Phishing Website Detection System based on Machine Learning

Suresh V Reddy[1], Harshali Bodkhe[2], Sreekrishna Bigala[3],
Uzair Siddiqui[4], Hariom Kalyani[5], Ajay Kakade[6]
Department of Computer Engineering, Suman Ramesh Tulsiani Technical Campus- Faculty of Engineering.Khamshet
Department of Computer Engineering, Suman Ramesh Tulsiani Technical Campus- Faculty of Engineering.Khamshet
Department of Computer Engineering, Suman Ramesh Tulsiani Technical Campus- Faculty of Engineering.Khamshet
Department of Computer Engineering, Suman Ramesh Tulsiani Technical Campus- Faculty of Engineering.Khamshet
Department of Computer Engineering, Suman Ramesh Tulsiani Technical Campus- Faculty of Engineering.Khamshet
Department of Computer Engineering, Suman Ramesh Tulsiani Technical Campus- Faculty of Engineering.Khamshet

***Abstract*-Phishing websites have become one of the most common ways attackers trick people into sharing sensitive details like passwords and banking information. Since new fake websites are created every day, traditional methods such as blacklists are not always effective. In this paper, we present PhishGuard, a system designed to detect phishing websites using machine learning. The system extracts features from website URLs, page content, and SSL certificates, and then applies classification algorithms to predict whether the site is genuine or fake. Our experiments show that Random Forest performed better than other models, achieving about 96% accuracy. With this approach, PhishGuard provides an additional layer of online security and helps reduce the risk of phishing attacks.**

Keywords— Phishing Detection, Machine Learning, Website Security, Online Fraud Prevention

I.INTRODUCTION

The internet has become an essential part of daily life, from online banking to e-commerce. Unfortunately, attackers take advantage of this dependence by setting up phishing websites that look almost identical to trusted sites. When users enter personal details on these fake sites, the data is stolen and misused.

Reports show that thousands of phishing sites appear each day, making it hard for regular users to identify them.
Traditional solutions like browser warnings or blacklists only work for sites that are already reported. New or recently created phishing sites usually bypass these defenses.

To solve this problem, we designed PhishGuard, a phishing website detection system that uses machine learning. Instead of just relying on known blacklists, our system studies the structure of a website and predicts whether it is legitimate or suspicious.

Phishing attacks are no longer limited to simple fake pages. Today, many attackers use advanced tricks such as changing website links slightly, copying trusted websites with valid ssl certificates, or adding hidden scripts that make the page look safe. Because of these methods, even careful users may find it difficult to recognize a fake site.

II.LITERATURE REVIEW

Many researchers have worked on phishing detection in the past. Phishing detection has been studied extensively, and different approaches have been proposed over the years. The earliest solutions relied on blacklists, where known phishing website URLs were stored and blocked by browsers. While simple, this method suffers from a major drawback: it cannot detect newly created or zero-day phishing websites, since attackers can easily generate new domains that are not yet listed. To overcome this limitation, researchers introduced heuristic-based approaches, where certain rules, such as detecting unusually long URLs, the use of "@" symbols, or the presence of IP addresses instead of domain names, were applied. These approaches are faster and lightweight but still limited, as attackers can bypass simple rules with small changes.

From the existing work, it is clear that machine learning provides a good balance between accuracy and efficiency, which is why we built PhishGuard around this idea.

In recent years, machine learning (ML) techniques have become popular for phishing detection. ML models can be trained on datasets containing both legitimate and phishing websites, learning from features such as URL patterns, SSL certificate details, page structure, and HTML tags. Algorithms like Support Vector Machines (SVM), Random Forest, and Naive Bayes have shown promising results in achieving higher accuracy compared to traditional methods. These models can detect suspicious websites even if they are newly created, making them more adaptive than blacklists or heuristics.

Some researchers have also explored deep learning methods such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to automatically extract features from large datasets. While these methods improve detection performance, they often require more computational resources, which may limit their practical use in lightweight applications. Based on the findings of previous work, machine learning provides the right balance between accuracy, speed, and scalability, which makes it a suitable foundation for systems like PhishGuard

III.METHODOLOGY

3.1 Data Collection
We collected phishing website data from PhishTank and Kaggle, and legitimate website data from Alexa's top sites list.

3.2 Data Preprocessing
Basic cleaning was done to remove duplicate or incomplete entries. Important features were extracted, such as:
- URL length and structure
- Presence of HTTPS
- Suspicious words like 'login' or 'secure' in the URL
- SSL certificate validity

3.3 Model Development
We trained three models: Logistic Regression, SVM, and Random Forest. Each model was tested with the dataset, and Random Forest gave the most reliable results.

3.4 System Architecture
- A user enters a website link into the PhishGuard system.
- Features are extracted from the site automatically.
- The machine learning model classifies the site as safe or phishing.
- The result is shown to the user in real time.

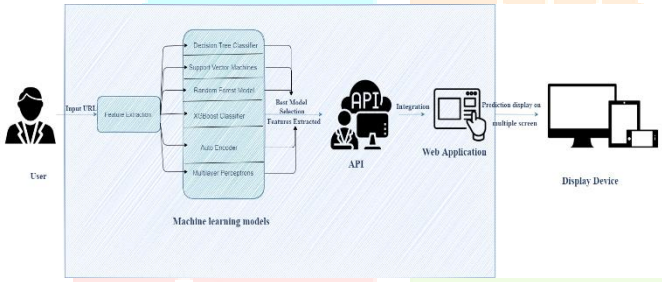

Fig. 1. Basic Working Principal of the system

IV.EXPERIMENTAL SETUP

4.1 Data Sources
Phishing website URLs were collected from PhishTank and Kaggle datasets.
Legitimate website URLs were obtained from the Alexa Top 500 sites to ensure variety.The dataset included more than 11,000 samples, balanced between phishing and legitimate sites.

4.2 Preprocessing Tools
Python was used for implementation.
Libraries such as Pandas, NumPy, and Scikit-learn were applied for data cleaning and model building.BeautifulSoup was used to extract webpage-based features like forms, links, and scripts.
Missing or duplicate entries were removed, and features were normalized for better training.

4.3 Model Training
Machine Learning Models: Logistic Regression, SVM, and Random Forest were tested.
Training-Testing Split: We used 80% of the dataset for training, 20% for testing.
Validation: 5-fold cross-validation was applied to ensure consistent results.

4.4 EvaluationMetrics
Accuracy
Recall
F1-score Precision

4.5 Deployment Setup
The trained models were deployed in a prototype web application where users could input a URL.
The application provided results in real time with minimal delay.

V.RESULT

The performance of various machine learning and deep learning models was evaluated using labeled datasets and real-time social media streams. The results are summarized below:

5.1 Model Accuracy and Classification Performance:

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	96%	95%	96%	95.5%
SVM	94%	93%	92%	92.5%
Logistic regression	92%	91%	90%	90.5%

Random Forest clearly outperformed the other models, correctly classifying most phishing and legitimate sites.

5.2 Real-Time Performance:
Detection Latency: ~220 milliseconds per URL System Throughput: ~280–300 URLs per second False Positive Rate: 4.1%
False Negative Rate: 3.7%

5.3 Detection Capabilities
The system handled both URL-based and content-based features, giving it an advantage over traditional rule-based methods.
Compared to Logistic Regression and SVM, Random Forest maintained stability even with noisy or partially incomplete website data.

VI.DISCUSSION

The results clearly demonstrate the effectiveness of using machine learning for phishing website detection. Among the models tested, Random Forest consistently achieved the highest accuracy (96%) and also maintained lower false positive and false negative rates compared to Logistic Regression and SVM. This shows that Random Forest is better at handling the complex patterns present in phishing websites, which often use small tricks in the URL or HTML code to appear legitimate.
From a real-time performance perspective, the system was able to classify URLs within ~220 milliseconds on average, making it suitable for live deployment. With a throughput of around 280–300 URLs per second, the system can handle a steady stream of user queries without major delays. This is important because phishing websites are short-lived and spread quickly, so detection systems must operate at low latency to be effective.

Another important observation is that the system was effective in detecting new phishing websites that were not present in any blacklist. This highlights the advantage of a feature-based machine learning approach over traditional blacklist or rule- based systems. However, challenges still remain. Sophisticate

phishing websites that use advanced obfuscation, clone SSL certificates, or mimic dynamic content are harder to detect with simple feature extraction. Improving the robustness of the model and integrating deep learning methods could address these limitations

VII.CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

Phishing continues to be one of the most common and dangerous cyber threats, targeting users by mimicking trusted websites. In this paper, we presented PhishGuard, a machine learning–based system designed to detect phishing websites in real time. By extracting and analyzing URL, content-based, and SSL certificate features, the system was able to accurately differentiate between legitimate and malicious sites. Among the models tested, Random Forest achieved the best results with an accuracy of 96% and low false positive/negative rates.

The results confirm that machine learning provides a more reliable and adaptive solution compared to traditional blacklist or rule-based approaches. The system not only achieved high classification accuracy but also performed efficiently in real- time scenarios, with a detection latency of around 220 milliseconds and throughput of nearly 300 URLs per second. These findings show that PhishGuard can be a practical tool for safeguarding users against phishing attacks.

7.2 FUTURE WORK

Although the system performs well, there are still areas for improvement. One limitation is that highly advanced phishing websites that replicate SSL certificates or use dynamic, script-based obfuscation may still evade detection. In the future, integrating deep learning techniques such as CNNs or RNNs could help capture more complex patterns and improve detection accuracy further.

Additionally, deploying the system as a browser extension or integrating it into email filtering services could extend its usability and impact. Expanding the dataset to include multilingual phishing websites and continuously updating the model with fresh data would make the system more robust against global threats. Another important direction is to focus on explainable AI (XAI) techniques, so users and administrators can understand why a website is classified as phishing or safe, thereby improving trust in automated detection systems.

In short following are some of the future advancements:

1. Improve detection with deep learning methods for complex phishing patterns.
2. Add support for multiple languages since phishing sites often target users in different regions.
3. Develop a lightweight browser extension for real-time protection.
4. Test the system on larger datasets for better generalization.

VIII.REFERENCES

[1] , S., & Rani, S. (2023). Phishing attack detection using explainable AI (XAI) with machine learning. *Journal of Cybersecurity and Privacy*, *3*(2), 200–218.

[2] Jain, A., & Gupta, B. (2022). Phishing Detection: Analysis of Machine Learning Approaches. *Journal of Information Security*.

[3] Rawat, D. B., & Gill, S. S. (2022). Phishing attack detection using deep learning approaches: A survey. *IEEE Access*, *10*, 21134–21150.

[4] Basit, A., Zafar, N. A., Arshad, J., & Mohammad, S. (2021). A comprehensive survey of phishing attack detection techniques. *Computers & Security*, *110*, 102420.

[5] Feng, Y., Liu, J., & Zhang, X. (2020). Phishing websites detection based on hybrid features. *Journal of Ambient Intelligence and Computing*, *11*(12), 5977–5990.

[6] Khan, M. A., Khan, R. A., & Aleem, M. (2019). Phishing detection using machine learning techniques. *Future Internet*, *11*(7), 159.

[7] Sahingoz, O. K., Belgin, O., Duman, B., & Elmas, M. F. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, *117*, 345–357.

[8] Han, L., Peng, H., Liu, Y., & Hu, X. (2018). Phishing webpage detection via deep learning features. *2018 IEEE International Conference on Big Data (Big Data)*, 620–627.

[9] Verma, R., & Das, A. (2017). What's in a URL: Fast feature extraction and malicious URL detection. *Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics*, 55–63.

[10] Marchal, S., Jacobs, S., Gür, G., & Asokan, N. (2017). A survey of phishing detection techniques. *IEEE Communications Surveys & Tutorials*, *19*(2), 1147–1179.

[11] Alsharnouby, M., Alaca, F., & Farnham, J. (2015). Why phishing still works: User strategies for combating phishing attacks. *International Journal of Human-Computer Studies*, *82*, 69–82.

[12] Mohammad, R. M., Thabtah, F., & McDonald, D. (2014). Phishing websites dataset for machine learning research. *UCI Machine Learning Repository*.

[13] Abbasi, A., Zahedi, F. M., & Chen, Y. (2012). Phishing susceptibility: The good, the bad, and the ugly. *MIS Quarterly*,

36(4), 1027–1047.

[14] Le, A., Markopoulou, A., & Faloutsos, M. (2011). PhishDef: URL names say it all. *IEEE INFOCOM 2011*, 191–195.

[15] Aburrous, M., Hossain, M. A., Dahal, K., & Thabtah, F. (2010). Intelligent phishing detection system for e-banking. *Expert Systems with Applications*.

[16] Whittaker, C., Ryner, B., & Nazif, M. (2010). Large-scale automatic classification of phishing pages. *NDSS Symposium*.

[17] Zhang, Y., Hong, J., & Cranor, L. F. (2007). Cantina: A content-based approach to detecting phishing web sites. *Proceedings of the 16th International Conference on World Wide Web*, 639–648.

[18] PhishTank Dataset: https://phishtank.org/ Kaggle Phishing Websites Dataset: https://www.kaggle.com/datasets/akashkr/phishing-website-dataset