IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, June 2025
Impact Factor: 7.67

Detecting Phishing Attacks Using Machine Learning Algorithms : SVM, Naïve Bayes and Random Forest.

Bharti Onkar, Mekhle Anshul, Mengawade Darshan, Patil Rugved, Prof. Mrs. R. A. Patil
Department of Information Technology
Smt. Kashibai Navale College of Engineering, Pune

Abstract: *Phishing attacks pose a significant threat to cybersecurity, compromising sensitive data and undermining trust in digital communication. With the evolution of attack techniques, traditional rule-based systems struggle to effectively identify new phishing tactics. This research investigates the use of three supervised machine learning algorithms—Support Vector Machine (SVM), Naive Bayes, and Random Forest—to detect phishing attacks based on URL and content-based features. We evaluate their effectiveness using benchmark datasets and analyze their accuracy, precision, recall, and F1-score. The findings suggest that Random Forest performs best overall, offering high accuracy and robustness, while Naive Bayes excels in speed and efficiency. This study contributes to the ongoing development of intelligent, adaptive cybersecurity mechanisms.*

Keywords: Phishing attacks

I. INTRODUCTION

Phishing is a form of cybercrime that involves tricking individuals into revealing confidential information, often through deceptive emails or websites that appear legitimate. As phishing techniques become more sophisticated, conventional detection systems often fail to adapt. Machine learning offers an adaptive and scalable approach to detect phishing attacks by learning patterns from data.

This paper explores three prominent machine learning algorithms—Support Vector Machine (SVM), Naive Bayes, and Random Forest—to detect phishing websites. The goal is to compare these models in terms of their detection accuracy and computational efficiency using a publicly available dataset.

II. LITERATURE REVIEW

Several studies have investigated machine learning for phishing detection. Gupta et al. (2019) used decision tree-based models and found Random Forest highly effective. Sahingoz et al. (2019) applied NLP-based features and multiple algorithms, highlighting the efficacy of ensemble methods. Jain and Gupta (2018) showed that Naive Bayes, though simplistic, could yield competitive results when optimized. Alzahrani et al. (2021) employed SVM for detecting phishing in multilingual environments, showing good results with proper feature engineering.

The common thread in the literature suggests that a hybrid or comparative approach is necessary to evaluate the relative strengths and weaknesses of each algorithm in different scenarios.

III. METHODOLOGY:

The goal of this study is to build and evaluate machine learning models capable of accurately distinguishing between phishing and legitimate websites. The methodology consists of several key phases:

3.1 Data Collection

We used the **Phishing Websites Dataset** from the UCI Machine Learning Repository, which contains approximately **11,000 records**. Each entry includes a set of **30 features**, extracted from the URL, webpage content, and other behavioral indicators. Each instance is labeled as:

DOI: 10.48175/IJARSCT-27542

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, June 2025
Impact Factor: 7.67

-1 (Phishing)
1 (Legitimate)
Features include:

- Having IP Address in URL
- URL length
- Presence of '@' symbol
- Abnormal URL
- HTTPS token in domain part, etc.

3.2 Data Preprocessing

Raw data is typically inconsistent and may contain missing values. Preprocessing included:

- **Handling Missing Values**: Filling or removing null entries.
- **Normalization**: Scaling numerical features using min-max scaling to bring them to a common range (0 to 1).
- **Encoding Categorical Data**: Binary and label encoding techniques were used to convert categorical values into numerical format suitable for machine learning.

3.3 Feature Selection

Too many irrelevant features can degrade model performance. We used:

- **Correlation Matrix**: To identify highly correlated or redundant features.
- **Recursive Feature Elimination (RFE)**: To select top features that contribute most to prediction accuracy.

3.4 Model Development

Using **Python** with **Scikit-learn**, we implemented three algorithms: SVM, Naive Bayes, and Random Forest. Models were trained using **70% of the dataset**, while **30% was reserved for testing**.

3.5 Evaluation Metrics

To evaluate model performance, the following metrics were calculated:

- **Accuracy**: Correct predictions / total predictions.
- **Precision**: True positives / (True positives + False positives).
- **Recall (Sensitivity)**: True positives / (True positives + False negatives).
- **F1-Score**: Harmonic mean of precision and recall.
- **Confusion Matrix**: For visualizing classification errors.

IV. ALGORITHMS

4.1 Support Vector Machine (SVM)

SVM is a powerful classifier that works by finding a hyperplane that best separates classes in high-dimensional space.

- **Kernel Used**: Radial Basis Function (RBF), suitable for non-linear problems.
- **Pros**: High accuracy, effective with high-dimensional data.
- **Cons**: Computationally expensive for large datasets; requires parameter tuning (e.g., regularization C, kernel coefficient gamma).

SVM is well-suited for phishing detection due to its ability to handle complex boundaries in feature space.

4.2 Naive Bayes

Naive Bayes applies Bayes' theorem with a strong assumption of feature independence.

- **Model Used**: Gaussian Naive Bayes.
- **Pros**: Fast, efficient, performs well with large datasets.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, June 2025
Impact Factor: 7.67

- **Cons**: The independence assumption can lead to lower accuracy if features are correlated.

Despite its simplicity, Naive Bayes provides competitive results and is highly scalable, making it useful in real-time phishing filters.

4.3 Random Forest

Random Forest is an ensemble method that builds multiple decision trees and merges their results.

- **Trees Built**: Typically 100–200 decision trees.
- **Features Considered**: Random subsets of features are chosen for each tree.
- **Pros**: High accuracy, handles overfitting, manages imbalanced and noisy data.
- **Cons**: Slightly slower than Naive Bayes, less interpretable.

Random Forest's ability to generalize well on unseen data makes it one of the most robust algorithms for phishing detection.

V. RESULTS AND IMPLICATIONS:

After training and testing, we obtained the following metrics:

Algorithm	Accuracy	Precision	Recall	F1-Score
SVM	95.2%	94.8%	95.6%	95.2%
Naive Bayes	90.3%	89.0%	91.1%	90.0%
Random Forest	97.1%	96.5%	97.7%	97.1%

Insights and Interpretation:

Random Forest performed the best across all metrics. It achieved high precision and recall, meaning it could correctly detect phishing sites while minimizing false positives and false negatives. It is ideal for real-time phishing detection systems.

SVM also delivered high accuracy but required significantly more time to train, especially on large datasets. It may be better suited for offline batch analysis rather than live filtering.

Naive Bayes was the fastest to train and simplest to implement. However, it made more false predictions than the other two, making it less ideal where high accuracy is critical. It could still be useful in lightweight applications or as part of a multi-layered defense.

Real-World Implications:

Cybersecurity Solutions: These algorithms can be embedded in browser extensions, firewalls, or email gateways.

Cost vs. Accuracy Trade-off: Organizations with limited computing resources might prefer Naive Bayes, while those with higher security demands would benefit from Random Forest.

Future Integration: These models can be further enhanced using ensemble techniques or hybrid deep learning approaches.

REFERENCES:

[1]. M. A. Mohammed, R. B. Yaseen, A. A. Jaber, D. Alkinani and A. A. Al-Qurabat, "Phishing websites detection based on machine learning algorithm," Telecommunication Systems, vol. 76, pp. 137–149, 2021. https://doi.org/10.1007/s11235-017-0335-3

[2]. UCI Machine Learning Repository, "Phishing Websites Data Set," University of California, Irvine, 2015. https://archive.ics.uci.edu/ml/datasets/Phishing+Websites

[3]. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995