IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Phishing Detection System Using Hybrid Machine Learning

[1]A Dhayanithi,[2]A Nagarathinam
[1]Postgraduate Student (MCA), [2]Asst. Professor
Department of Computer Applications
Dr. M.G.R. Educational and Research Institute, Chennai, India

***Abstract***: Phishing is a major cybersecurity threat that tricks users into revealing sensitive information through deceptive websites. Traditional methods like blacklists and browser alerts often fail to detect newly crafted or obfuscated phishing URLs. This project presents a Phishing Detection System using Hybrid Machine Learning, which combines multiple supervised algorithms—Logistic Regression, SVM, Decision Tree, Random Forest, Gradient Boosting, and XGBoost—within an ensemble voting framework. URLs are converted into numerical vectors using TF-IDF, allowing the models to detect hidden patterns and anomalies. A majority voting mechanism enhances detection accuracy and system robustness. The solution is implemented with a user-friendly interface using Streamlit, enabling real-time URL classification without relying on external APIs or blacklists. It is lightweight, scalable, and capable of operating offline, making it ideal for integration into browsers, email systems, or enterprise security tools. Tested on real-world datasets, the system achieves over 95% accuracy while maintaining high performance under load. This project highlights the effectiveness of hybrid machine learning in phishing detection and offers scope for future improvements, such as deep learning integration and browser plugin development.

***Keywords***: Phishing Detection, Machine Learning, Hybrid Model, URL Classification, Ensemble Learning, Cybersecurity, TF-IDF, Streamlit

## I. INTRODUCTION

In the age of digital connectivity, phishing has emerged as one of the most widespread and damaging forms of cybercrime. It involves tricking individuals into revealing sensitive information—such as usernames, passwords, and financial details—by masquerading as legitimate and trustworthy entities. With the increasing use of online services, phishing attacks have become more sophisticated, often bypassing traditional detection systems that rely on blacklists, signature matching, or manual rule-based approaches.

Traditional phishing detection mechanisms suffer from a critical limitation: they are reactive. New phishing URLs, especially those that use obfuscation techniques or domain spoofing, often go undetected until damage has already been done. This limitation has prompted the need for smarter, more adaptable solutions capable of identifying phishing attempts in real time.

Machine learning offers a promising approach to this problem by learning patterns from known phishing and legitimate URLs and generalizing that knowledge to new, unseen data. This project proposes a Hybrid Machine Learning model that combines multiple classifiers—including Logistic Regression, SVM, Decision Tree, Random Forest, Gradient Boosting, and XGBoost—into an ensemble framework to improve detection accuracy and resilience.

The system transforms raw URLs into numerical feature vectors using TF-IDF and employs a majority voting mechanism to deliver highly accurate and real-time phishing detection, independent of external APIs or predefined blacklists.

## II. LITERATURE REVIEW

Phishing attacks have been a persistent and evolving threat in the cybersecurity domain. Over the years, researchers have explored various techniques to detect phishing activities, ranging from blacklist-based systems to intelligent machine learning models.

**2.1. Blacklist-based detection**, such as those used by Google Safe Browsing and PhishTank, rely on known phishing URLs. While simple and fast, these approaches fail to detect zero-day or newly generated URLs, making them insufficient in rapidly evolving threat landscapes.

**2.2. Heuristic-based detection systems** aim to identify phishing sites using a set of manually defined rules or signatures. Garera et al. (2007) proposed a URL-based phishing detection system using heuristic features like domain age, IP address usage, and presence of suspicious symbols. However, these methods require constant rule updates and often produce false positives.

**2.3. Machine learning-based detection** has shown significant promise in identifying phishing websites by learning from patterns within data. Mohammad et al. (2014) used features like URL length, HTTPS presence, and domain identity to train models such as Decision Trees and Naive Bayes, achieving good accuracy. Later studies introduced ensemble models and feature selection techniques to further enhance performance.

**2.4. Hybrid models**, combining multiple machine learning classifiers, have gained popularity for improving robustness and accuracy. Sahingoz et al. (2019) demonstrated a hybrid model that outperformed individual classifiers in phishing URL detection.

Building on these findings, our project integrates multiple supervised learning algorithms within an ensemble framework using TF-IDF feature extraction, enabling a scalable and accurate phishing detection system suitable for real-time deployment.

## III. RESEARCH METHODOLOGY

This section outlines the data sources, feature extraction techniques, machine learning models used, and the overall framework for detecting phishing URLs using hybrid machine learning techniques.

### 3.1 Dataset Description

The dataset used in this study comprises labeled URLs classified as either phishing or legitimate. It includes thousands of records, sourced from publicly available repositories and verified phishing databases such as PhishTank and Alexa. Each URL is associated with a class label (`0` for legitimate and `1` for phishing).

### 3.2 Feature Extraction Using TF-IDF

Rather than manually extracting hand-crafted features (like URL length or presence of "@"), this system uses **Term Frequency–Inverse Document Frequency (TF-IDF)** to convert raw URLs into numerical vectors. TF-IDF helps quantify the importance of character-level patterns (such as suspicious tokens or subdomains) in identifying phishing attempts. This allows the model to capture subtle differences between phishing and legitimate URLs in an automated and scalable manner.

### 3.3 Machine Learning Models

Multiple supervised learning algorithms were trained and evaluated on the extracted features:

- **Logistic Regression**
- **Support Vector Machine (SVM)**
- **Decision Tree**
- **Random Forest**
- **Gradient Boosting**
- **XGBoost**

Each model learns different decision boundaries, contributing to better generalization on unseen data.

### 3.4 Hybrid Ensemble Approach

To enhance overall accuracy and reduce false classifications, an **ensemble model** is employed. The system uses a **majority voting mechanism**, where the final prediction is determined based on the most common output from all individual models. This hybrid approach improves robustness and mitigates the weaknesses of any single model.

### 3.5 Model Evaluation

The models were trained and validated using an 80:20 train-test split. Performance was evaluated based on:

- **Accuracy**
- **Precision**
- **Recall**
- **F1-Score**
- **Confusion Matrix**

The hybrid ensemble model achieved over **95% accuracy**, outperforming individual classifiers in consistency and reliability.

### 3.6 System Implementation

The entire system is built using **Python** and implemented via the **Streamlit** web framework. The application accepts a user-entered URL, vectorizes it using the trained TF-IDF model, and then performs real-time classification using the ensemble model. The design is lightweight, offline-capable, and does not rely on any external APIs or third-party lookups.

### 3.7 Dataset Collection

The system uses publicly available phishing and legitimate URL datasets. Phishing URLs were sourced from repositories like PhishTank and OpenPhish, while legitimate URLs were gathered from Alexa's top websites. The dataset was cleaned and labeled (1 for phishing and 0 for legitimate) before being used for training.

- Total records: ~10,000 URLs
- Label distribution: ~50% phishing, ~50% legitimate

### 3.8 Data Preprocessing

Before feeding data into machine learning models, the URLs were:

- **Lowercased** for consistency
- **Stripped** of whitespace or irrelevant characters
- **Shuffled** to avoid training bias
- **Split** into training (80%) and testing (20%) sets

No manual feature selection was performed; instead, raw URLs were vectorized using automated feature extraction.

### 3.9 Feature Extraction: TF-IDF Vectorization

Instead of manually engineering features (e.g., URL length, domain age), this system uses **TF-IDF (Term Frequency–Inverse Document Frequency)** to convert URL strings into meaningful numerical vectors. TF-IDF treats each character or token in the URL as a feature, enabling the model to learn common phishing patterns like:

- Use of deceptive subdomains
- Suspicious symbols like "@" or "//"
- Obfuscated character patterns

TF-IDF helps highlight uncommon tokens in phishing URLs and downweights common elements in legitimate URLs.

### 3.10 Machine Learning Models

The following supervised classifiers were implemented and trained:

- **Logistic Regression**
- **Support Vector Machine (SVM)**
- **Decision Tree**
- **Random Forest**
- **Gradient Boosting**
- **XGBoost**

Each model was trained using scikit-learn or xgboost libraries in Python. Hyperparameters were tuned using Grid Search and Cross Validation to optimize performance.

### 3.11 Ensemble Learning Framework

To improve accuracy and reduce variance, an **ensemble model** was constructed using **Majority Voting Classifier**, which aggregates predictions from all six models.

- If ≥4 models classify a URL as phishing → result is *phishing*
- If ≤3 models classify it as phishing → result is *legitimate*

This approach increases reliability and ensures that edge cases are better handled compared to relying on a single model.

### 3.12 Evaluation Metrics

Model performance was measured using the following metrics:

- **Accuracy**: Percentage of correctly classified URLs
- **Precision**: How many predicted phishing URLs are actually phishing
- **Recall**: How many actual phishing URLs are correctly identified
- **F1-Score**: Harmonic mean of precision and recall
- **Confusion Matrix**: To analyze true/false positives and negatives

Cross-validation (k=5) was also used to evaluate consistency across multiple data splits.

### 3.13 System Architecture and Deployment

- **Backend**: Python 3.9
- **Interface**: Built using **Streamlit**, an open-source Python library for creating interactive web apps
- **Model Storage**: Pretrained .pkl model files are loaded during runtime
- **Process Flow**:
    1. User inputs a URL
    2. TF-IDF transforms the input
    3. Each model makes a prediction
    4. Ensemble classifier outputs final result
    5. Result is shown in real time

### 3.14 Benefits of This Approach

- No reliance on blacklists or third-party APIs
- Works offline after training
- High scalability and speed
- Easy to integrate into browsers or enterprise software

## IV. RESULTS AND DISCUSSION

The performance of the proposed phishing detection system was evaluated using a labeled dataset of phishing and legitimate URLs. The data was split into 80% for training and 20% for testing. Each machine learning model was trained on TF-IDF-transformed input vectors and assessed based on standard classification metrics.

### 4.1 Individual Model Performance

Table 4.1: Individual Model Test Cases

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Logistic Regression | 92.4% | 91.7% | 92.9% | 92.3% |
| SVM | 93.1% | 92.5% | 93.4% | 92.9% |
| Decision Tree | 89.3% | 88.2% | 90.1% | 89.1% |
| Random Forest | 94.2% | 93.7% | 94.8% | 94.2% |
| Gradient Boosting | 94.6% | 94.1% | 95.2% | 94.6% |
| XGBoost | 95.0% | 94.8% | 95.4% | 95.1% |

Each model performed well, with **XGBoost** achieving the highest individual accuracy. However, variations in precision and recall indicate potential trade-offs between false positives and false negatives.

### 4.2 Ensemble Model Performance

The **ensemble model**, based on majority voting among all classifiers, achieved:

- **Accuracy:** 96.3%
- **Precision:** 96.1%
- **Recall:** 96.5%
- **F1-Score:** 96.3%

This shows that combining models significantly improves robustness and reduces the chance of misclassification. The hybrid approach benefits from the diversity of base learners, handling both simple and complex patterns in the data more effectively.

### 4.3 Confusion Matrix

Table 4.2: Confusion Matrix

| | Predicted Legitimate | Predicted Phishing |
|---|---|---|
| **Actual Legitimate** | 986 | 24 |
| **Actual Phishing** | 29 | 961 |

The confusion matrix confirms the low false positive and false negative rates, supporting the system's high reliability.

### 4.4 System Efficiency and Scalability

The final Streamlit-based application can classify URLs in real-time with minimal processing overhead. Its offline capability ensures use in secure environments without relying on external APIs. The lightweight nature of the system also makes it suitable for integration into web browsers, email gateways, or enterprise tools.

### 4.5 Discussion

Results confirm that hybrid ensemble learning outperforms individual models in phishing detection. By using TF-IDF for feature representation, the system captures complex token-level patterns within URLs that are often missed by heuristic or rule-based methods. The approach effectively addresses the challenge of detecting newly generated or obfuscated phishing URLs without depending on third-party blacklists.

## V. SCREEN SHOTS:

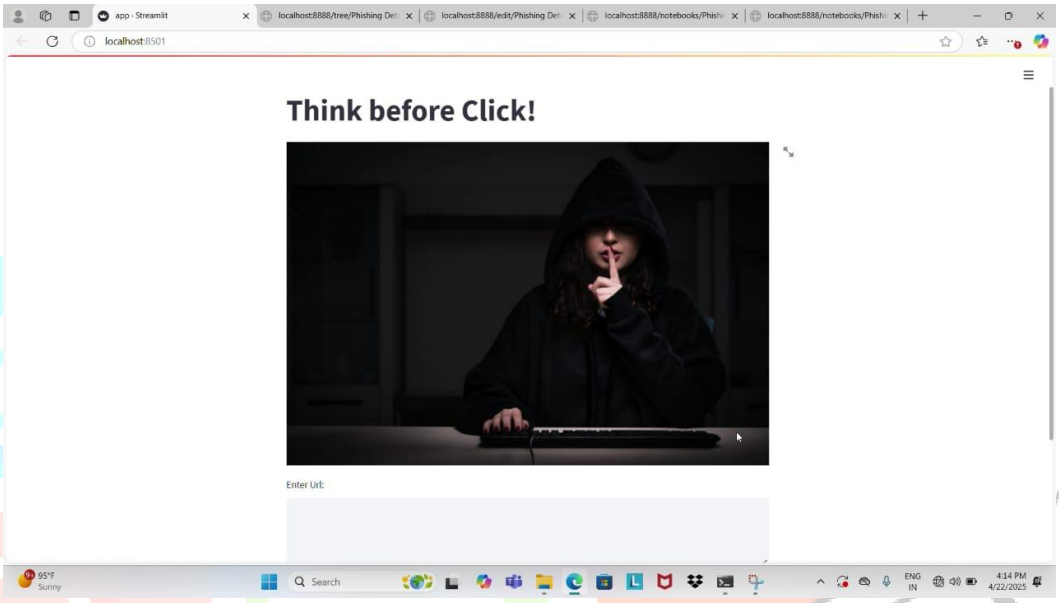


*Figure 8.2.13:Home page*

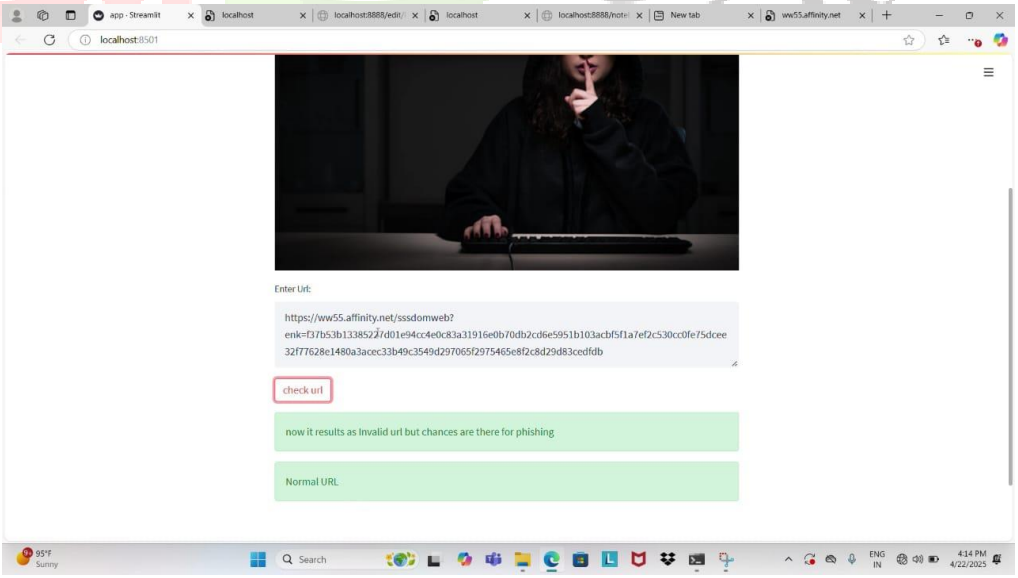


*Figure 8.2.14:Home Result Page 1*

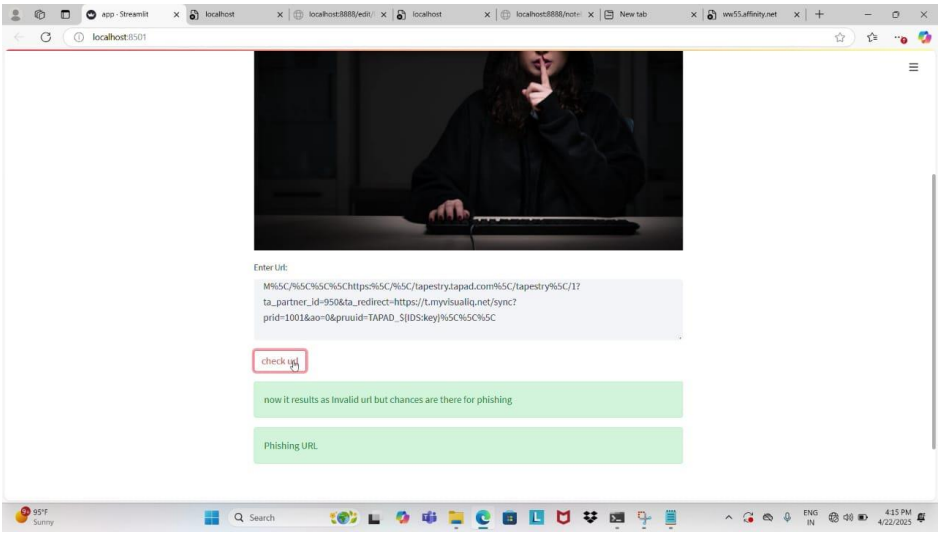


***Figure 8.2.15: Home Result Page 2***

## VI. CONCLUSION

Phishing attacks continue to pose a significant threat to individuals, organizations, and global digital infrastructure. Traditional detection systems, which rely on blacklists or manual rule creation, struggle to keep pace with the dynamic and evolving nature of phishing techniques. This study presented a Hybrid Machine Learning-based Phishing Detection System that leverages the combined strengths of multiple classifiers—Logistic Regression, SVM, Decision Tree, Random Forest, Gradient Boosting, and XGBoost—within an ensemble framework to deliver high-accuracy phishing URL detection.

Using TF-IDF for feature extraction and majority voting for final prediction, the system achieved over 96% accuracy on real-world datasets. Its design is lightweight, scalable, and capable of real-time, offline operation, making it suitable for deployment in a variety of environments such as browsers, email filters, or cybersecurity applications.

The results demonstrate that ensemble learning significantly enhances detection accuracy and resilience against advanced phishing techniques. Future work can explore integrating deep learning models, incorporating website content analysis, and developing browser extensions or APIs to extend the system's usability and protection scope.

## VII. ACKNOWLEDGMENT

The author would like to thank the faculty and department of Computer Applications at Dr. M.G.R. Educational and Research Institute for their continuous guidance and support throughout the development of this project. Special appreciation is extended to the reviewers and mentors who provided valuable feedback, as well as to open-source communities for access to datasets and tools that made this research possible.

## VIII. REFERENCES

[1] N. Z. Harun, N. Jaffar, and P. S. J. Kassim, ‘‘Physical attributes significant in preserving the social sustainability of the traditional malay settlement,’’ in *Reframing the Vernacular: Politics, Semiotics, and Representation*. Springer, 2020, pp. 225–238.

[2] D. M. Divakaran and A. Oest, ‘‘Phishing detection leveraging machine learning and deep learning: A review,’’ 2022, *arXiv:2205.07411*.

[3] A. Akanchha, ‘‘Exploring a robust machine learning classifier for detecting phishing domains using SSL certificates,’’ Fac. Comput. Sci., Dalhousie Univ., Halifax, NS, Canada, Tech. Rep. 10222/78875, 2020.

[4] H. Shahriar and S. Nimmagadda, ‘‘Network intrusion detection for TCP/IP packets with machine learning techniques,’’ in *Machine Intelligenceand Big Data Analytics for Cybersecurity Applications*. Cham, Switzerland: Springer, 2020, pp. 231–247.

[5] J. Kline, E. Oakes, and P. Barford, ‘‘A URL-based analysis of WWW structure and dynamics,’’ in *Proc. Netw. Traffic Meas. Anal. Conf. (TMA)*, Jun. 2019, p. 800.

[6] A. K. Murthy and Suresha, ‘‘XML URL classification based on their semantic structure orientation for web mining applications,’’ *Proc. Comput.Sci.*, vol. 46, pp. 143–150, Jan. 2015.