

PhishGuard: ML-Based Phishing URL Detector with Virus Total API & Chrome Extension

Prof. Dr. Sulochana Sonkamble, Sujal Bhangale, Amol Matsagar, Samaksh Parate, Vishal Mane

Department of Computer Engineering, JSPM Narhe Technical Campus, Pune, Maharashtra, India

Abstract - In recent years, phishing attacks have been recognized as one of the most critical cybersecurity threats facing individuals, organizations, and financial institutions in the modern digital world. The attacks are based on exploiting human nature, wherein attackers trick users to access malicious websites, which are replicas of genuine websites, to steal confidential information such as login information, financial information, and personal information. However, it has been recognized that traditional methods such as blacklist-based and rule-based systems are no longer effective against modern phishing attacks, such as zero-day attacks, which are highly sophisticated and dynamic in nature.

In this direction, this research proposes an innovative phishing URL detection system called PhishGuard, which utilizes machine learning to identify malicious phishing URLs. The proposed system is based on an extensive set of features, which include more than 100+ features (total 111 features), such as lexical, domain, and security-related features of URLs. The proposed system utilizes supervised machine learning models such as Random Forest, Logistic Regression, and Decision Trees to improve the accuracy of phishing URL detection. The proposed system also utilizes an external validation approach based on the VirusTotal API to improve its accuracy. The proposed system is light-weight, scalable, and can be deployed as a web application and browser extension. The proposed system has shown promising results in terms of improving phishing URL detection accuracy with minimal dependency on traditional static methods.

Key Words: Phishing Detection, Machine Learning, URL Analysis, Cybersecurity, Flask Web Application, VirusTotal API

1. INTRODUCTION

With the rapid growth of internet usage, the number of cybercrimes is on the rise. Among the different types of cybercrimes, phishing is one of the most common and dangerous types of cybercrime. Phishing is a type of cybercrime where attackers pretend to be legitimate users or organizations and trick victims into revealing their secret information like user IDs, passwords, credit card numbers, etc.

Nowadays, phishing is no longer just about attackers trying to create fake websites. Phishing attackers are using sophisticated techniques like:

- URL Obfuscation.
- Homographs.
- HTTPS Spoofing.
- Shortened URLs.
- Domain Spoofing.

Using these techniques, attackers have made phishing websites indistinguishable from legitimate ones.

Traditionally, phishing is detected by techniques like:

- Blacklists.
- Rule-Based Systems.

But these techniques have some disadvantages like:

- Unable to detect zero-day attacks.
- Need for continuous updation.
- Easy to evade by attacker.

To overcome these disadvantages, machine learning techniques have been proposed as an effective solution for phishing attacks. Machine learning techniques can learn from large datasets and detect phishing URLs based on their features.

The proposed system, PhishGuard, is based on the multi-feature machine learning technique. This technique extracts 111 features from URLs and uses different machine learning algorithms for classification.

Objectives of the Research:

The objectives of the research are as follows:

- To design an intelligent phishing detection system.
- To improve the accuracy of the proposed system.
- To provide real-time detection.
- To provide a practical solution.

2. Body of Paper

2.1 Literature Review

The field of phishing detection has been a research area for many years. Several approaches have been proposed, ranging from traditional blacklists to more advanced machine learning methods.

The first approaches to detecting phishing sites were based on blacklists. Blacklists store a list of known phishing URLs and compare them to the input URL. However, such approaches were unable to identify new, generated phishing URLs.

One of the first frameworks in the field of phishing detection using URL features was proposed by Garera et al. in 2007. Their research focused on the structural features of URLs.

Aburrous et al. proposed a fuzzy data mining method for detecting phishing sites in e-banking systems in 2010. Their system was effective but lacked scalability and had to deal with complex rule definitions.

Another method proposed by Jain and Gupta in 2017 focused on visual similarity-based phishing detection. This method identifies phishing sites based on their similarity to legitimate sites. However, such methods require image processing and are computationally expensive.

Recent research has focused on machine learning methods, such as:

Decision Trees
Random Forests
Support Vector Machines
Neural Networks

These methods have achieved high accuracy in detecting phishing sites using URL features.

However, several challenges persist in the field of phishing detection. These challenges are:

Complexity in feature selection
Imbalanced datasets
Real-time processing limitations

The proposed system will address the above challenges.

2.2 Problem Statement

Phishing attacks continue to pose a serious threat despite the presence of existing detection systems. Traditional approaches are insufficient due to their inability to detect newly generated phishing URLs.

The major problems include:

- Ineffectiveness against zero-day attacks
- High dependency on blacklists
- Evasion techniques used by attackers
- Lack of real-time detection

Therefore, there is a need for a system that:

- Can detect phishing URLs dynamically
- Does not rely on pre-existing databases
- Works efficiently in real-time

2.3 Methodology

The methodology employed by the PhishGuard system is based on a series of stages:

4.1 Data Collection

The datasets are collected from:

PhishTank
UCI Repository
Open datasets

4.2 Data Preprocessing
Removal of duplicate data
Handling missing values
Normalization

4.3 Feature Extraction

A total of 111 features are extracted:

Lexical Features
URL Length
Number of Dots
Presence of Special Characters
Presence of IP Address
Domain Features
Domain Age
DNS Records
WHOIS Records
Security Features
HTTPS
SSL Certificate Validity
Behavioral Features
Redirects
Google Indexing

4.4 Model Training

Models employed:

Random Forest (main model)
Logistic Regression
Decision Tree

4.5 Model Evaluation

Metrics used:

Accuracy

Precision

Recall

F1-score

4.6 Deployment

Flask web app

Chrome extension

VirusTotal API integration

2.5 System Architecture

The architecture of the proposed PhishGuard model is designed in such a way that it can provide an efficient, scalable, and real-time phishing detection mechanism. In the proposed architecture, the design is based on the modular and layered approach, where each layer is responsible for performing specific operations in the phishing detection process.

The entire workflow of the proposed system can be defined as follows:

User Input → Feature Extraction → Machine Learning Classification → API Validation → Result Output

5.1 Input Layer

The input layer acts as the first point of entry for the system. At this point, the URLs are provided to the system or retrieved from external sources such as emails, browsers, or security tools. This layer is responsible for processing different types of input data, which may include user-provided URLs as well as URLs retrieved from other sources such as browser activities.

In order to maintain uniformity in data, the input URLs are validated and then standardized. This involves removing unnecessary spaces in the URL and confirming whether the input data is a valid URL or not.

5.2 Feature Extraction Layer

The feature extraction layer is arguably the most important layer in the system. Within this layer, the input URL will be analyzed in order to identify significant features that will aid in the differentiation between legitimate and phishing URLs.

The system has been able to identify a comprehensive range of 111 features, which fall within the following categories:

Lexical Features - These features will include the URL length, the number of special characters in the URL, the number of subdomains in the URL, the presence of suspicious keywords in the URL, and the use of IP addresses in the URL.

Domain-Based Features - These features will include the domain age, the domain registration information, the DNS records, and the WHOIS information.

Security Features - These features will include the presence of HTTPS in the URL, the SSL certificate validity, and the encryption information.

Behavioral Features - These features will include the URL redirection information, the URL shortening service information, and the search engine information.

5.3 Machine Learning Layer

The machine learning layer is responsible for classifying the features obtained from the system as phishing or legitimate. This layer makes use of various supervised machine learning algorithms to improve its accuracy and efficiency. The main machine learning algorithm employed in this system is Random Forest, which is highly efficient in handling large numbers of features. The algorithm is highly accurate as well. Apart from this, Logistic Regression and Decision Tree algorithms are also employed for validation purposes.

The machine learning models are trained on various labeled datasets, enabling them to learn patterns from phishing URLs. The models can then be employed to make predictions in real-time. The output of this layer is classified as a feature with a probability value indicating the likelihood of it being a phishing URL.

5.4 API Integration Layer

In order to increase the reliability of the system in detecting potential threats, the system will be able to integrate with different security services such as the VirusTotal API.

This will be used as an additional validation mechanism.

When the URL is being processed in the system, it will be sent to the API, which will then check the URL with different security engines.

The response will contain threat intelligence information that will be used in conjunction with the machine learning prediction.

5.5 Output Layer

The output layer shows the result to the user in a presentable and user-friendly way. It will show the user if the URL is “Phishing” or “Legitimate.” It will also show additional information like the risk score and the confidence level.

This layer will be used in:

Web Applications (Using Flask interface)
Browser Extensions
Email Filtering Systems

The output is generated in real-time, giving the user instant feedback before they interact with the URL.

5.6 System Advantages

The proposed architecture has several advantages:

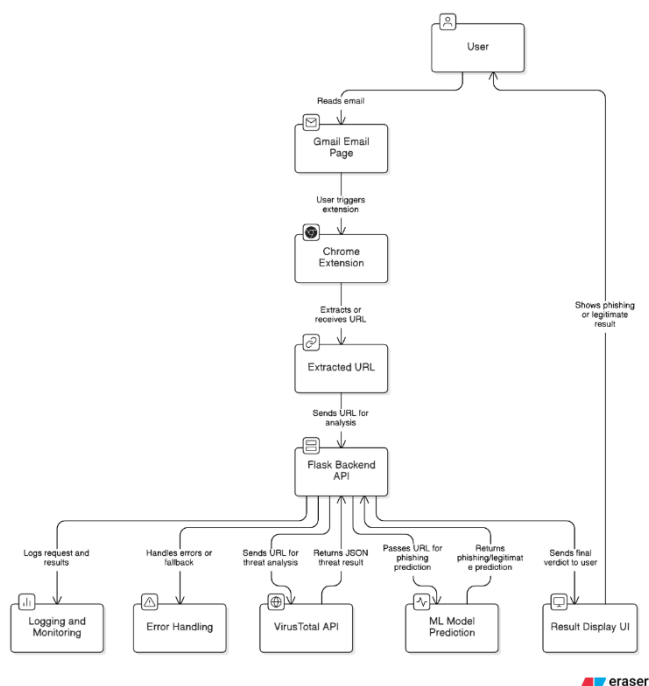
Modularity: Each layer in the architecture operates independently.

Scalability: It is suitable for large-scale deployment.

Real-Time Detection: It can perform instant detections.

High Accuracy: It uses ML algorithms and API validation.

Integration Capability: It can be integrated with various platforms.



2.6 Machine Learning Models

The performance of the proposed PhishGuard system relies on the performance of the machine learning algorithms used in the classification process. In the

proposed system, multiple machine learning models are used for the classification process to ensure the accuracy, robustness, and generalization of the model.

The main machine learning models used in the proposed system include the Random Forest model, the Logistic Regression model, and the Decision Tree model. Each model has its advantages in the phishing detection process.

Model Comparison and Selection

The application of different machine learning models enables the comparative evaluation based on the accuracy, precision, recall, and F1-score.

Random Forest model has the highest accuracy compared to other models because it is based on the concept of ensembling

Logistic Regression has the advantage of faster computation and easier understanding

Decision Tree has the advantage of easier understanding.

Based on the experimental evaluation, the Random Forest model has been chosen as the final model for deployment.

2.7 Methodology

The methodology of the proposed PhishGuard system involves multiple stages, starting from data collection to final classification.

1. Data Collection

Datasets for phishing and legitimate URLs will be collected from sources such as:

- PhishTank
- UCI Machine Learning Repository
- Open-source cybersecurity datasets

These datasets will include labeled URLs categorized as phishing or legitimate.

2. Data Preprocessing

The collected data will undergo preprocessing steps such as:

- Removal of duplicate entries
- Handling missing values
- Normalization of URL formats

3. Feature Extraction

A total of multiple features will be extracted from URLs, categorized as:

- Lexical features (URL length, symbols, keywords)
- Domain features (domain age, DNS records)
- Security features (SSL certificate, HTTPS usage)

4. Model Training

Supervised machine learning algorithms such as:

- Random Forest
- Logistic Regression
- Decision Tree

will be trained on the dataset.

5. Model Evaluation

The performance of the model will be evaluated using:

- Accuracy
- Precision
- Recall
- F1-score

6. Deployment

The trained model will be integrated into:

- Flask-based web application
- Browser extension

This enables real-time phishing detection.

2.8 Performance Metrics

To evaluate the effectiveness of the proposed system, several performance metrics are used:

- **Accuracy:** Overall correctness of the model
- **Precision:** Correct phishing predictions out of total predicted phishing

- **Recall:** Ability to detect actual phishing URLs
- **F1-Score:** Balance between precision and recall

These metrics help in analyzing the robustness and reliability of the model.

2.9 Limitations

Despite the effectiveness of the proposed PhishGuard system in detecting phishing URLs, there are certain limitations that need to be acknowledged. The first limitation is the fact that the system is based mainly on URL-based feature analysis; therefore, it is not able to carry out content-based feature analysis of the website. As a result of this limitation, it is not able to identify phishing websites with highly sophisticated content.

Another limitation is the fact that the system is based on the quality of the dataset used in the training process. The machine learning model is able to perform well only when it is trained using large datasets. As a result of the limitation in the dataset used in the machine learning model, the system is not able to generalize well.

In addition, the system is likely to produce false positives and false negatives. A false positive is the occurrence where the system is able to identify a legitimate URL as a phishing URL. The occurrence of false positives is likely to affect the trust of the users. A false negative is the occurrence where the system is not able to identify a phishing URL as such. The occurrence of false negatives is likely to affect the security of the users.

The system is likely to be affected by certain limitations as a result of the fact that it is dependent on the VirusTotal API. The VirusTotal API is likely to be affected by certain limitations such as the rate at which it is able to respond. The system is likely to be affected by the fact that the attackers continue to evolve their attack methods.

2.10 Future Enhancements

The PhishGuard system can be used as a solid base for phishing detection. However, there are many areas for improvement in the future to make it even better.

Firstly, one improvement can be made to PhishGuard in the future is to integrate deep learning algorithms into the system. Deep learning algorithms such as Artificial Neural Networks (ANNs) and Recurrent Neural Networks (RNNs) can be used for improving phishing

detection accuracy. These algorithms have the ability to learn complex relationships between data and can perform better than traditional machine learning algorithms.

Another improvement can be made to PhishGuard in the future is to integrate content-based phishing detection into the system. This will involve analyzing the content of a webpage instead of just analyzing the URL. This will make PhishGuard more robust as a phishing detection tool.

Another improvement can be made to PhishGuard in the future is to integrate Natural Language Processing (NLP) into the system for detecting phishing attacks in emails or messages. This will make PhishGuard more robust as a phishing detection tool as it will be able to analyze emails and messages as well.

Another improvement can be made to PhishGuard in the future is to integrate a continuous learning algorithm into the system. This will make PhishGuard more robust as a phishing detection tool as it will be able to keep learning and updating itself for better accuracy in the future.

Another improvement can be made to PhishGuard in the future is to develop a browser extension for PhishGuard. This will make PhishGuard more robust as a phishing detection tool as it will be able to protect users while browsing the internet.

Another improvement can be made to PhishGuard in the future is to integrate PhishGuard into enterprise-level security systems or cloud-based platforms. This will make PhishGuard more robust as a phishing detection tool as it will be able to protect users while browsing the internet.

3. CONCLUSION

In the above paper, the conceptual framework of PhishGuard, a machine learning-based phishing URL detection system, has been discussed. The proposed system has been designed to overcome the drawbacks of existing security systems. By employing feature-based analysis, the proposed system has the potential to become an effective phishing URL detection system.

From the proposed system, it is clear that machine learning has the potential to become an effective tool in identifying phishing URLs and increasing the awareness

of users. The proposed system has the potential to be implemented in real-world scenarios in various domains.

In the future, the proposed system will be implemented and enhanced in terms of feature engineering, deep learning, and content-based phishing detection.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the faculty members and mentors for their continuous guidance, valuable suggestions, and support throughout the development of this research work. Their insights have been instrumental in shaping the conceptual framework of the proposed system.

The authors also acknowledge the availability of publicly accessible datasets and research resources that contributed significantly to understanding phishing detection techniques. Special thanks are extended to the open-source community for providing tools and libraries that facilitated the conceptual design of the system.

Finally, the authors are grateful to their institution for providing the necessary environment and resources to carry out this work successfully.

REFERENCES

- [1] A. K. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity based approaches," *Security and Communication Networks*, vol. 2017, pp. 1–20, 2017.
- [2] M. Aburrous, M. A. Hossain, F. Thabtah, and K. Dahal, "Intelligent phishing detection system for e-banking using fuzzy data mining," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7913–7921, 2010.
- [3] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and measurement of phishing attacks," in *Proceedings of the 2007 ACM Workshop on Recurring Malcode*, pp. 1–8, 2007.
- [4] R. Verma and A. Das, "What you tweet can reveal: A study of phishing detection using machine learning," in *Proceedings of the International Conference on Data Mining Workshops*, 2017, pp. 1–8.
- [5] UCI Machine Learning Repository, "Phishing Websites Dataset,"
- [6] PhishTank, "Phishing URL Dataset,"
- [7] Google Safe Browsing, "Safe Browsing API,"
- [8] VirusTotal, "VirusTotal Public API,"