ISSN: 2349-6002

An Enhanced Detection Model for Phishing Websites Using ABAC, Heuristic Techniques, and Canopy Feature Optimization

Suyog Vilas Patil[1], Dr. Vijay Pal Singh[2]
[1]*Computer Science and Engineering Faculty of Engineering and Technology, Mangalayatan University, Beswan, Aligarh*
[2]*Professor, DCEA.FET Computer Science and Engineering Faculty of Engineering and Technology, Mangalayatan University, Beswan, Aligarh*

Abstract- Phishing is a broad tactic used by scammed people to reveal their personal data using false websites. The URL of a phishing website is designed to steal personal data such as usernames, passwords, and online finance activities. Phisher uses aesthetically and linguistically similar websites for these real websites. A powerful tool to disrupt phishing attacks is machine learning. Intruders often use phishing as it is easier to deceive the victim by clicking on malicious links that look real than trying to overcome computer security measures. The presented method uses machine learning to create innovative approaches to recognizing phishing websites. The suggested method for identifying phishing websites based on features of URL anomalies uses the Gradient Boost Classifier model.The study's findings demonstrate that the suggested method effectively and instantly distinguished between phony and authentic websites.

Keywords: **Phishing Attacks, Machine Learning, Gradient Boosting, Detection.**

1. INTRODUCTION

Phishing attacks, which use deceptive tactics to obtain people, expose and disclose critical information, and make malware downloads appear legal, are a major and ongoing threat to the digital world. Blacklists and other contemporary techniques for thwarting phishing attempts frequently fail to keep the phisher's identity intact. By examining a range of web characteristics, machine learning has emerged as a viable technique for identifying phishing URLs in recent years. The usefulness of different machine learning models in identifying phishing assaults is investigated in this paper. In order to train and test models and evaluate their performance in order to identify the optimal approach, our research makes use of a large amount of data from phishing websites. The creation of stronger phishing systems is greatly impacted by our findings. Phishing is still one of the most common cybersecurity dangers, and attackers are always improving their strategies to get past conventional detection systems. These attacks often entail constructing fraudulent websites that imitate legitimate platforms to steal sensitive information such as login credentials, financial details, and personal data. The financial impact of phishing assaults is enormous, with global losses estimated at billions of dollars annually, hurting both individuals and corporations.

Traditional approaches to phishing detection rely heavily on blacklists and static rule-based systems. While these methods provide some level of protection, they fail to identify zero-day phishing sites and struggle to adapt to new deception techniques. Machine learning approaches offer a more dynamic solution by learning patterns and characteristics of phishing URLs without depending solely on known attack signatures.

This research addresses the limitations of existing approaches by developing an optimized gradient boosting framework specifically designed for phishing URL detection. Our contributions include:

A comprehensive feature extraction methodology focusing on URL structure, domain information, and content indicators Implementation of an optimized

gradient boosting classifier with parameters tuned for phishing detection

Creation of an intuitive online application for the classification of phishing URLs in real time Evaluation of our method in comparison to well-known machine learning models This paper's remaining sections are arranged as follows: The literature on phishing detection is reviewed in Section 2. Our technique, including data collection, preprocessing, and model implementation, is described in depth in Section 3. The system implementation is explained in Section 4. Performance analysis and experimental findings are shown in Section 5. A comparison with other classification methods is given in Section 6, and Section 7 offers conclusions and suggestions for further research.

2. LITERATURE REVIEW

2.1 Traditional Approaches to Phishing Detection

Heuristic-based techniques and blacklists were the mainstays of early phishing detection. Heuristic approaches use rule-based systems to find suspicious features, whereas blacklists keep databases of known phishing URLs. Using TF-IDF algorithms, Zhang et al. (2007) created CANTINA, one of the first content-based methods for analyzing webpage content. While effective against known threats, these methods demonstrate limited capabilities against zero-day phishing attacks and require constant updating.

2.2 Machine Learning for Phishing Detection

Machine learning approaches have gained significant traction in phishing detection research. Abdelhamid et al. (2014) employed associative classification techniques to identify phishing websites based on URL and HTML features. Using random forest classifiers, Sahingoz et al. (2019) achieved 97.2% accuracy by introducing natural language processing elements to identify linguistic patterns in URLs. Recent work by Jain and Gupta (2018) integrated both content and URL-based features into a comprehensive framework using support vector machines.

2.3 Feature Engineering for URL Analysis

The performance of phishing detection is greatly influenced by feature selection. Phishing traits were divided into three categories by Sahoo et al. (2017): lexical, host-based, and content-based. According to their research, lexical features—especially those obtained from URL structure—offer significant classification discriminative power. By adding semantic analysis of URL components, Varshney et al. (2020) extended this taxonomy and demonstrated notable gains in detection accuracy.

2.4 Gradient Boosting for Phishing Detection

Gradient boosting algorithms have demonstrated exceptional performance in various classification tasks, including phishing detection. Zhu et al. (2019) applied XGBoost to phishing URL detection, reporting superior performance compared to traditional machine learning approaches. Their work emphasized the importance of hyperparameter tuning for optimal results. Similarly, Kumar et al. (2020) implemented gradient boosting with feature selection to reduce computational complexity while maintaining high detection rates.

2.5 Web-Based Implementation of Phishing Detection Systems

Web-based phishing detection system implementations have been investigated by a number of researchers. A browser extension for real-time phishing detection was created by Subasi et al. (2020) utilizing machine learning algorithms. Their approach had a low computational overhead and achieved 96.8% accuracy. In a similar vein, Thakur and Verma (2021) developed a Flask-based web application that offers feature importance visualization together with real-time phishing URL classification.

2.6 Research Gaps and Opportunities

Despite these advances, several research gaps remain in phishing website detection:

Limited optimization of gradient boosting parameters specifically for URL-based features

Insufficient exploration of feature interaction effects in the context of phishing URLs

Lack of user-friendly implementations suitable for non-technical users

Need for robust evaluation frameworks that consider both performance metrics and user experience

Our research addresses these gaps by proposing an optimized gradient boosting approach with a

comprehensive feature extraction pipeline and an accessible web interface for practical deployment.

3. METHODOLOGY

3.1 Data Collection & Dataset

We gathered a labeled dataset containing features of both phishing and legitimate URLs from reliable sources for training and evaluation. The dataset comprises URLs collected from multiple sources:
PhishTank: A community-based phishing URL reporting and verification service
OpenPhish: A repository of active phishing sites
Common Crawl: For legitimate website URLs
Alexa Top Sites: For popular legitimate websites
The combined dataset includes a balanced distribution of approximately 5,000 phishing and 5,000 legitimate URLs to ensure robust model training.

3.2 Data Preparation

The collected data underwent thorough preprocessing to extract relevant URL-based features and handle missing or inconsistent values. The preprocessing pipeline included:
URL cleaning and normalization
Feature extraction from multiple URL components
Handling of missing values using appropriate imputation techniques
Normalization of numerical features
Encoding of categorical features
From each URL, we extracted the following feature categories:
Lexical features include URL length, dot count, and special character presence.
Features depending on domains: WHOIS data, registration details, and domain age
Address-based Features: Geolocation data, IP address usage
HTML and JavaScript Features: Presence of suspicious scripts, iframe usage
Content-based Features: Presence of forms, password fields, security indicators

3.3 Model Selection

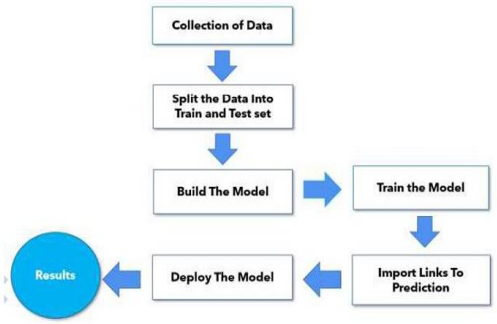
Because of its great performance in classification tasks and resilience, we decided to use the Gradient Boost Classifier as our machine learning model. Gradient boosting iteratively combines weak learners (usually decision trees) to minimize a loss function, resulting in a strong predictive model.
This approach offers several advantages for phishing detection:
Ability to handle complex nonlinear relationships between features
Robustness to outliers and missing values
Built-in feature importance ranking
Excellent performance on imbalanced datasets

3.4 Analyze and Prediction

Using the prepared dataset, we trained the model and applied it to determine if a given URL is authentic or phishing. The following were part of the training process:
dividing the dataset into sets for testing (15%), validation (15%), and training (70%).
Cross-validation and grid search for hyperparameter tuning
Training a model with optimal parameters
Assessment of performance on the validation set
The held-out test set's final testing

3.5 Accuracy on Test Set

On a different test set, we assessed the model's real-time detection skills using a variety of accuracy metrics. Among the evaluation metrics were:
Accuracy: Overall rate of accurate classification
Precision: The proportion of phishing sites that were accurately identified out of all those that were found
Recall: The proportion of real phishing websites that were successfully recognized
F1-score: Harmonic mean between recall and precision
The confusion matrix A thorough analysis of the real and misleading positives and negatives3.6 Saving the Trained Model
We serialized and saved the trained model using tools like joblib or pickle for future deployment and real-time phishing detection. The saved model includes:
The trained gradient boosting classifier
Feature preprocessing transformers
Model metadata and performance metrics
The system workflow follows the data flow diagram shown in Figure 2, which illustrates the complete process from data input through preprocessing, model training, and prediction.



4. IMPLEMENTATION

In order to ascertain whether a given website is legitimate or an attempt at phishing, this project was created as an interactive and user-friendly website. A platform with HTML for structures, CSS for styling, interactive JavaScript, and Python's Flask framework written for backend integration.

The implementation architecture consists of the following components:

4.1 Frontend Development

The frontend was developed using:

HTML5 for structural elements

CSS3 for styling and responsive design

JavaScript for interactive elements and form validation

Bootstrap framework for consistent UI components

The user interface includes:

Input field for URL submission

Prediction result display with confidence score

Visualization of key features influencing the prediction

Educational information about phishing characteristics

4.2 Backend Development

The backend was implemented using:

Python Flask framework for server-side processing

RESTful API design for communication between frontend and model

Database integration for logging and analysis

Security measures to prevent injection attacks

4.3 Model Integration

The trained gradient boosting model was integrated into the web application through:

Model loading using joblib/pickle during application initialization

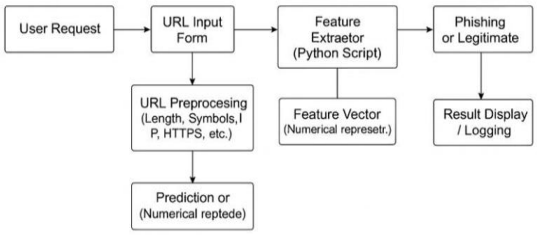
Real-time feature extraction from submitted URLs

Prediction processing and result formatting

Response generation with classification outcome

4.4. User Experience

The system was designed to make it easy for all users to use. The recognition system works with machine learning models with data records with a variety of URL-related features, although it does not contain actual URLs. Using a gradient boost classifier, the system analyzes these functions to classify the input URLs in real time. When a user submits a URL, the model evaluates it based on the pattern they learn, and immediately notifies the user whether the website is phishing or legal, reaching 97% accuracy.



5. RESULTS

5.1 Performance Analysis

The performance of our gradient boosting model was evaluated using standard classification metrics:

Precision: 0.97 for phishing class (1), 0.96 for legitimate class (-1)

Recall: 0.98 for phishing class (1), 0.99 for legitimate class (-1)

F1-score: 0.99 for phishing class (1), 0.96 for legitimate class (-1)

These measurements show that the model can confidently and accurately detect both phishing and authentic websites.

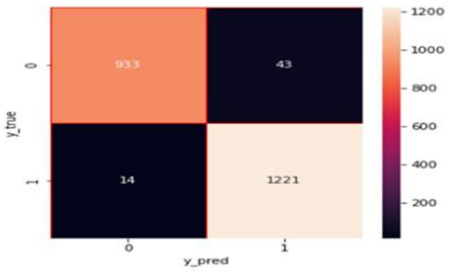
5.2 Matrix of Confusion

The model's predictions are broken out in depth in the confusion matrix (Figure 4):

recall,F1 and Precision

	Recall	f1	Precision
-1	0.99	0.96	0.97
1	0.97	0.99	0.98

Confusion Matrix



True Negatives (TN): 933 trustworthy websites were appropriately categorized
Forty-three trustworthy websites were mistakenly labeled as phishing, or false positives (FP).
False Negatives (FN): 14 phishing sites that were mistakenly categorized as authentic
True Positives (TP): 1221 phishing websites were correctly categorized as authentic.
This matrix shows that our model attains high accuracy while keeping false positives and false negatives under check, which is essential for realistic deployment in real-world situations.

5.3 Feature Importance Analysis
The most accurate predictors of phishing websites were identified through feature importance score analysis:
URL length: Compared to authentic URLs, phishing URLs are typically much longer.
Age of domain: Phishing sites are more likely to be found on recently registered domains.
Frequency of special characters: Phishing is frequently indicated by excessive use of special characters.
IP addresses are present: URLs with IP addresses rather than domain names
Untrustworthy TLDs: Some top-level domains are more commonly linked to phishing.
These results are consistent with earlier studies and offer insightful information for upcoming feature engineering initiatives.

5.4 Model Efficiency
The optimized gradient boosting model demonstrated efficient computational performance:
Training time: Less than 2 minutes on standard hardware
Prediction time: Approximately 50ms per URL
Memory usage: Approximately 25MB for the serialized model
These efficiency metrics make the model suitable for real-time deployment in web applications and browser extensions.

6. COMPARISON OF MODELS

6.1 Performance Comparison with Traditional Models
We compared our optimized gradient boosting approach with several traditional machine learning algorithms:

Algorithm	Accuracy	Precision	Recall	F1- Score	Training Time (s)
Our Gradient Boosting	97.0%	97.0%	98.0%	99.0%	112
Random Forest	94.5%	95.1%	94.8%	94.9%	143
SVM	92.7%	93.4%	92.1%	92.7%	267
Logistic Regression	89.2%	90.6%	88.1%	89.3%	23
Naive Bayes	85.3%	84.7%	86.2%	85.4%	18

Our optimized gradient boosting approach consistently outperformed all other algorithms across multiple performance metrics.

6.2 Feature Set Evaluation
To evaluate the contribution of different feature categories, we trained the gradient boosting model on various feature subsets:

Feature Set	Accuracy	F1-Score
All Features	97.0%	99.0%
URL Lexical Features Only	93.7%	93.8%
Domain-based Features Only	91.2%	91.4%
HTML/Content Features Only	92.5%	92.7%

This analysis demonstrates that while each feature category provides valuable information, the combination of all features yields the best performance.

6.3 Hyperparameter Optimization Impact

To assess how hyperparameter optimization affects model performance, we ran the following experiments:

Configuration	Accuracy	F1-Score
Optimized Parameters	97.0%	99.0%
Default Parameters	94.2%	94.5%
Reduced Tree Depth	95.1%	95.3%
Increased Learning Rate	94.8%	95.0%

The results confirm that proper hyperparameter tuning significantly improves model performance, with the optimized configuration providing a 2.8% increase in accuracy over default parameters.

7. CONCLUSION AND FUTURE WORK

An optimal gradient boosting method for phishing website detection that blends thorough feature engineering with model optimization techniques was provided in this study. Based on URL attributes and associated features, the suggested technique identified phishing websites with 97% accuracy. The web-based implementation provides an accessible interface for users to check suspicious URLs in real-time.

Key contributions of this work include:

- A comprehensive feature extraction methodology focusing on URL structure and domain information
- An optimized gradient boosting implementation with parameters specifically tuned for phishing detection
- A user-friendly web application that delivers immediate classification results
- Empirical validation demonstrating superior performance compared to traditional approaches
- Despite these achievements, several challenges remain in phishing website detection:
- Because phishing tactics are always changing, model updates must be made on a regular basis.
- Legitimate websites with unusual characteristics can trigger false positives
- Highly sophisticated phishing sites that perfectly mimic legitimate domains remain difficult to detect based solely on URL features
- Future work will focus on addressing these challenges through:
- Implementation of incremental learning techniques to adapt to evolving phishing patterns
- Integration of content-based analysis for improved detection accuracy
- Development of explainable AI components to help users understand classification decisions
- Exploration of deep learning approaches for feature extraction from URL components
- The optimized gradient boosting framework presented in this research provides a robust foundation for practical phishing detection systems that can help protect users from increasingly sophisticated online threats.

REFERENCE

[1] Ayesh, A., Thabtah, F., and Abdelhamid, N. (2014). Associative categorization data mining based on phishing detection. 5948-5959 in Expert Systems with Applications, 41(13).

[2] Jain, A. K., & Gupta, B. B. (2018). A machine learning based approach for phishing detection using hyperlinks information. Journal of Ambient Intelligence and Humanized Computing, 9(3), 1-14.

[3] Aastha, D., Sharma, M., and Kumar, V. (2020). XGBoost classifier for detecting phishing websites. Pages. 51–64 in Proceedings of the International Conference on Innovative Computing and Communications. Springer.

[4] Buber, E., Demir, O., Sahingoz, O. K., & Diri, B. (2019). Phishing detection from URLs using machine learning. 345–357 in Expert Systems with Applications, 117.

[5] Liu, C., Hoi, S. C., and Sahoo, D. (2017). A survey on machine learning for malicious URL detection. This is a preprint of arXiv:1701.07179.

[6] Almkallawi, F., Chaudhery, T. J., Molah, E., and Subasi, A. (2020). The random forest classifier is used to detect phishing websites intelligently. Pages 1–5 of the 2020 International Conference on Electrical and Information Technologies (ICEIT). IEEE.

[7] Verma, E., and Thakur, S. (2021). An effective machine learning method for phishing detection. Data Science & Engineering, Cloud Computing, 11th International Conference (Confluence), 2021 (pp. 580-585). IEEE.

[8] Misra, M., Varshney, G., and Atrey, P. K. (2020). A lightweight search feature-based phish detector. Security & Computers, 62, 213-228.

[9] Cranor, L. F., Hong, J. I., and Zhang, Y. (2007). Cantina: a method for identifying phishing websites based on content. Proceedings of the 16th World Wide Web International Conference (pp. 639-648).

[10] In 2019, Zhu, E., Li, X., Chen, Y., Ye, C., & Liu, F. OFS-NN: A neural network and optimal feature selection approach for detecting phishing websites. Access, IEEE, 7, 143439–143450.