** of known phishing URLs
- **Whitelists** of trusted and verified domains

These methods are widely used in practice and have shown high accuracies using rule-based classification algorithms. In general, list-based methods are efficient but require frequent updates to remain effective.

d. Machine Learning-Based Techniques - Machine learning techniques involve extracting features such as:

- URL structure
- HTML and JavaScript code patterns
- Site metadata

These features are used to train classification models like Random Forest, Support Vector Machines (SVM), and Decision Trees. Machine learning methods have consistently show Random Forest as one of the best-performing classifiers in this domain.

e. Deep Learning-Based Techniques - Deep learning techniques have recently gained popularity due to their ability to automatically learn complex patterns from large datasets. Common architectures used include:

- Deep Neural Networks (DNN)
- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)

Among these, CNN-based methods have demonstrated the highest accuracy. Even with massive datasets containing millions of URLs, deep learning techniques continue to outperform traditional methods.

6. Feature Extraction

Feature extraction is a critical step in phishing detection as it directly impacts the performance and accuracy of the detection model. It involves identifying and selecting the most relevant characteristics from a website, email, or URL that help differentiate between legitimate and phishing content. Depending on the detection approach, different types of features are considered:

- **URL-Based Features:** These include lexical characteristics such as the length of the URL, presence of special characters (e.g., '@', '-', or '='), number of subdomains, use of IP addresses, and abnormal domain names. These features are lightweight and can be extracted without visiting the website, making them fast and safe to analyze.

- **HTML and JavaScript Features:** These are extracted from the webpage source code and include suspicious scripts, form handlers, hidden elements, use of iframe tags, and redirect scripts. These features provide deeper insights into the page behavior but require content rendering or parsing.

- **Visual Features:** These are used in visual similarity-based techniques and include logos, images, layout structure, font styles, and page rendering. These features help identify visually deceptive phishing pages that mimic legitimate websites.

- **Content-Based Features:** These features are extracted from the textual content of emails or web pages. They include keyword patterns, spelling errors, urgency in language, and phishing-related terms (e.g., "verify your account", "login now"). Natural Language Processing (NLP) techniques are often used to process and analyze such content.

- **Network-Based Features:** These include information from the domain's WHOIS record, SSL certificate properties, domain registration time, server location, and response time. Such features are useful in identifying newly created or suspicious domains often used in phishing attacks.

7. Data Extraction Techniques

a. From URLs

Lexical and structural analysis of URLs is a common and effective method in phishing detection systems. Numerous studies have extracted features such as:

- **URL Length**: Phishing URLs are typically longer to obfuscate their malicious intent.
- **Use of IP Addresses**: The presence of an IP address in the domain portion of the URL instead of a proper hostname is often indicative of phishing.
- **Number of Subdomains**: Multiple subdomains can be used to mimic legitimate domains (e.g., login.bank.example.com.phishingsite.com).
- **Special Characters**: Suspicious characters like '@', '%', '-', and encoded strings are often used for redirection or deception.
- **HTTPS Usage**: The absence of SSL/TLS (i.e., no HTTPS) in URLs may point toward fraudulent activity.
- **Domain Age and Expiry**: WHOIS-based features such as newly registered domains or those nearing expiration are common among phishing websites.

These features are generally extracted using tools like urlparse, regular expressions, and WHOIS queries.

b. From Webpages

In content-based detection systems, especially those targeting real-time webpage analysis, several dynamic and static features are captured:

- **Textual Content**: Pages that request users to "login", "verify your account", or "update password" are often phishing attempts.
- **Form Actions**: The destination of forms (e.g., external or untrusted domains) is analyzed to detect malicious data collection.
- **JavaScript Behavior**: Embedded scripts are examined for obfuscated or malicious code patterns.
- **Visual Cloning**: Layout and design similarities with legitimate sites are detected using visual comparison algorithms.

Common tools for such extraction include HTML parsers like *BeautifulSoup*, automated headless browsers such as *Selenium*, and perceptual hashing techniques for visual analysis.

c. From Emails

Phishing detection in emails involves Natural Language Processing (NLP) and metadata inspection. Key features include:

- **Sender Domain and Email Header**: Anomalies or spoofing in the sender's address and domain.
- **Subject Line**: NLP-based sentiment and urgency detection (e.g., “urgent”, “account locked”).
- **Email Body**: Extraction of key phrases, suspicious links, and overall tone using text preprocessing, tokenization, and named entity recognition.
- **Embedded URLs**: All hyperlinks are extracted and analyzed for suspicious redirection or domain structure.

This is typically accomplished using standard email parsing libraries, along with NLP techniques such as TF-IDF, word embeddings (Word2Vec, GloVe), or transformer-based models (e.g., BERT) for semantic understanding.

d. From Datasets

Publicly available datasets serve as essential resources for benchmarking phishing detection systems. Notable datasets include **PhishTank**, **UCI ML Repository**, and **Kaggle phishing detection datasets**. These datasets often contain:

- **Labeled URL data** (legitimate vs. phishing)
- **Webpage content and metadata**
- **Timestamped logs and domain information**

Preprocessing steps typically involve data cleaning, class balancing (to address data imbalance), and normalization or encoding of features for compatibility with machine learning pipelines.

8. Results and Discussion

The review of multiple phishing detection systems indicates that hybrid models combining lexical URL features, webpage content analysis, and email NLP techniques outperform single-method approaches in accuracy and robustness. Deep learning models, particularly those leveraging CNNs, RNNs, and transformer-based architectures (e.g., BERT), have demonstrated accuracy rates exceeding 95% in controlled environments. Moreover, models incorporating domain knowledge and continual learning techniques (e.g., Life-long Phishing Detection using Continual Learning) show strong adaptability to evolving phishing tactics.

In studies such as *PhishHaven* and *DEPHIDES*, models using real-time URL analysis with backend AI classifiers were able to detect phishing attempts in under 2 seconds. Furthermore, NLP-based phishing email detection systems demonstrated high F1-scores (~93%) in recent benchmarks.

However, models that rely heavily on third-party services (e.g., WHOIS, blacklists) suffer from latency and API availability issues. Lightweight models that emphasize feature selection (e.g., using Extra-Trees or SVMs) offer faster execution times, making them more suitable for integration into browser-based tools.

9. Datasets

Phishing detection systems heavily rely on the quality and diversity of datasets to ensure accurate and generalizable models. Among the most frequently used datasets is PhishTank, often combined with Alexa Top Sites to represent legitimate URLs. While these sources are widely adopted due to accessibility, they suffer from limitations such as small sample sizes, outdated entries, and lack of feature diversity. Studies using these datasets, such as those by Shirazi et al. and Chiew et al., demonstrate respectable accuracy but are constrained by biased and incomplete data.

UCI and Kaggle datasets offer structured features and ease of use, making them popular among researchers like Basit et al. and Abedin et al. However, these datasets often lack original URLs and are based on normalized attributes, restricting their use in advanced detection methods involving visual or linguistic analysis.

Custom datasets like Ebbu2017, PWD2016, and ISCXURL-2016 provide tailored attributes and higher feature richness, but are limited in availability and size, hindering reproducibility. Additionally, large-scale web data from Common Crawl or Mendeley repositories offers better diversity but requires extensive preprocessing.

Overall, while the diversity of datasets reflects the growth of phishing detection research, many are outdated, limited in scope, or lack standardization. This underscores the need for a unified, comprehensive, and publicly available dataset that balances phishing and legitimate URLs with raw and feature-rich data to support effective and reproducible model development.

10. Conclusion

Phishing continues to be one of the most prevalent and evolving cyber threats. Through a comprehensive survey and analysis of existing systems, it is evident that combining multiple feature sets—lexical, behavioral, and contextual—results in more accurate and adaptive phishing detection systems. While deep learning models offer exceptional accuracy, their real-time application may be constrained by resource demands. In contrast, hybrid and lightweight models balance accuracy with execution speed, making them suitable for real-world deployment.

Our study suggests that future phishing detection efforts must focus on scalability, adaptability to new phishing patterns, and seamless user experience, especially when integrated into client-side environments.

11. Future Scope

As phishing techniques continue to evolve rapidly, it is imperative for detection systems to adapt accordingly and offer real-time, user-friendly solutions. A promising direction for future work lies in the development of a Chrome browser extension capable of detecting and preventing phishing attempts during user browsing sessions. Such a tool would integrate the findings of this research into a practical, accessible solution for end-users.

This extension would serve as a scalable, platform-independent defense mechanism, extending phishing protection to non-technical users without requiring specialized software or system-level access. Future enhancements may include more evolved and enhanced DL-based detection of phishing emails viewed within browser-based clients and user feedback mechanisms to enable continual learning.

REFERENCES

A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 2, pp. 590–611, Feb. 2023.

2. S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "Phishing email detection using natural language processing techniques: A literature survey," *Procedia Computer Science*, vol. 2021, Year.

3. S. Asiri, Y. Xiao, S. Alzahrani, S. Li, and T. Li, "A survey of intelligent detection designs of HTML URL phishing attacks," *IEEE Access*, 2023, Year.

4. S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "A systematic literature review on phishing email detection using natural language processing techniques," *IEEE Access*, vol. 2024, Year.

5. Y. A. Alsariera, V. E. Adeyemo, A. O. Balogun, and A. K. Alazzawi, "AI meta-learners and Extra-Trees algorithm for the detection of phishing websites," *IEEE Access*, vol. 2021, Year.

6. Shouq Alnameri, Majid Alshammari , "Detecting phishing domains using machine learning," *MDPI- Applied Sciences*. 2023, Year.

7. O. K. Sahingoz, E. Buber, and E. Kugu, "DEPHIDES: Deep learning based phishing detection system," *IEEE Access*, 2021, Year.

A. Garje, N. Tanwani, S. Kandale, T. Zope, and S. Gore, "Detecting phishing websites using machine learning," *IJCRT*, 2023

8. M. Sameen, K. Han, and S. O. Hwang, "PhishHaven—An efficient real-time AI phishing URLs detection system," *IEEE Access*, 2023, Year.

A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing detection system through hybrid machine learning based on URL," *IEEE Access*, 2023, Year.

B. Catal, G. Giray, B. Tekinerdogan, S. Kumar, and S. Shukla, "Applications of deep learning for phishing detection: A systematic literature review," *Scientific Reports*, vol. 12, no. 1, p. 9276, Jun. 2022.

9. M. Zouina and B. Outtaj, "A novel lightweight URL phishing detection system using SVM and similarity index," *Human-centric Computing and Information Sciences*, vol. 7, no. 17, Jun. 2023.

A. Ejaz, A. N. Mian, and S. Manzoor, "Life-long phishing attack detection using continual learning," *Scientific Reports*, vol. 13, p. 11488, Jul. 2023.

10. K. Althobaiti, M. K. Wolters, N. Alsufyani, and K. Vaniea, "Using clustering algorithms to automatically identify phishing campaigns," *IEEE Access*, Aug. 2023.