Phoenix: International Multidisciplinary Research Journal
Vol.3, No. 3.1, July-Sep, 2025
ISSN: 2583-6897

MACHINE LEARNING BASED PHISHING WEBSITE DETECTION SYSTEM VIA CLOUD

Misbah Jahagirdar

Student of Computer Science and Engineering, Secab Institute of Engineering and Technology, Vijayapur 586101, Karnataka India.

Rizwan Malik

Faculty of Computer Science and Engineering, Secab Institute of Engineering and Technology, Vijayapur 586101, Karnataka India.

https://doie.org/10.65985/pimrj.2025600097

Abstract

In this digital era, the risk of cyberattacks such as phishing has risen significantly. Phishing attacks trick users into revealing sensitive information by disguising malicious websites as legitimate ones. This project focuses on detecting phishing websites using a machine learning-based approach hosted entirely in the cloud. The system is deployed on an AWS EC2 instance and integrated with a custom domain through the Web-Space-Kit platform, providing a seamless and secure web interface for real-time URL analysis. The dataset used in this study comprises 11,000 samples with 33 features extracted from URLs, encompassing both structural and content-based attributes. Logistic Regression, Decision Tree, Support Vector Machine (SVM), Random Forest, and XG-Boost were among the few supervised machine learning algorithms that were implemented and evaluated. The models were evaluated using accuracy as the primary performance metric. Experimental results showed that Logistic Regression achieved 93.42% accuracy, Decision Tree achieved 92.15%, SVM reached 91.64%, Random Forest attained 97.82%, and XG-Boost achieved 96.99%. Among them, Random Forest emerged as the most reliable model due to its ability to handle complex feature interactions and deliver the highest prediction accuracy. The system's cloud-based deployment allows users to enter any URL via a secure HTTPS web portal, instantly obtain phishing or legitimate classification, view an explanation of the decision, and monitor response times. This approach demonstrates how machine learning models, combined with scalable cloud infrastructure, can effectively mitigate phishing risks and support a safer online environment. Future enhancements could include integrating deep learning models, continuous learning to detect new phishing patterns, and browser extension integration for real-time protection.

Keywords- Phishing website detection, cloud deployment, AWS EC2, Random Forest, XG-Boost, machine learning, Web-Space-Kit, secure web application

1. Introduction

The rise of mobile and wireless connectivity allows users to access networks from anywhere, but it also exposes them to new attack surfaces. Many individuals maintain multiple accounts across social media, email services, and financial platforms, making them prime targets for phishing. Often, users are unaware of the risks and unknowingly interact with fraudulent links [1].

These links typically lead to fake websites designed to closely mimic legitimate ones, capturing user data and redirecting it to an attacker-controlled server instead of the intended destination. This project focuses on the detection of phishing websites using advanced machine learning techniques, hosted and deployed on the cloud for real-world accessibility [2].

The implemented system classifies URLs as legitimate or phishing by analysing extracted features. The complete application is deployed on an AWS EC2 instance, with the backend powered by Flask and integrated with a custom-built user interface hosted through the Web-Space-Kit platform using a custom domain (https://cloudphishingportal.info). The deployment also includes SSL certification for secure communication.

The explanation module works by extracting specific characteristics and patterns from the input URL that are commonly used by attackers to deceive victims. Once a prediction is made by the trained machine learning model, the system analyses the URL against a list of phishing indicators, and each matched indicator is returned as part of the explanation. Recently, researchers have also explored deep learning approaches, which further enhance scalability and detection accuracy in phishing website classification [3].

The following key factors are considered:

1. Presence of Special Characters – Many phishing URLs contain special characters such as @, -, _, or %20 to disguise their appearance or create confusion for the user. For example, an attacker may insert @ in a URL so that browsers ignore the preceding part, leading users to believe they are on a trusted site when they are redirected to a malicious one.

2. Use of Numbers in the Domain Name – Legitimate domains rarely contain random numerical sequences in their main address. Attackers often add numbers either to mimic an original domain (e.g., paypal123.com) or because the exact legitimate domain name is already taken.

3. Unsecured Protocol (HTTP) – Modern legitimate websites use HTTPS to encrypt data during transmission. URLs using HTTP without encryption are flagged as suspicious since phishing sites often skip security certificates to avoid cost and verification checks.

4. URL Shortening Services – Services like bit.ly or TINY-URL can be exploited by phishers to hide the actual target link. The system detects shortened URLs and expands them before analysis, checking whether the destination is malicious.

5. Excessive URL Length – Extremely long URLs, especially those exceeding 100 characters, are suspicious because attackers may embed multiple redirects or obfuscate the real destination with unnecessary parameters.

6. Multiple Subdomains – A legitimate website typically uses one or two subdomains. Phishing URLs may use several nested subdomains (e.g., login.bank.secure-update.verify.com) to trick users into thinking they are on a trusted domain.

7. Suspicious Redirection Patterns – If the URL redirects multiple times before landing on the final page, it could indicate an attempt to bypass detection systems or mask the actual phishing page. In

addition to providing prediction and explanation, the system measures the total execution time from when the URL is entered until the classification result is returned.

I. 2. Related Work

Over the past two decades, phishing detection research has evolved from basic list-based filtering to advanced artificial intelligence and cloud-integrated solutions. Traditional anti-phishing methods, such as those implemented by Google Safe Browsing, rely heavily on blacklist-based approaches, where incoming URLs are checked against a continuously updated repository of known malicious domains. If a match is found, the user is warned or blocked from accessing the site. While this method is effective for previously reported phishing domains, it fails to detect zero-day phishing attacks or newly generated malicious URLs that have not yet been reported [4], [5].

To improve blacklist coverage, researchers introduced methods like Phish Net, which expands the blacklist by generating possible variations of known phishing URLs using heuristics such as Top-Level Domain (TLD) substitution, directory structure similarity, IP address equivalence, query string manipulation, and brand name substitution [4].

While this improves the detection rate for URLs that are slightly altered from known malicious domains, it still suffers from high false-negative rates when attackers use entirely new structures. Furthermore, maintaining and updating such lists creates significant operational overhead [6]. With the increasing availability of phishing kits—prepackaged tools that allow attackers with minimal skills to deploy convincing phishing sites within minutes—the sophistication of phishing attacks has risen sharply [7]. These kits can easily mimic legitimate website designs, making traditional detection mechanisms like whitelisting, blacklisting, heuristic analysis, and visual similarity comparison less effective [8], [9].

Whitelisting ensures access only to trusted websites, but struggles with scalability and becomes ineffective if a trusted site is compromised. Blacklisting, on the other hand, is reactive rather than proactive, and heuristic approaches—although faster and capable of detecting new patterns—can still be bypassed by attackers who replicate legitimate structures [9]. Visual similarity-based approaches compare the suspected website's layout and elements with a database of known trusted sites [10]. While accurate in certain scenarios, these methods are computationally expensive and unsuitable for real-time phishing detection in large-scale systems [11]. Moreover, since phishing websites often change hosting and design rapidly, static visual signatures quickly become outdated. To overcome these challenges, researchers have turned towards machine learning-based approaches for phishing detection. These methods use URL-based lexical features, HTML content features, and host-based attributes to classify a website as phishing or legitimate [8], [12], [13].

Supervised machine learning algorithms, such as Random Forest, Support Vector Machines, and Logistic Regression, have been widely applied for this task [14]. Random Forest models, in particular, have shown high robustness in handling large feature sets and detecting phishing URLs with high accuracy [15]. The integration of cloud computing into phishing detection systems has opened new opportunities for scalability, accessibility, and real-time threat response [16], [17].

Cloud-based deployments, such as those on AWS EC2, allow the model to handle high-volume traffic and provide services to users across different regions without performance degradation. Additionally, by hosting models in the cloud, retraining and updating them becomes seamless, enabling continuous learning from newly detected phishing threats [17].

In order to provide transparency in the classification results, recent research has also investigated explainable AI (XAI) techniques in phishing detection. In order to justify its choice, a system can, for instance, highlight lengthy URLs, the use of special characters, HTTP rather than HTTPS, the inclusion of numerical IP addresses, or questionable usage of URL shortening services [13], [15].

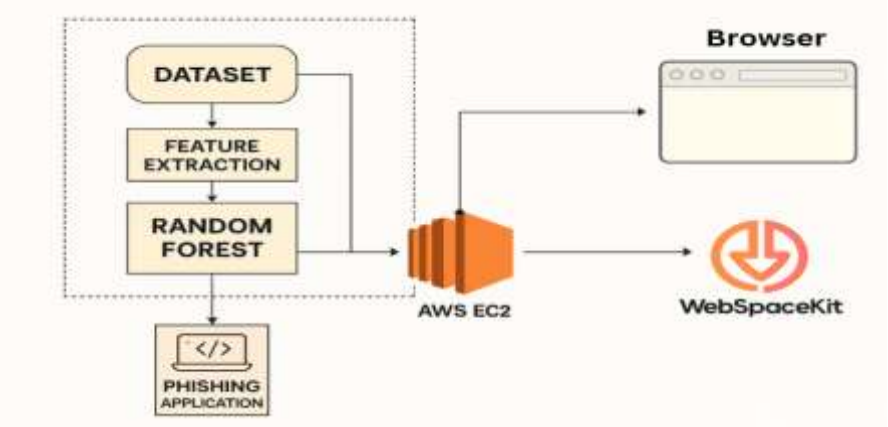
Furthermore, deep learning techniques, such as Convolutional Neural Networks (CNNs) and Deep Reinforcement Learning, have been investigated for extracting high-level patterns from webpage layouts and content [12], [17]. These models can capture subtle similarities between phishing and legitimate sites that traditional algorithms might miss. However, deep learning models require large-scale datasets, high computation power, and are often less interpretable, making them challenging for deployment in resource-limited environments.

In conclusion, while traditional methods like blacklisting and heuristics still play a role in phishing detection, machine learning integrated with cloud-based infrastructure offers a more scalable, accurate, and proactive solution. Our project leverages Random Forest for its proven high accuracy in phishing detection [15] and deploys it in the AWS cloud environment to ensure global accessibility, real-time analysis, and continuous threat intelligence integration [16], [17].

3. Proposed system

The first step in the process is to obtain the dataset from reliable open-source repositories such as Kaggle. For this project, we utilized a dataset consisting of 11,000 samples and 33 extracted features, representing both phishing and legitimate websites. This dataset was selected because it has already undergone partial pre-processing, ensuring a certain level of reliability while still requiring further cleaning for optimal machine learning performance.

Figure 1: Methodology Diagram



1) ***A. Dataset Preparation***

The phishing dataset was collected from trusted repositories such as Kaggle and UCI, containing both legitimate and malicious URLs.

B. Feature Extraction

After dataset preparation, relevant features were extracted from the URLs.
These included **URL-based features** (length, presence of IP address, special characters), **domain-related features** (WHOIS data, DNS records, registration length), and **content-based features** (usage of HTTPS token, favicon consistency, redirects, and external objects).

2) ***C. Model Training – Random Forest***

3) The extracted features were used to train the ***Random Forest model****, chosen for its robustness, ability to handle imbalanced data, and effectiveness in minimizing overfitting.*

4)

5) The following are the main features extracted for classification:

1. Presence of IP Address – URLs containing direct IP addresses (e.g., `http://192.168.0.1/login`) are highly suspicious.

2. URL Length – Extremely long URLs often attempt to obfuscate malicious intent or hide tracking codes.

3. Use of Shortening Services – Services like *bit.ly* or *tinyurl* are frequently exploited to disguise phishing URLs.

4. Use of "@" Symbol – The portion of the URL before "@" is ignored by browsers, allowing attackers to mislead users.

5. Multiple Redirects (//) – Repeated slashes indicate hidden redirections to other malicious domains.

6. Prefix or Suffix in Domain – Phishers imitate trustworthy domains (like paypal-login-secure.com) by adding prefixes or dashes.

7. Domain Registration Length – Domains with very short registration durations are strong phishing indicators.

8. Favicon Consistency – A mismatched favicon, especially one loaded from an external site, suggests phishing intent.

9. Unusual Ports – Legitimate sites rarely use ports outside standard HTTP (80)/HTTPS (443). Suspicious ports indicate abuse.

10. Fake HTTPS Token in Domain – Misleading domains such as *http://https-secure-login.com* try to appear safe.

11. Request URL Analysis – Checks if resources (images, scripts, media) are loaded from unrelated or malicious domains.

12. Abnormal URL Structure – Inconsistency between the domain name and hosting information suggests manipulation.

13. Server Form Handler (SFH) – Empty or abnormal SFH values indicate fake login or form submissions.

14. Email Submission in Forms – Phishing sites often submit sensitive data directly to attacker emails.

15. Redirection Frequency – The final landing page is obscured by too many redirects.

16. On-Mouse-Over Events – JavaScript tricks hide the true URL shown on the browser status bar.

17. Iframe Usage – Hidden iframes embed invisible malicious content within legitimate-looking pages.

18. DNS Record Availability – Absence of valid DNS records strongly suggests fraudulent domains.

19. PageRank Value – Low or zero PageRank indicates untrusted or newly created domains.

20. Google Index Status – Legitimate domains are usually indexed, while phishing domains are often absent.

21. Number of External Links – Very few or no external references indicate fake isolated websites.

22. HTTPS Presence (SSL Certificate Validity) – Phishers may mimic HTTPS invalid or expired certificates are suspicious.

23. SSL Certificate Issuer Check – Certificates from untrusted or free issuers (e.g., self-signed) raise concerns.

24. Domain Age – Phishing campaigns often use domains that were just formed.

25. WHOIS Registration Consistency – Fake or missing WHOIS data indicates malicious intent.

26. Suspicious Subdomain Count Multiple nested subdomains (e.g., *login.verify.bank.secure.com.fake.net*) raise red flags.

27. Presence of Numeric Characters in Domain – Attackers often insert numbers (e.g., *paypa1.com*) to mimic trusted domains.

28. Use of Special Characters in URL – Symbols like `-`, `=`, or `%` are often misused in phishing URLs.

29. Domain Ownership Concealment – Use of private/proxy WHOIS services is common among phishing domains.

30. Anchor Tag Consistency – Fake sites often use anchor tags pointing to empty or unrelated domains.

31. Suspicious JavaScript Functions – Scripts like, or `onLoad()`, can manipulate user actions.

32. Presence of Pop-Ups – Legitimate banking or e-commerce sites rarely use pop-ups for credential entry.

33. URL Containing Sensitive Keywords – Words like *login*, *secure*, *verify*, *update*, or brand names (e.g., *PayPal*, *bank*) in suspicious contexts are typical of phishing attempts.

6) D. Cloud Deployment with AWS EC2

Once trained, the Random Forest model was integrated into a phishing detection application built using **Flask**. The application was deployed on **AWS EC2,** enabling scalability and real-time URL classification. The EC2 instance acts as the backend server, processing user requests and returning classification results.

7) E. Web Hosting via Web-Space-Kit

To make the system accessible to users, the phishing detection application hosted on AWS EC2 was linked to a **public domain through Web-Space-Kit.** This provided a user-friendly web interface where users can enter URLs, which are dynamically analysed by the model, and classified as **phishing** or **legitimate.**

4. Results and discussions

The main goal of this project was to develop a phishing detection model capable of accurately classifying URLs as phishing or legitimate based on extracted features.

A. Model Evaluation

The models were evaluated using standard machine learning performance metrics such as Accuracy, Precision, Recall, F1-Score, and False Positive Rate (FPR). Additionally, confusion matrices were generated for each algorithm to provide deeper insights into the distribution of predictions across true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

B. Accuracy Score

The accuracy score provides a quick, overall indication of model performance by calculating the proportion of correctly predicted instances among the total predictions. However, accuracy alone may not be sufficient to assess the reliability of phishing detection, so it was interpreted alongside precision, recall, and F1-score.

Figure 2: Comparative Analysis of Algorithms

Model	Feature Set Used	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random Forest	33 Features (Auto)	97.8	98.1	97.5	97.8
XGBoost	33 Features (Auto)	96.9	97.2	96.3	96.75
Logistic Regression	33 Features (Auto)	93.4	92.7	93.8	93.25
Decision Tree	33 Features (Auto)	92.1	91.9	91.8	91.85
SVM	33 Features (Auto)	91.6	90.2	92	91.1

The developed web-based interface allowed users to input a URL, which was then analysed by the deployed machine learning model hosted on AWS EC2 and integrated with a domain purchased through Web-Space-Kit. The system provided:

1. Prediction Output- Whether the URL is phishing or legitimate.

2. Reason Explanation Details- on why a URL was flagged (e.g., presence of special characters, numeric values in the domain, use of http instead of https, shortened URLs, excessive subdomains, or suspicious symbols like @).

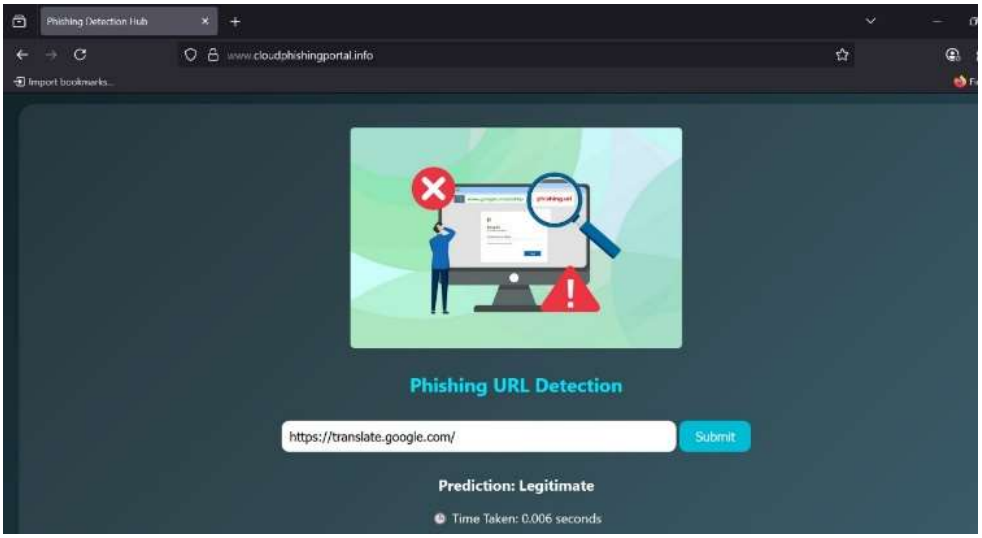
3. Execution Time: The time taken by the system to process and classify the URL, allowing users to see how quickly predictions are generated.

C. Observations

Random Forest is the most suitable model for deployment in real-world phishing detection due to its high accuracy, high precision, and low false positive rate. XG-Boost performs very closely to Random Forest and could serve as a strong alternative, especially when fine-tuned for performance optimization.

Figure 3: Execution of Phishing Output

Figure 4: Execution of Legitimate Output



5.Conclusion

The accuracy achieved in the XG-Boost, Logistic Regression, and Random Forest models for this study is 96.9%, 93.4%, and 97.8%, respectively. Among these, the Random Forest model demonstrated the highest accuracy and lowest false positive rate, making it the most reliable choice for final deployment in detecting phishing websites. The integration of the system into a cloud-based infrastructure via AWSEC2 and its public hosting through a secure domain enhances accessibility, scalability, and usability for real-time phishing detection. The system's ability to provide detailed explanations for classification—such as identifying long URLs, the presence of special characters, unsecured HTTP protocols, or suspicious numerical patterns—adds transparency and trustworthiness to the results.

The primary defence against phishing remains a combination of robust technological tools and user awareness. Educating users to verify URLs, avoid clicking unverified links, and remain cautious when entering sensitive information is crucial. Looking ahead, the framework can be enhanced to continuously monitor active browsing sessions in the background, autonomously flagging and blocking malicious URLs before they are accessed. This would further strengthen protection against phishing attacks while maintaining a seamless user experience.

In conclusion, the **Machine Learning Based Phishing Website Detection System via Cloud** successfully combines high-performing machine learning models with scalable cloud deployment to create a reliable, transparent, and user-friendly security solution. By merging technological innovation with user education, the proposed system provides a comprehensive approach to mitigating the growing threat of phishing attacks, making it a valuable asset in today's cybersecurity landscape.

II. References

[1] C. Gu, 2021, "A Lightweight Phishing Website Detection Algorithm by Machine Learning," 2021 International Conference on Signal Processing and Machine Learning (CONF-SPML), pp. 245–249.

[2] J. Tanimu and S. Shiaeles, 2022, "Phishing Detection Using Machine Learning Algorithm," 2022 IEEE International Conference on Cyber Security and Resilience (CSR), pp. 282–287.

[3] U. Zara, K. Ayub, H. U. Khan, A. Daud, et al., 2024, "Phishing Website Detection Using Deep Learning Models," IEEE Access, vol. 12, pp. 1–12.

[4] R. S. Rao and S. T. Ali, 2015, "PhishShield: A Desktop Application for Detecting Phishing Webpages Using Heuristic Techniques," Procedia Computer Science, vol. 54, pp. 147–156.

[5] H. Sampat, M. Shankar, A. Pandey, and H. Lopes, 2018, "Detection of Phishing Websites Using Machine Learning Approaches," International Research Journal of Engineering and Technology (IRJET), vol. 5, no. 3, pp. 2500–2504.

[6] S. C. Jeeva and E. B. R. Singh, 2016, "Intelligent Phishing URL Detection Using Association Rule Mining," International Journal of Computer Applications, Karunya University, India.

[7] S. A. Al-Saaidah, 2017, "Detecting Phishing Emails Using Machine Learning Techniques," Middle East University, Department of Computer Science.

[8] R. B. Basnet, A. H. Sung, and Q. Liu, 2014, "Learning to Detect Phishing URLs International Journal of Research in Engineering and Technology (IJRET), Colorado Mesa University, USA.

[9] A. K. Jain and B. B. Gupta, 2017, "Phishing Detection: Analysis of Visual Similarity-Based Approaches," Security and Communication Networks, vol. 2017, Hindawi.

[10] J. Mao, J. Bian, W. Tian, S. Zhu, T. Wei, A. Li, and Z. Liang, 2018, "Detecting Phishing Websites via Aggregation Analysis of Page Layouts," Procedia Computer Science, vol. 129, pp. 224–230.

[11] R. Kiruthiga and D. Akila, 2019, "Phishing Website Detection Using Machine Learning Techniques," International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 2S11, pp. 123–127.

[12] M. Chatterjee and A. S. Namin, 2019, "Detecting Phishing Websites through Deep Reinforcement Learning," IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), pp. 536–541.

[13] M. E. Pratiwi, 2018, "Phishing Site Detection Analysis Using Artificial Neural Networks," Journal of Physics: Conference Series, vol. 1140, no. 1, doi:10.1088/1742-6596/1140/1/012048. [14] R. Mahajen and I. Siddavatam, 2018, "Detection of Phishing Websites Using Machine Learning Algorithms," International Journal of Computer Applications (IJCA), vol. 182, no. 12, pp. 1–6.

[15] A. K. Dutta, 2021, "Phishing Website Detection Using Machine Learning Techniques," Open Access Journal of Information Security, vol. 12, pp. 15–22.

[16] N. Md. Norzaidah and M. N. Bin, 2021, "Phishing Website Detection Using Random Forest in Cloud-Based Environments," 2nd International Conference on Artificial Intelligence and Data Sciences (AiDAS), pp. 134 139.

[17] A. Al swailem and B. Alabdullah, 2020, "Deep Learning Approach for Phishing Website Detection in Cloud Platforms," International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 5, pp. 120–125.