Phishing Link Detection Using Machine Learning, Flask and Web Technologies

Abhinav Dyan Samantara, Dr. Vanitha Kakollu
PG Student, GITAM Deemed to be University, Visakhapatnam
Assistant Professor, GITAM Deemed to be University, Visakhapatnam
doi.org/10.64643/IJIRTV11I11-175397-459

Abstract: ***Phishing attacks are a serious cybersecurity risk in which individuals and businesses are targeted with the intention of stealing private data, involving financial information, usernames, and passwords. Phishing attacks are carried out by means of misleading links in emails, messages, or fake websites that imitate authentic websites. Rule-based detection techniques are no longer adequate because of the evolving nature of phishing attacks, particularly when obfuscation techniques like domain spoofing and URL shortening are used. This situation requires the employment of sophisticated and adaptive detection systems.***

To identify if a URL is authentic or phishing, the project plans to create a Phishing Link Detection System using machine learning (ML). System identifies notable features—lexical patterns, host-based features, and structural features—to train an ML model to detect phishing correctly. In contrast to static rule-based systems, the ML-based system learns new phishing patterns with time and hence increases detection accuracy.

Real-time URL analysis is supported by a Flask-based user interface, which lets users enter URLs and get predictions about the chance of phishing. System also provides explanations for predictions and stores detection results in a backend database for monitoring and analysis. With the inclusion of ML-based analysis, real-time detection, and user awareness, this project offers an effective solution to counter phishing attacks and enhance cybersecurity awareness.

Index Terms - phishing, machine learning, cybersecurity, URL classification, browser extension, feature engineering

I. INTRODUCTION

Cyberattacks have increased as a result of our increasing reliance on online platforms for communication, financial transactions, and information sharing. Phishing is the most prevalent form of online fraud. To get sensitive user data, involving banking information, login credentials, and personal information, phishing is a social engineering attack in which the attackers create fake websites that look like legitimate organizations. Phishing websites are easy to deploy and keep changing, hence hard to identify with classical security mechanisms.

Existing phishing detection methods can be generally categorized into three broad categories: blacklist-based, rule-based heuristic, and ML-based. Newly established or zero-day phishing websites are not intercepted by blacklist-based detection techniques, which depend on keeping a list of known phishing URLs. Rule-based methods use heuristic thresholds to detect malicious URLs; however, these are not much flexible in nature and generate many false positives. Advances in ML and AI (artificial intelligence) in recent times have made it possible to utilize more flexible and robust phishing detection mechanisms that can identify previously unknown phishing attacks.

Using a collection of lexical, structural, and domain-based elements that are retrieved from URLs, this work aims to develop an ML-powered phishing detection system that can determine whether a URL is phishing or real. Several ML models are trained and compared based on their ability to detect phishing assaults using a sizable labelled dataset of URLs. Class imbalance, a problem that could impair model performance, is also addressed in this research by the use of suitable data balancing techniques.

II. LITERATURE SURVEY

A sizable labelled dataset of URLs is used to train and test several ML models based on how well they can detect phishing attempts. Traditional methods involving rule-based and list-based systems have limitations in detecting new and evolving phishing strategies. With their dynamic and adaptive detection capabilities, ML-based approaches have become a more viable option. The following section reviews the findings from four research papers that contribute to phishing detection techniques.

Paper 1: Detection and Prevention of Phishing Attacks

This paper explores multiple phishing detection techniques, involving list-based, ML approaches and heuristic-based. Limits of blacklist-based detection, which depend on keeping track of known phishing URLs, are highlighted in the paper. While this technique effectively blocks previously identified phishing websites, it fails to detect zero-day attacks and minor URL modifications. The paper discusses visual similarity-based detection, where phishing websites are compared to legitimate ones using text, images, and HTML structures. This method improves detection accuracy but is computationally expensive. The study concludes that ML models involving RF (Random Forest) and NNs (Neural Networks) provide better detection rates compared to traditional approaches. Nonetheless, the research highlights the difficulties with real-time processing and feature selection, which affect how effective ML-based models are.

Paper 2: Detection of Phishing Websites Using Ensemble Machine Learning Approach

This study uses RF and XGBoost classifiers to provide an ensemble learning technique for phishing detection. The research highlights the ineffectiveness of rule-based systems due to their reliance on predefined patterns that fail to adapt to new phishing strategies. Instead, ensemble learning techniques improve detection accuracy by combining multiple classifiers to make a more robust prediction. The study focuses on feature selection, identifying key URL characteristics involving length, domain age, presence of special characters, as well as keyword analysis. The experimental results demonstrate that ensemble models outperform traditional classifiers involving Naïve Bayes and SVM, achieving higher accuracy and recall rates. The study does, however, recognize the difficulty of high-dimensional feature spaces, which can result in overfitting and more complicated computations.

Paper 3: Phishing Short URL Detection Based on Link Jumping on Social Networks

This paper addresses a unique phishing threat posed by shortened URLs in social networks. Phishing connections are frequently disguised by attackers using URL shorteners, which makes it challenging for consumers and detection systems to recognize malicious websites. To monitor link-hopping behaviour following a user's click on a brief URL, the paper presents a Hierarchical Hidden Markov Model (HHMM). By analysing redirections and multiple jump patterns, the model distinguishes between legitimate and phishing links. The research gathered and examined a dataset of 3,000 common short URLs and 1,000 phishing attempts from Weibo. HHMM model outperformed traditional phishing detection techniques, especially in handling multi-stage redirection attacks. Although the model was evaluated exclusively on social media phishing attempts rather than a wider range of phishing strategies, the article reveals limits in dataset diversity.

Paper 4: Phishing Detection Using Machine Learning and Heuristic-Based Approaches

This research explores a hybrid phishing detection approach, combining heuristic-based feature extraction with ML classification models. Study investigates various phishing characteristics, involving domain registration details, URL structure, and webpage content analysis. The research implements Random Forest and SVM classifiers, evaluating their performance on phishing datasets. The results indicate that hybrid models achieve higher detection rates compared to standalone machine learning or heuristic-based approaches. The impact of feature engineering on model accuracy is also covered in the research, which shows how choosing important phishing indicators improves classification performance. Notwithstanding these benefits, the study points out some possible drawbacks, namely reliance on heuristic rules and the requirement for frequent feature upgrades to accommodate changing phishing strategies.

III. PROPOSED SYSTEM

Phishing attacks continue to evolve, rendering traditional detection methods ineffective against zero-day threats and dynamic attack strategies. To overcome these obstacles, this study suggests an ML-based Phishing Link Detection System which increases precision as well as effectiveness of phishing detection by utilizing feature extraction, optimal classification models, and real-time detection methods.

A. Dataset

A thorough collection of URLs, involving authentic and fraudulent websites, makes up the dataset utilized

for phishing detection. Dataset is curated from multiple sources to ensure diversity, reducing bias in the model's learning process.

Dataset Composition

This dataset consists of over 800,000 URLs, carefully curated to represent a diverse range of online domains. It maintains a near-equal distribution, with approximately 52% legitimate URLs and 47% phishing URLs, ensuring a balanced dataset for effective model training.

Each entry includes two key attributes:

- URL: The web address being analysed.
- Status: A binary label where 0 indicates a phishing site (potential threat) and 1 represents a legitimate site (trustworthy domain).

By maintaining this balance, the dataset minimizes the risk of class imbalance, improving the reliability of ML models in identifying phishing threats. This methodical methodology enables academics and cybersecurity experts to create phishing detection systems that are more reliable and accurate.

B. Data Preprocessing and Feature Engineering

The system starts with data preprocessing and feature engineering in order to efficiently categorize URLs as either real or phishing. To detect questionable patterns frequently present in phishing websites, lexical data pertaining to URL length, special character count, and the existence of phishing-related terms are retrieved. Additionally, host-based attributes involving WHOIS information, domain age, and HTTPS usage are analysed to enhance classification accuracy. Class imbalance, in which phishing URLs are frequently underrepresented in comparison to legal URLs, is the main obstacle in phishing detection. To address this, the dataset undergoes resampling techniques to balance the classes, ensuring the machine learning model generalizes well to both categories.

C. Machine Learning-Based Classification

Once the features are extracted, the system employs ML algorithms to classify URLs. The classification models that have demonstrated great accuracy in phishing detection include RF, Gradient Boosting, and ensemble learning techniques. Hyperparameter tuning with GridSearchCV is used to further optimize model performance, guaranteeing that the chosen model runs with optimal parameters. The system is trained using a diverse and extensive dataset, allowing it to adapt to evolving phishing strategies and new attack patterns. This adaptability ensures robustness against emerging phishing techniques that may not be easily detectable using traditional rule-based systems.

Random Forest Classifier

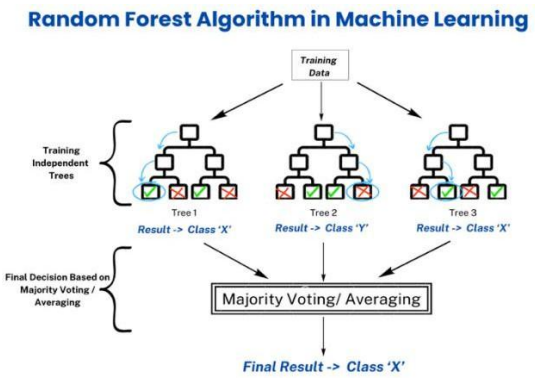

Figure 1.Random Forest Algorithm: Multiple decision trees classify data, with the final result determined by majority voting or averaging

An ensemble learning technique known as an RF Classifier builds several DTs and integrates their predictions to elevate classification accuracy. Rather than depending on a single DT, it creates a "forest" of trees, every one of which is trained using a distinct subset of the data. Model is more stable and less vulnerable to errors brought on by individual trees because the final categorization is decided by majority voting. By adding randomness to both feature selection and data sampling, RF has the major benefit of reducing overfitting, a common problem with single decision trees. It is a dependable option for phishing detection and other classification tasks since it is computationally efficient, can handle huge datasets, and performs well on both balanced and imbalanced datasets.

Gradient Boosting Classifier

An iterative ML model called the Gradient Boosting Classifier increases prediction accuracy by training many weak learners—usually DTs—one after the other. Gradient Boosting constructs trees step-by-step, learning from the mistakes of the preceding tree, in contrast to RF, which trains trees individually. By assisting the model in concentrating on incorrectly classified cases, this procedure progressively raises overall accuracy. It creates a robust prediction model

that can handle intricate patterns in data by combining several weak learners. Although Gradient Boosting provides high accuracy, it is computationally expensive compared to Random Forest, requiring more time for training, especially on large datasets. However, its capacity to identify complex correlations in data makes it an effective tool for classification jobs where accuracy is crucial, such as phishing detection.

D. Real-Time Phishing Detection and Deployment

Users may submit URLs for instantaneous classification utilizing the learned model, which is executed as a Flask-based web application for practical use. The system retrieves pertinent data from a URL submitted by the user, runs them through the trained model, and then generates a phishing probability score that indicates the possibility that the URL is fraudulent. System is designed with scalability and efficiency in mind, ensuring low-latency predictions suitable for real-world applications. By integrating an intuitive web-based interface, the proposed system enables seamless interaction, making phishing detection accessible to both technical and non-technical users.

E. Performance Evaluation

The efficacy and robustness of the proposed phishing detection system's classification are assessed using a number of important performance criteria. These metrics aid in evaluating model's ability to distinguish among authentic, phishing URLs.

Accuracy
By evaluating the percentage of correctly classified occurrences, accuracy gauges the model's overall accuracy.

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

Where:
TP (True Positive): Phishing URLs correctly classified as phishing.
TN (True Negative): Legitimate URLs correctly classified as legitimate.
FP (False Positive): Legitimate URLs incorrectly classified as phishing.
FN (False Negative): Phishing URLs incorrectly classified as legitimate.

Precision
Precision assesses the proportion of phishing-classified URLs that are, in fact, phishing. There are fewer false positives when the accuracy value is higher.

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

Recall (Sensitivity or True Positive Rate)
Recall calculate how well model identifies actual phishing URLs. An elevated recall signifies a reduction in false negatives.

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

F1-Score
When addressing class imbalance, F1-score, which is harmonic mean of recall and precision, offers a fair metric.

$$F1\text{-}Score = 2\ X \frac{Precision\ X\ Recall}{Precision+Recall}$$

ROC-AUC Score (Receiver Operating Characteristic - Area Under the Curve)
The ROC-AUC score is used to assess the model's capacity to differentiate among phishing and authentic URLs over a range of threshold values. Better classification performance is indicated by an AUC value that is greater (closer to 1).

$$AUC = \int_{-\infty}^{\infty} TPR(FPR)d(FPR)$$

Where:

TPR (True Positive Rate) = Recall

FPR (False Positive Rate) = $\frac{FP}{FP+TN}$

Feature Importance Analysis
The retrieved features (such as URL length, the presence of phishing-related keywords, and the quantity of special characters) that most contribute to phishing detection can be identified with the aid of feature significance analysis. The Gini Importance formula can be used to determine each feature's significance in a RF model:

$$Feature\ Importance = \frac{\sum(Decrease\ in\ Gini\ Index)}{Total\ Trees}$$

Where:
- The Gini Index measures the impurity of a node in the decision tree.
- A higher decrease in the Gini Index indicates a more important feature.

IV. METHODOLOGY

Phishing Link Detection System employs a systematic approach comprising data collection, preprocessing, feature extraction, model training, evaluation, and deployment. Primary objective is to develop an ML-based system capable of properly and efficiently identifying phishing or real URLs.

A. Data Collection

This dataset consists of over 800,000 URLs, carefully curated to represent a diverse range of online domains. It maintains a near-equal distribution, with approximately 52% legitimate URLs and 47% phishing URLs, ensuring a balanced dataset for effective model training.

B. Data Preprocessing and Feature Engineering

Dataset is extensively preprocessed prior to ML model training:

- Cleaning the URLs by removing duplicates and ensuring valid URL formatting.
- Lexical, host-based, as well as content-based attributes are all part of feature extraction, involving:
- URL length, occurrence of phishing-related terms, and quantity of special characters.
- Domain age, presence of HTTPS, WHOIS information, and several subdomains.
- Analysis of URL structure for redirection or obfuscation tactics.
- Handling class imbalance using oversampling techniques or weighted models to improve fairness in classification.

C. Model Selection and Training

After testing a number of machine learning models for phishing detection, RF and Gradient Boosting were chosen because of their excellent interpretability and accuracy.

- Feature scaling is performed using StandardScaler for numerical consistency.
- Data is split (80% training, 20% testing) to ensure proper model evaluation.
- To maximize model performance, GridSearchCV is used for hyperparameter optimization.
- Stratified K-Fold cross-validation guarantees that the model performs effectively when applied to various data subsets.

D. Model Evaluation and Performance Analysis

To determine the effectiveness of categorization, the trained model is assessed utilizing a variety of performance metrics:

- Accuracy – Evaluates overall accuracy.
- Precision & Recall – Ensures correct identification of phishing URLs while minimizing false positives.
- F1-Score – Strikes a balance among recall and precision for accurate classification.
- ROC-AUC Score – Assesses model's ability to distinguish among phishing and authentic URLs.
- Feature Importance Analysis – Determines which characteristics are most important for phishing detection.

E. Real-Time Deployment Using Flask

The trained model is included into a Flask-based web application for real-time phishing detection once it has reached peak performance.

- Users can enter a URL, and the trained model will process and categorize it.
- By offering a phishing probability score, the technology enables users to make well-informed choices.
- The Flask interface ensures fast, scalable, and user-friendly interaction.

V. IMPLEMENTATION

A. Random Forest Classifier

To improve classification accuracy, the RF Classifier is an ensemble learning method that builds several DTs and aggregates their predictions. Unlike a single DT, which could overfit to the training set, RF reduces variance by training each DT on a random portion of the data. A majority vote approach is used to decide the final categorization, guaranteeing a more reliable and broadly applicable model. Its capacity to handle both balanced and imbalanced datasets is one of its main advantages, which makes it a reliable option for phishing detection. Furthermore, it gives us information about the significance of features, enabling us to determine which URL attributes have the greatest influence on classification decisions.

$$P(y) = \frac{1}{N}\sum_{i=1}^{N} T_i(X)$$

Where:

- N = Number of DTs
- $T_i(X)$ = Prediction from the i^{th} decision tree

B. Gradient Boosting Classifier

A sequential ensemble technique called gradient boosting produces several poor learners, usually DT, and enhances their performance by learning from past mistakes. Gradient Boosting improves predictions at every level by modifying weights for cases that are misclassified, in contrast to RF, which trains trees independently. This method improves generality and accuracy by allowing the model to concentrate on hard-to-classify URLs. However, due to its iterative nature, Gradient Boosting is computationally more expensive compared to Random Forest, requiring more time for training. Despite this, its capacity to identify intricate links in data makes it a popular tool for phishing detection.

$$F_m(X) = F_{m-1}(X) + \gamma \hbar_m(X)$$

Where:

- $F_m(X)$ = Updated model
- γ = Learning rate
- $\hbar_m(X)$ = Weak learner trained on the residual errors

C. Feature Selection and Importance

Both RF and Gradient Boosting depend on feature selection to improve classification efficiency. To distinguish among phishing and authentic URLs, the phishing detection system extracts host-based, a variety of lexical, and content-based elements. Model predictions are heavily influenced by features involving URL length, special character presence, number of subdomains, and keywords associated to phishing. Feature importance analysis helps in identifying which attributes contribute the most to classification, enabling optimization and reducing unnecessary computation. This approach ensures that the model remains interpretable while maintaining high accuracy.

D. Hyperparameter Tuning for Optimization

To enhance model performance, GridSearchCV is used for hyperparameter tuning. Key parameters involving tree depth, number of trees, as well as minimum samples required for a split are adjusted to find the best combination. By doing this, overfitting is avoided and model's ability to generalize to unknown URLs is guaranteed. By assessing the model's performance on several data subsets, cross-validation methods such as Stratified K-Fold Cross-Validation help to further confirm model's stability. By fine-tuning hyperparameters, the phishing detection system achieves a balance of accuracy, computational efficiency, and precision.

E. Model Deployment and Real-Time Detection

A Flask-based web application is used to deploy the model for real-time phishing detection after it is trained and improved. URL entered by the user is analyzed by the trained model to identify if it is authentic or phishing. System extracts features from the given URL, applies feature scaling, and passes it through the classification model. The final prediction is displayed to the user along with a confidence score, indicating the likelihood of the URL being phishing. The deployment ensures low-latency predictions, making the system scalable and efficient for practical cybersecurity applications.

VI. RESULTS

The dataset used for phishing link detection contained a total of 822,010 URLs, consisting of 427,028 phishing URLs and 394,982 legitimate URLs. The dataset included multiple extracted features, involving num_special_chars, is_https, url_length, num_subdomains, and has_phishing_keyword. To ensure data completeness, a preliminary check of missing values verified that there were no null entries. The dataset exhibited a slight imbalance, with phishing URLs making up approximately 52% of the data, while legitimate URLs accounted for the remaining 48%.

A statistical analysis of key features revealed that the average URL length was 47.09 characters, with a minimum of 1 character and a maximum of 3,992 characters. The HTTPS protocol was present in only 16.15% of the URLs, suggesting that a significant portion of phishing URLs still operate without secure connections. Additionally, 6.67% of all URLs contained phishing-related keywords, which can be an important indicator of malicious intent. To guarantee efficient model evaluation, the dataset was categorized into a training set (657,608 samples) as well as testing set (164,402 samples).

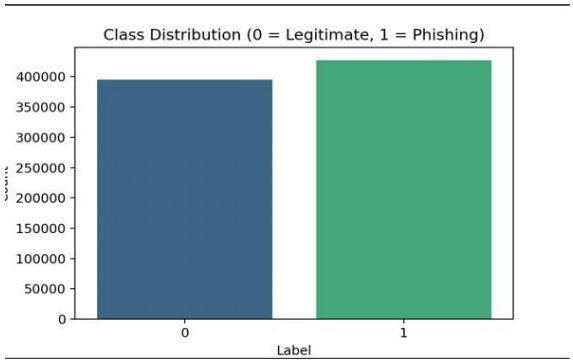

Figure 2: Distribution of phishing and legitimate URLs in the dataset.

An RF classifier was used for model training, and cross-validation was carried out to guarantee robustness. The cross-validation accuracy scores across five folds were [0.8041, 0.8009, 0.8028, 0.8009, 0.8016], resulting in an average accuracy of 80.21%. This suggests that the algorithm has a comparatively high level of accuracy in differentiating among phishing and authentic URLs. Using Grid Search, hyperparameter tuning was carried out to further improve performance by experimenting with different max_depth, min_samples_split, n_estimator and min_samples_leaf, combinations. Training times for different configurations ranged from 15 seconds to over one minute, reliant on the complexity of the model parameters.

```
Classification Report with Best Model:
              precision    recall  f1-score   support

           0       0.83      0.74      0.78     79122
           1       0.78      0.86      0.82     85280

    accuracy                           0.80    164402
   macro avg       0.81      0.80      0.80    164402
weighted avg       0.81      0.80      0.80    164402

Model and scaler saved successfully!
ROC-AUC Score: 0.8770859342873402
```

Figure 3: Model Performance

The findings imply that the extracted features—specifically, URL length, special character presence, and HTTPS usage—play a major role in phishing identification. Although additional improvements, involving feature engineering and ensemble approaches, could further enhance accuracy, the model's performance shows that ML can help detect phishing websites. Future work may also involve addressing class imbalance through data augmentation or weighted classification techniques to enhance model generalizability.

Figure 4: Final Website for Phishing Link Detection

REFERENCES

[1] Gary Warner & Anthony Skjellum(2011). *High-performance content-based phishing attack detection.*

[2] Dharani M., Soumya Badkul, Kimaya Gharat, Amarsinh Vidhate & Dhanashri Bhosale(2021). *Detection of Phishing Websites Using Ensemble Machine Learning Approach.* International Conference on Automation, Computing and Communication 2021 (ICACC-2021).

[3] Abu Saad Choudhary, Rucha Desai, Lavkush Gupta, Madhuri Gedam(2021). *Detection and prevention of Phishing* attacks. Asian Journal of Convergence in Technology(AJCT) 7(1):193-196.

[4] Siti Nur Aqilah Kamarudin, Isredza Rahmi a Hamid, Cik Feresa Mohd Foozy, Zubaile Abdullah(2022). *Feature selection approach to detect phishing websites using a machine learning algorithm.* AIP Conference Proceedings 2644(1):040003.

[5] Rania Zaimi, Mohamed Hafidi & Mahnane Lamia(2024). *A deep learning mechanism to detect phishing URLs using the permutation importance method and SMOTE-Tomek link.* The Journal of Supercomputing (2024) 80:17159–17191.

[6] Orunsolu Abdul(2020). *Linkcalculator -An Efficient Link-Based Phishing Detection Tool.* Acta Informatica Malaysia 4(2):37-44.

[7] *Bailin* Xie, *Qi* Li & *Na* Wei. *Phishing short URL detection based on link jumping on social networks.* ITM Web of Conferences 47, 01009 (2022).

[8] Du Shu-Ying & He Wang(2019). *Phishing Website Detection Algorithm Based on Link Structure.* IOP Conf. Ser.: Mater. Sci. Eng. 563 052091.

[9] Lizhen Tang & Qusay H. Mahmoud(2021). *A Survey of Machine Learning-Based Solutions for Phishing Website Detection.*

[10] Chidimma Opara, Yingke Chen, Bo Wei. *Look Before You Leap: Detecting Phishing Web Pages by Exploiting Raw URL And HTML Characteristic*

AUTHORS PROFILE

Abhinav Dyan Samantara, pursuing Master of Science(Data Science), Department of CS, GSS, GITAM (Deemed to be University), Visakhapatnam. His area of interest in Machine Learning and Deep Learning

Dr. Vanitha Kakollu is currently working as an Assistant Professor in the Department of Computer Science, GSS, GITAM(Deemed to be University). Her main areas of research include Machine Learning and Data Mining.