



# Phishing Web Sites Detection Using Machine Learning Project.

<sup>1</sup> Anand S, <sup>2</sup> Gagan R, <sup>3</sup>, Kartik Naik, <sup>4</sup> D M Likhith, <sup>5</sup> Prof. Aravind Naik, <sup>6</sup> Dr. Suresha D, <sup>7</sup> Prof. Shreya Shetty

<sup>1</sup>Student, <sup>2</sup> Student, <sup>3</sup> Student, <sup>4</sup> Student, <sup>5</sup> Assistant Professor, <sup>6</sup> Head of the Department, <sup>7</sup> Assistant Professor

Computer Science And Engineering,

Shrinivas Institute Of Technology, Mangalore, India

**ABSTRACT:** Phishing attacks pose a severe cybersecurity risk websites to steal data, such as login credentials and financial information. Traditional detection methods like blacklists and rule-based systems often fail to adapt to rapidly evolving phishing techniques. This study proposes a machine learning (ML)-based solution to identify phishing websites by analyzing URL, domain, and content-based features. A diverse dataset of phishing and benign URLs is preprocessed and used to train multiple supervised learning algorithms. The system is designed for real-time deployment, offering scalability and minimal false positives to enhance user protection.

## I. INTRODUCTION

As digital services like e-commerce, banking, and social networking become more intertwined with daily life, cybersecurity threats have escalated, with phishing standing out as a particularly widespread and dangerous attack method. Phishing attacks are designed to deceive individuals into disclosing confidential information—such as login details, financial credentials, or personal information—by imitating trustworthy websites. Such fraudulent methods frequently include fraudulent emails, misleading URLs, and convincingly designed web pages that closely mirror legitimate sources.

Traditional approaches, such as blacklists and manual verification, typically they are reactive and find it challenging to stay updated with the ever-changing phishing landscape. Because malicious sites can appear and disappear rapidly, conventional detection methods often fail to identify new threats in a timely manner. To counter these evolving tactics, there is a growing need for intelligent, real-time detection systems capable of adapting to new attack strategies.

Machine learning (ML) offers a powerful solution by enabling automated systems to analyse patterns and make predictive decisions on unseen data. By evaluating multiple website attributes—including URL structure, domain registration details, SSL certificate validity, HTML content, and embedded links—ML algorithms can distinguish between authentic and phishing websites with greater accuracy. This study explores the implementation of a machine learning-driven phishing identification system employing Python 3.6.8, assessing multiple supervised learning algorithms to identify the most effective and scalable approach. The ultimate aim is to create an adaptive security solution that enhances online protection and safeguards users from deceptive threats.

## II. ISSUE STATEMENT

Phishing attacks evade traditional defenses by mimicking trusted websites. Rule-based systems lack adaptability, while ML approaches face challenges like imbalanced datasets and computational overhead. This project addresses these gaps by optimizing feature selection and leveraging ensemble learning for higher accuracy.

## III. LITERATURE SURVEY

Jain and Gupta (2018) conducted an early analysis on Machine learning-based phishing detection classifiers. Their research showed that Support Vector Machines (SVM) and Decision Trees could accurately identify phishing URLs by examining lexical and host-based features.

Verma and Das (2020) extended this work by comparing a broader set of algorithms, including Random Forest, Gradient Boosting, and Logistic Regression. Their experiments confirmed that ensemble models significantly improve prediction performance and reduce misclassification rates, especially when trained on a well-balanced dataset.

Sahingoz (2019) introduced a deep learning-based approach that utilized NLP techniques to analyse URL strings as character sequences. Their LSTM and CNN models delivered superior accuracy but also required significant computational resources, which may serve as a limitation in lightweight real-time systems.

Zhou (2021) proposed a hybrid phishing identification framework that combined URL-based features with natural language attributes derived from webpage content. Their method incorporated both supervised learning and attention mechanisms to detect disguised malicious links. Their work emphasized the importance of integrating textual and visual content analysis.

Alam and Patwary (2022) explored the combination of feature engineering and ensemble machine learning in detecting phishing attacks. They applied XGBoost and CatBoost on newly collected datasets containing updated phishing behaviours and found that boosting algorithms consistently outperformed traditional classifiers.

Kumawat (2023) introduced an end-to-end phishing detection pipeline using auto-encoders and anomaly detection techniques. Their unsupervised learning strategy addressed the cold-start problem in phishing detection where new attack patterns lack labelled training data.

Nguyen (2023) proposed a real-time phishing URL identification system that runs in web browsers employing lightweight decision tree classifiers. Their research focused on optimizing prediction speed and low memory consumption while maintaining over 95% accuracy.

#### IV. PROPOSED METHODOLOGY

The proposed phishing detection system follows a structured machine learning pipeline, consisting of six key phases: data acquisition, preprocessing, feature extraction, model selection, performance evaluation, and deployment. The approach prioritizes both classification accuracy and real-world feasibility.

The dataset includes verified phishing and legitimate websites sourced from platforms like PhishTank, OpenPhish, and Kaggle, ensuring diversity in URL patterns and site characteristics. After collection, data preprocessing involves cleaning duplicates, managing missing data, encoding categorical features, and normalizing numerical features.

Feature extraction is performed across three primary categories: URL-based (length, special characters, subdomains, redirection count), domain-based (WHOIS lookup, domain age, DNS record presence), and content-based (JavaScript presence, iframes, embedded login forms). These features capture critical phishing indicators.

For classification, multiple supervised learning models—including Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, and CatBoost—are trained using stratified k-fold cross-validation to enhance generalization and mitigate overfitting.

Evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC serve to evaluate model effectiveness, balancing detection rates with minimizing false positives. The final deployment integrates the best-performing model into a Flask-based web application, enabling real-time phishing classification with MongoDB for query storage. Designed for scalability, the system can be adapted as a browser extension or API endpoint, ensuring practical implementation beyond experimental settings.

#### V. MODULE DESCRIPTION

The system architecture is organized into distinct modules, each performing a specific role in the phishing detection workflow. This modular structure promotes ease of development, testing, and future enhancements.

**1. URL Acquisition Module:** This module is responsible for gathering a diverse collection of web addresses from credible databases. It includes both known phishing links and safe websites to build a balanced and comprehensive dataset suitable for model training.

**2. Data Preparation Module:** Once the data is collected, this module processes it by eliminating errors and inconsistencies. It handles missing values, encodes categorical entries, and applies normalization techniques where applicable to ensure compatibility with the machine learning models.

**3. Feature Analysis Module:** In this phase, a set of characteristics are derived from each URL. These include aspects such as:

- Usage of secure protocols (HTTPS)
- Length and structure of the URL
- Inclusion of suspicious characters or subdomains
- Age of the domain and registrar details
- Presence of hidden scripts or redirect chains

These extracted features are compiled into a structured dataset for training purposes.

#### 4. Model Training Module:

This component runs the core classification algorithms. Multiple algorithms like Random Forest, XGBoost, and SVM are trained using the feature set. Each model undergoes tuning and validation to enhance performance and reliability.

#### 5. Real-Time Prediction Module:

Once the model is finalized, this module handles live classification of input URLs. When a user enters a website link, the system evaluates it and returns a prediction score indicating whether the site is safe or potentially harmful.

#### 6. Logging and Monitoring Module:

All prediction outcomes and query data are stored securely for audit and analysis purposes. This helps in refining the model over time and allows for real-time monitoring of system behaviour.

#### 7. User Interface Layer:

A lightweight web interface, developed with Flask, allows users to interact with the system by submitting URLs and viewing the results. The interface is simple, responsive, and suitable for deployment as a standalone tool or plugin.

## VI. RESULTS AND DISCUSSION

The performance of the phishing website detection system was assessed by training and testing multiple machine learning algorithms on a labelled dataset containing both legitimate and phishing URLs.

### 1. Evaluation Metrics

To quantify model performance, the following metrics were used:

- **Accuracy:** Proportion of total correct predictions.
- **Precision:** Ability to correctly identify phishing websites among all predicted phishing cases.
- **Recall:** Ability to detect actual phishing websites.
- **F1-Score:** Average of precision used for balanced evaluation.
- **ROC-AUC:** Reflects the trade-off between true positive rate and false positive rate.

### 2. Model Comparison

All classifiers were train on 80% of the data, then tested on the remaining 20%. Cross-validation was utilized to reduce bias and improve generalization.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	91.2%	90.4%	89.1%	89.7%
Decision Tree	93.0%	92.1%	91.5%	91.8%
Random Forest	95.6%	95.1%	94.7%	94.9%
XGBoost	97.1%	96.8%	96.5%	96.6%
CatBoost	96.4%	95.7%	95.2%	95.4%
Gradient Boosting	96.8%	96.3%	95.9%	96.1%

### 3. Discussion

From the results, it is evident that ensemble-based techniques including XGBoost, Gradient Boosting, and CatBoost outperform traditional classifiers with respect to overall detection performance. These models demonstrate precision and recall, making them suitable for real-time phishing prevention where minimizing false alarms is critical.

While simpler models tend to miss complex patterns found in deceptive phishing pages. On the other hand, boosting models capture these non-linear relationships effectively, leading to fewer misclassifications.

Latency testing showed that XGBoost provided near-instant predictions even under load, making it ideal for integration into real-time systems like browser extensions or network gateways.

## VII. CONCLUSION

This research presents an effective and scalable machine learning-based using a combination of URL, domain, and content-based features. Through the evaluation of multiple classification models, it was observed that ensemble methods—particularly XGBoost, Gradient Boosting, and CatBoost—achieve superior performance in terms of accuracy, precision, and recall, making them suitable for real-time deployment in cybersecurity environments.

The system successfully addresses several limitations of traditional phishing detection methods by offering adaptability to new attack patterns, faster response times, and a lower rate of false positives. Additionally, the modular design of the application supports easy integration with web browsers, APIs, and security platforms, enabling broader protection across user devices and networks.

By leveraging supervised learning techniques and a well-curated feature set, the project successfully fulfills its goal of identifying phishing sites but also establishes a strong foundation for future enhancements. With further development, such as the adoption of deep learning models or threat intelligence feeds, the system can evolve into a more advanced and intelligent web security solution.

## VIII. ACKNOWLEDGMENT

The authors intend to express their heartfelt appreciation to Prof. Aravind Naik, Assistant Professor, Department of CSE, Srinivas Institute of Technology, for his invaluable guidance, continuous support, and constructive feedback throughout the course of this project. His mentorship contributed significantly to shaping the direction and effective fulfillment of this research.

We would also like to express our heartfelt thanks to Dr. Suresha D, Head of the Department, and Prof. Shreya Shetty Assistant Professor, for their encouragement and for providing a conducive academic environment that fostered innovation and learning.

Finally, we acknowledge our institution and peers for the resources, collaboration, and motivation that contributed significantly to the development of this system.

## REFERENCES

- [1] Jain, A., & Gupta, B. (2018). Phishing detection using machine learning techniques. *Proceedings of the 2018 International Conference on System Modeling & Advancement in Research Trends (SMART)*, IEEE, 138–143.
- [2] Verma, R., & Das, A. (2020). Comparative study of various machine learning algorithms for phishing detection. *International Journal of Information Security Science*, 9(1), 20–30.
- [3] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning-based phishing detection from URLs. *Expert Systems with Applications*, 117, 345–357.
- [4] Zhou, Y., Cheng, L., & Lu, Y. (2021). A hybrid model for phishing detection using NLP and attention mechanisms. *Journal of Cybersecurity and Information Integrity*, 4(2), 55–67.
- [5] Alam, M., & Patwary, M. M. A. (2022). Phishing website detection using advanced feature engineering and ensemble learning. *Computers & Security*, 113, 102578.
- [6] Kumawat, D., Bansal, A., & Rawat, S. (2023). An unsupervised approach for detecting phishing URLs using autoencoders. *Procedia Computer Science*, 218, 1123–1130.
- [7] Nguyen, H. T., Vo, T. A., & Phan, K. (2023). Lightweight phishing detection for browser-based systems using optimized decision trees. *International Journal of Computer Applications*, 185(6), 34–40.