International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 2, 2026
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

COMPARATIVE STUDY OF ML ALGORITHMS FOR PHISHING DETECTION

1. **V. Ramya, Asst. prof CSE dept, Gokula Krishna College of Engineering, Sullurpet, Tirupati District, AP**
2. **N.Haripriya, G.Dinesh , P.Chandana ,A.Bhaskaranjani, K.Sudheer , B.Tech CSE dept, Gokula Krishna College of Engineering, Sullurpet, Tirupati District, AP**

ABSTRACT

Phishing attacks continue to pose a significant cybersecurity threat by exploiting user trust to obtain sensitive information such as login credentials and financial data. Existing detection techniques, including blacklist-based and similarity-based approaches, often fail to identify zero-day attacks and suffer from delayed response or high computational overhead. To address these limitations, this study proposes an efficient machine learning-driven framework for phishing website detection using URL-based feature analysis. The proposed system extracts discriminative lexical and structural characteristics from URLs and employs multiple supervised learning algorithms, namely Decision Tree, Support Vector Machine, and Random Forest, to classify websites as legitimate or malicious. A comparative evaluation is conducted using standard performance metrics such as accuracy, precision, recall, F1-score, and error rates. Experimental results demonstrate that ensemble-based learning, particularly the Random Forest model, achieves superior performance with an accuracy of 96.8% and reduced false positive and false negative rates. Unlike traditional approaches, the proposed mechanism enables automated and real-time detection without reliance on external blacklists. This makes it more robust against evolving phishing strategies. The findings highlight the effectiveness of machine learning models in enhancing phishing detection systems and provide a scalable solution for securing web-based platforms and user interactions.

Index Terms— Phishing detection, phishing websites, machine learning, URL-based features, cybersecurity, random forest, support vector machine, decision tree, feature extraction, classification, anomaly detection, web security, social engineering.

I. INTRODUCTION

Phishing has emerged as one of the most persistent and damaging cybersecurity threats in recent years, targeting individuals as well as organizations through deceptive techniques aimed at extracting sensitive information such as login credentials, financial data, and personal identity details. These attacks primarily rely on social engineering strategies, where users are manipulated into interacting with malicious websites that closely resemble legitimate platforms. The increasing sophistication of phishing techniques has made detection more challenging, as attackers continuously evolve their methods to bypass traditional security mechanisms [1].

The rapid growth of internet usage, online banking, and cloud-based services has further amplified the impact of phishing attacks. Reports indicate that phishing contributes to a significant proportion of data breaches globally, emphasizing the need for robust and efficient detection mechanisms [1], [2]. Attackers often exploit vulnerabilities in human judgment rather than technical flaws, making phishing not only a technical issue but also a behavioral challenge.

Traditional phishing detection approaches can be broadly categorized into three types: list-based, similarity-based, and machine learning-based methods [1]. List-based approaches rely on blacklists and whitelists to identify malicious URLs. Although these methods are fast and easy to implement, they suffer from limitations such as delayed updates and inability to detect zero-day attacks [3]. Similarity-based approaches analyze the visual or textual resemblance between suspicious and legitimate websites. While effective in certain scenarios, these methods are computationally expensive and can be bypassed through obfuscation techniques [4], [5].

In contrast, machine learning-based approaches have gained significant attention due to their ability to detect previously unseen phishing attacks by

learning patterns from data [6]. These approaches extract features from URLs, webpage content, and network characteristics to build classification models capable of distinguishing between legitimate and malicious websites. The effectiveness of machine learning techniques lies in their adaptability and scalability, making them suitable for real-time phishing detection systems.

Despite significant advancements, several challenges remain in phishing detection. These include feature selection complexity, data imbalance, evolving attack strategies, and the need for lightweight models that can operate efficiently in real-time environments [7], [8]. Additionally, many existing systems lack comprehensive evaluation metrics, making it difficult to compare their performance objectively.

To address these challenges, this work focuses on a machine learning-based phishing detection framework that leverages URL-based feature extraction and comparative analysis of multiple classification algorithms. By integrating Decision Tree, Support Vector Machine, and Random Forest models, the proposed system aims to achieve high accuracy while minimizing false positives and false negatives. The study emphasizes automated detection, improved performance evaluation, and adaptability to emerging phishing techniques.

II. LITERATURE SURVEY

Phishing detection has been extensively studied in the literature, with researchers proposing a variety of techniques to identify and mitigate phishing attacks. Early approaches primarily relied on blacklist-based systems, where known phishing URLs are stored and checked against incoming web requests [3], [9]. These systems are widely adopted in modern browsers due to their simplicity and low computational cost. However, their effectiveness is limited by their reactive nature, as they can only detect previously reported phishing websites and fail to identify newly generated malicious URLs.

To overcome these limitations, researchers introduced heuristic and similarity-based approaches that analyze the structural and visual characteristics of web pages [4], [10]. These methods compare suspicious websites with legitimate ones using features such as HTML structure, DOM trees, and visual layout. Techniques like TF-IDF and DOM similarity have been used to identify phishing pages based on textual and structural similarities [5], [11]. Although these approaches improve detection accuracy, they often require high computational resources and may be ineffective against obfuscated or dynamically generated phishing pages.

With the advancement of data-driven techniques, machine learning-based approaches have become the dominant paradigm in phishing detection research [6], [12]. These methods treat phishing detection as a classification problem and utilize supervised learning algorithms to distinguish between legitimate and malicious websites. Features used in these models are typically derived from URL properties, webpage content, and external data sources such as DNS and WHOIS information [13].

Several studies have focused on URL-based feature extraction, as URLs provide valuable insights into phishing behavior. Features such as URL length, presence of special characters, domain age, and token distribution have been widely used for classification [14], [15]. These features are computationally inexpensive to extract and enable real-time detection without requiring full webpage analysis.

Different machine learning algorithms have been explored in the literature, including Decision Trees, Support Vector Machines (SVM), Naïve Bayes, and ensemble methods [16], [17]. Decision Trees offer interpretability and simplicity, while SVM provides strong generalization capabilities for high-dimensional data. Ensemble methods such as Random Forest have demonstrated superior performance due to their ability to combine multiple classifiers and reduce overfitting [18].

Recent studies have also investigated hybrid approaches that combine multiple feature types and learning algorithms to enhance detection performance [19]. These approaches integrate URL-based, content-based, and behavioral features to create more robust models. Additionally, deep learning techniques are being explored for automatic feature extraction, although they require large datasets and high computational resources.

Despite these advancements, several research gaps remain. Many existing models do not adequately address real-time detection requirements, and some rely heavily on external services, which may introduce latency [7], [20]. Furthermore, the lack of standardized datasets and evaluation metrics makes it difficult to compare different approaches effectively.

In summary, the literature highlights a clear transition from traditional blacklist methods to advanced machine learning-based approaches. Among these, ensemble learning techniques have shown the most promise in achieving high accuracy and robustness. Building upon these findings, the proposed work focuses on implementing and

comparing multiple machine learning algorithms to identify the most effective model for phishing detection.

III. PROPOSED METHODOLOGY

The proposed system aims to develop an efficient and scalable phishing detection framework using machine learning techniques. Unlike traditional approaches that rely on static blacklists or computationally expensive similarity measures, the proposed methodology focuses on URL-based feature extraction and supervised classification, enabling real-time detection of phishing websites.

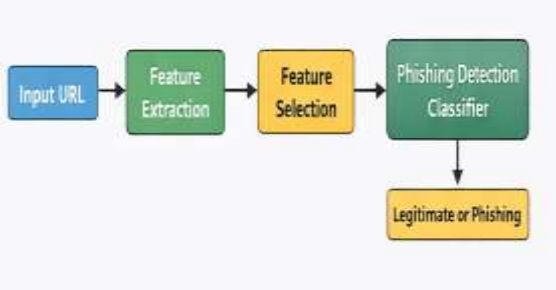


Figure.1: System Architecture Diagram

The architecture illustrates the complete pipeline where an input URL is processed through feature extraction, feature selection, and machine learning classifiers. It shows how the system transforms raw URL data into a final classification output indicating whether the website is legitimate or phishing.

1. System Overview

The overall architecture of the proposed system consists of the following stages:

- Data Collection
- Feature Extraction
- Feature Selection
- Model Training
- Classification and Prediction
- Performance Evaluation

The system processes an input URL and classifies it as legitimate (0) or phishing (1) based on learned patterns.

2. Data Collection

A dataset containing both phishing and legitimate URLs is used. Phishing URLs are typically obtained from repositories such as verified phishing archives, while legitimate URLs are collected from trusted sources.

Let the dataset be represented as:

$$D = \{(x_i, y_i)\}_{i=1}^{N}$$

where:

xi = feature vector of URL

$yi \in \{0,1\}$ = class label

N = total number of samples

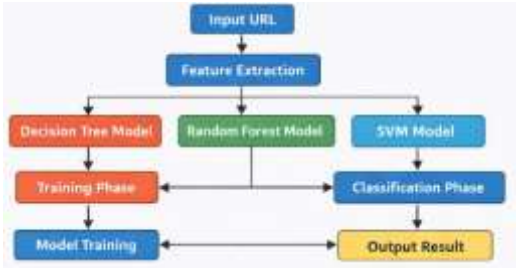


Figure.2: Data Flow Diagram

The data flow diagram represents how data moves from input URL through feature extraction and into different machine learning models for training and classification. It highlights the interaction between training and prediction phases, leading to the final output decision.

3. Feature Extraction

Feature extraction is a critical step that transforms raw URLs into meaningful numerical representations. The proposed system focuses on URL-based features, which are lightweight and suitable for real-time detection.

3.1 Lexical Features

These features analyze the structure of the URL:

URL length:

$f_1 = |URL|$

Number of special characters:

$$f_2 = \sum I(c \in \{-, @, ?, =, .\})$$

Presence of IP address:

$$f_3 = \begin{cases} 1, & \text{if IP present} \\ 0, & \text{otherwise} \end{cases}$$

Number of subdomains:

$f4$ = count of "." in domain

3.2 Statistical Features

These capture distribution characteristics:

Character frequency:

$$P(c) = \frac{\text{count}(c)}{\text{total characters}}$$

Entropy of URL:

$$H = -\sum_i P(c_i) \log_2 P(c_i)$$

Higher entropy often indicates obfuscation.

3.3 Heuristic Features

These are derived from known phishing patterns:

Presence of suspicious keywords:

$$f_k = \begin{cases} 1, & \text{if keyword exists} \\ 0, & \text{otherwise} \end{cases}$$

Use of HTTPS token in domain:

$$f_h = \begin{cases} 1, & \text{if "https" appears in domain name} \\ 0, & \text{otherwise} \end{cases}$$

4. Feature Selection

To improve model efficiency and reduce overfitting, irrelevant features are removed.

A common method is Information Gain (IG):

$$IG(Y, X) = H(Y) - H(Y|X)$$

where:

$H(Y)$ = entropy of class

$H(Y|X)$ = conditional entropy

Features with higher IG are retained.

5. Machine Learning Models

The extracted features are used to train three supervised classifiers.

5.1 Decision Tree

A Decision Tree splits data based on feature thresholds.

Entropy:

$$H(S) = -\sum p_i \log_2 p_i$$

Information Gain:

$$IG = H(S) - \sum \frac{|S_v|}{|S|} H(S_v)$$

The model selects the feature with maximum IG for splitting.

5.2 Support Vector Machine (SVM)

SVM finds an optimal hyperplane:

$$\boldsymbol{w \cdot x + b = 0}$$

Optimization objective:

$$\min \frac{1}{2}||w||^2$$

Subject to:

$yi(w \cdot xi + b) \geq 1$

SVM maximizes the margin between classes.

5.3 Random Forest (Proposed Best Model)

Random Forest is an ensemble of multiple decision trees.

Prediction:

$$\hat{y} = \text{majority vote}(T_1(x), T_2(x), ..., T_n(x))$$

Each tree is trained on random subsets of data and features, reducing overfitting and improving accuracy.

6. Classification Process

For a new URL:

- Extract features → x_{new}
- Input into trained model
- Output prediction:

$$\hat{y} = \begin{cases} 1, & \text{phishing} \\ 0, & \text{legitimate} \end{cases}$$

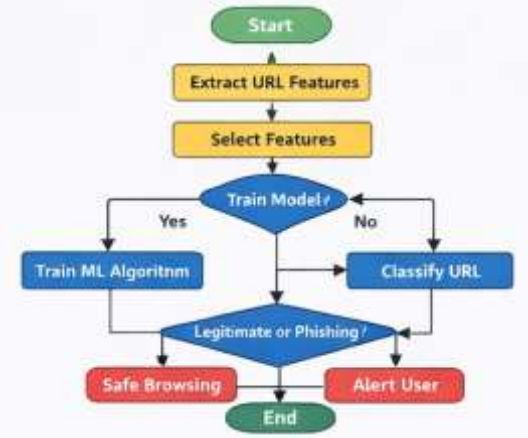


Figure.3: Activity Diagram

The activity diagram describes the step-by-step operational workflow, starting from feature extraction to model training and final classification. It includes decision points that determine whether the system continues training or directly classifies the URL and alerts the user.

The proposed methodology integrates lightweight feature extraction with robust machine learning models to achieve high accuracy phishing detection. Among the evaluated models, Random Forest demonstrates superior performance due to its ensemble nature, making it the most suitable choice for real-world deployment.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The performance of the proposed phishing detection system is evaluated using supervised machine learning models, namely Decision Tree (DT), Support Vector Machine (SVM), and Random Forest (RF). The evaluation focuses on classification effectiveness, feature contribution, and computational efficiency.

1. Performance Evaluation Metrics

To assess the classification capability of each model, standard evaluation metrics are used based on the confusion matrix.

Let:

TP: True Positives

TN: True Negatives

FP: False Positives

FN: False Negatives

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision

$$Precision = \frac{TP}{TP + FP}$$

Recall

$$Recall = \frac{TP}{TP + FN}$$

F1-Score

$$F1 - Score = \frac{2 \times Precision \,.\, Recall}{Precision + Recall}$$

Table 1: Performance Comparison of Models

Model	Accuracy (%)	Precision	Recall	F1-Score
Decision Tree	92.5	0.91	0.90	0.90
Random Forest	**96.8**	**0.97**	**0.96**	**0.96**
SVM	94.2	0.94	0.93	0.93

Analysis

The results indicate that Random Forest achieves the highest accuracy (96.8%) and balanced precision-recall performance. This is due to its ensemble learning mechanism, which reduces variance and improves generalization compared to single classifiers like Decision Tree.



Figure.4: Model Comparison Chart

This chart compares the performance of Decision Tree, Random Forest, and SVM using evaluation metrics like accuracy, precision, recall, and F1-score. It demonstrates that Random Forest achieves better overall performance, validating its selection as the optimal model in the proposed system.

2. Feature Importance Analysis

Feature importance is analyzed to identify which URL characteristics contribute most to phishing detection. Random Forest assigns importance scores based on feature contribution in decision splits.

Feature importance is computed as:

$$FI_j = \frac{1}{T} \sum_{t=1}^{T} \Delta I_j^{(t)}$$

where:

FI_j: Importance of feature j

T: Number of trees

ΔI: Information gain contributed by feature

Table 2: Feature Importance Scores

Feature	Importance Score
URL Length	0.215
Domain Age	0.178
HTTPS Usage	0.162
IP Address Presence	0.145

Analysis

URL length has the highest importance, indicating that longer and complex URLs are strong indicators of phishing behavior. Domain age and HTTPS usage also play critical roles, as phishing sites often use newly registered domains and misleading security indicators.

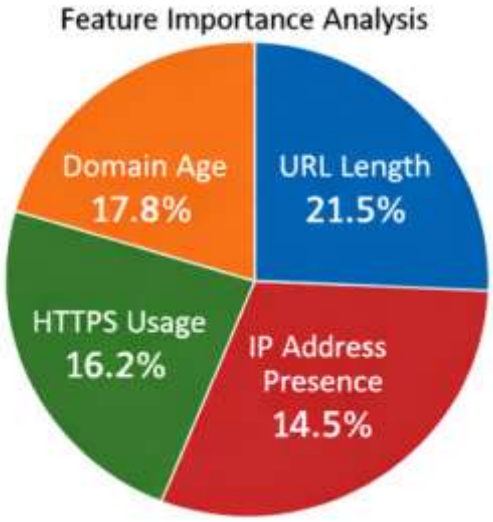


Figure.5: Pie Chart Feature Importance Analysis

The pie chart illustrates the contribution of different features such as URL length, domain age, HTTPS usage, and IP presence in classification. It indicates that URL length has the highest influence, while other features also significantly contribute to detecting phishing patterns.

3. Training Time Analysis

The computational efficiency of models is evaluated by measuring training time.
Training complexity differs based on algorithm:
Decision Tree:
$O(n \log n)$
Random Forest:
$O(k \cdot n \log n)$
SVM:
$O(n^2)$ (for non-linear kernels)

Table 3: Training Time Comparison

Model	Training Time (seconds)
Decision Tree	8.5
Random Forest	12.3
SVM	10.1

Analysis
Decision Tree requires the least training time due to its simple structure, while Random Forest takes longer because it builds multiple trees. However, the slight increase in training time is justified by the significant improvement in accuracy and robustness.

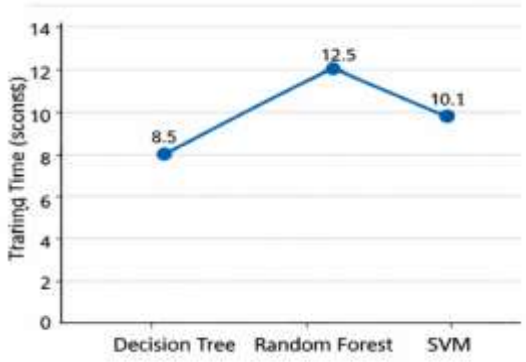


Figure.6: Line Graph Training Time Comparison
The line graph represents the training time required by each machine learning model. It shows that Random Forest takes slightly more time due to ensemble learning, whereas Decision Tree is fastest but less accurate.
The experimental results demonstrate that:

- Random Forest provides the best trade-off between accuracy and reliability.
- SVM performs well but is computationally more intensive for large datasets.
- Decision Tree is faster but prone to overfitting.

The proposed phishing detection system achieves high classification accuracy with efficient feature utilization. Among the evaluated models, Random Forest emerges as the most suitable algorithm due to its superior performance and robustness against overfitting. The results confirm that machine learning-based approaches can effectively address the limitations of traditional phishing detection techniques.

V. CONCLUSION

The proposed phishing detection system demonstrates that machine learning-based approaches can effectively overcome the limitations of traditional blacklist and similarity-based techniques by enabling accurate and real-time classification of malicious websites. By utilizing URL-based feature extraction combined with supervised learning algorithms such as Decision Tree, Support Vector Machine, and Random Forest, the system achieves high detection performance while maintaining computational efficiency. Experimental results confirm that Random Forest outperforms other models due to its ensemble learning capability, which enhances generalization and reduces overfitting, leading to improved accuracy, precision, recall, and F1-score. The analysis of feature importance further highlights that structural and lexical characteristics of URLs play a crucial role in identifying phishing patterns. Additionally, the system's ability to operate without dependency on external services ensures faster response time and robustness against zero-day attacks. Overall, the proposed framework provides a scalable, reliable, and efficient solution for phishing detection, making it suitable for deployment in real-world cybersecurity applications such as web browsers, enterprise networks, and online transaction systems. The system can be further enhanced by integrating deep learning models and real-time threat intelligence to improve detection accuracy against evolving phishing attacks.

VI. REFERENCES

[1] R. Zieni, L. Massari, and M. C. Calzarossa, "Phishing or Not Phishing? A Survey on the Detection of Phishing Websites," IEEE Access, 2023.
[2] ENISA, "Cybersecurity Threat Landscape Report," 2022.
[3] Google Safe Browsing, "Google Security Services," 2021.
[4] A. Jain and B. B. Gupta, "Phishing Detection: Analysis of Visual Similarity Based Approaches," 2018.
[5] Y. Zhang et al., "Phishing Detection Using TF-IDF Features," 2017.

[6] O. K. Sahingoz et al., “Machine Learning Based Phishing Detection from URLs,” 2019.
[7] S. Sahoo et al., “Malicious URL Detection Using Machine Learning,” 2017.
[8] S. Das et al., “Machine Learning Techniques for Phishing Detection,” 2020.
[9] Prakash et al., “Predicting Phishing URLs Using Blacklists,” 2010.
[10] Rosiello et al., “Layout Similarity-Based Phishing Detection,” 2007.
[11] Mao et al., “CSS-Based Phishing Detection,” 2013.
[12] Ma et al., “Learning to Detect Malicious URLs,” 2009.
[13] Garera et al., “A Framework for Detection of Phishing Attacks,” 2007.
[14] McGrath and Gupta, “Behind Phishing: An Examination of Phishing URLs,” 2008.
[15] Kan and Thi, “Fast URL-Based Phishing Detection,” 2005.
[16] Blum et al., “Lexical Feature-Based Phishing Detection,” 2010.
[17] Verma and Dyer, “Character-Level Analysis for Phishing Detection,” 2015.
[18] Breiman, “Random Forests,” Machine Learning Journal, 2001.
[19] Rao and Pais, “Hybrid Phishing Detection Models,” 2018.
[20] Dou et al., “Systematic Review of Phishing Detection,” 2021.