A Study on Phishing Website Detection using Machine Learning

Mahmudul Haque Tafhim
dept. of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh

Tabia Morshed
dept. of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh

Tanvir Md Raiyan
dept. of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh

Musaddique Ali Erfan
dept. of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh

***Abstract*—Phishing attacks pose a persistent cybersecurity threat, tricking users into divulging sensitive information through fraudulent websites. This study examines how machine learning models—specifically Random Forest, K-Nearest Neighbor (KNN), and Extra Trees (ET)—can effectively detect phishing URLs. Using datasets like the UCI phishing website dataset, the research evaluates key features that distinguish between legitimate and malicious URLs. Performance metrics such as accuracy, precision, recall, and false positive rates are employed to measure each model's effectiveness. The study identifies strengths and limitations of these algorithms and proposes future research directions to improve phishing detection systems.**

***Index Terms*—Phishing attack, Machine learning, KNN, Network security, Random Forest, URLs, Phishing detection, Ensemble classifier, Hyperparameter tuning, Fake website, Fraudulent, Identification, Authentication, Feature selection.**

I. Introduction

Phishing is a type of cyber attack that targets individuals by masquerading as legitimate entities to steal sensitive information such as usernames, passwords, and financial details. These attacks often involve deceptive emails, messages, or websites that trick users into providing their personal data. Phishing remains a significant threat in the cybersecurity landscape due to its simplicity and effectiveness. The rise in online transactions, e-commerce, and digital banking has increased the vulnerability of users to phishing attacks. The economic and personal damage caused by phishing is substantial, necessitating the development of robust detection mechanisms to protect users from such fraudulent activities. In this paper, we consolidate findings from six different studies on phishing detection. These studies cover various aspects of phishing, including its prevalence and impact, and propose multiple machine learning-based methods to enhance detection accuracy and efficiency. By analyzing the strengths and limitations of these approaches, we aim to provide a comprehensive overview of current advancements in phishing detection and suggest potential areas for future research to improve user safety against phishing attacks.

II. Literature Review

[1] In this paper, a more robust approach is adopted by using the Random Forest algorithm to predict fake websites with a higher accuracy rate of 96%. Unlike previous studies, this project focuses on real-time detection, allowing users to enter URLs and receive instant feedback. The system utilizes a large dataset of 400,000 entries, sourced from Jcharis/Machine-Learning-In-Julia-JcharisTech, ensuring comprehensive coverage of potential fake websites. The system's architecture involves sending URLs not present in the database to the admin for processing and updating the status in the MongoDB Cloud database, enhancing its adaptability and accuracy over time. This approach addresses the limitations of previous studies by providing a scalable, efficient, and user-friendly solution for fake website detection.The proposed system does not block fake websites; it only displays the status as good or bad. Future enhancements could include functionality to block fake websites or provide alerts when users encounter them.

[2] This paper addresses the challenge of phishing detection by focusing on optimizing feature selection for machine learning models. By combining two datasets to identify 18 common features and reducing this set to 13 optimal features through feature selection techniques, the authors aim to build a more efficient and robust phishing detection model. Key contributions include emphasizing feature selection to improve model efficiency, integrating multiple datasets for comprehensive feature selection, and achieving a detection accuracy of 93.7% with the selected features, demonstrating the effectiveness of fewer, well-chosen features. The practical application is highlighted by addressing the impracticality of using large feature sets in real-time detection systems. However, the study does not focus on the feasibility of real-time feature extraction, which is critical for practical deployment. Future research could explore testing the real-time extraction of these optimal features to enhance the system's practicality. By concentrating on feature selection, this paper provides a streamlined and practical approach to phishing website detection, potentially

improving the efficiency and effectiveness of machine learning models in this domain.

[3] The paper "K-Nearest Neighbor Based URL Identification Model for Phishing Attack Detection" investigates the efficacy of the K-Nearest Neighbor (KNN) algorithm in identifying phishing URLs. Utilizing data from the Kaggle Website phishing data repository, the study encompasses 1,353 observations with 10 descriptive features. The correlation analysis revealed URL length as a significant indicator of phishing attacks. The model, implemented in Python within a Jupyter Notebook environment, demonstrated a peak accuracy of 87.82% at K=10, indicating the importance of selecting an optimal K value for performance. The confusion matrix analysis on a test set of 106 observations highlighted a few misclassifications, suggesting areas for improvement. However, the model's overall performance was promising, achieving an 85.08% accuracy rate. A noted limitation is the dataset's small size, potentially affecting generalizability. Future research should explore other machine learning algorithms like SVM, decision trees, and random forests, and expand the dataset to enhance model robustness. Additionally, focusing on the real-time applicability and efficiency of feature extraction could further improve phishing detection systems' practical deployment.

[4] The increasing prevalence of phishing attacks, particularly during the COVID-19 pandemic, has heightened the need for effective detection mechanisms. Various machine learning techniques have been employed for phishing website detection, with approaches such as Decision Tree Classifiers, K-Nearest Neighbors, Linear SVC Classifiers, Random Forest Classifiers, and One-Class SVM Classifiers being explored. Random Forest has shown the highest accuracy of about 96.87% in comparison to other methods, making it a preferred choice due to its robustness against overfitting. Despite the advancements, current techniques often struggle with high rates of false positives and the need for extensive labeled training data, which is difficult to obtain and maintain. Future work should focus on developing methods that require minimal training data and improving the detection accuracy without compromising the speed of identification. Additionally, integrating advanced neural network models and enhancing feature selection techniques could further refine phishing detection systems.

[5] In this paper, the authors address the growing threat of phishing attacks, which exploit various communication channels such as emails and messages to deceive users into clicking malicious links. Existing research has extensively explored the use of machine learning algorithms for detecting phishing websites, with the Random Forest algorithm demonstrating high accuracy and efficiency. This study proposes a system that utilizes a user interface (UI) to analyze URLs based on nine features, including URL length, presence of HTTP, and suspicious characters. The Random Forest algorithm is employed to classify URLs as phishing or legitimate, achieving an accuracy of 85%. Despite its effectiveness, the system has limitations, such as requiring users to manually copy and paste URLs into the UI and the challenge of predicting URLs not included in the training dataset. Future work should focus on enhancing the system's capabilities by automating URL detection directly within applications, expanding feature analysis to include malicious clickable images and QR codes, and improving the algorithm's ability to handle new, unseen URLs. These advancements aim to create a more seamless and robust phishing detection system, ultimately enhancing user protection and data security.

[6] This paper explores the effectiveness of ensemble classifiers, specifically Random Forest (RF) and Extra Trees (ET), in detecting phishing websites using the UCI phishing website dataset. The dataset encompasses 11,055 entries categorized into 6,157 legitimate and 4,898 phishing URLs, characterized by 30 distinct features. The study investigates both baseline models and hyperparameter-tuned variants of RF and ET, evaluating their performance based on key metrics such as accuracy, precision, recall, and false positive rate. Results indicate that the baseline ET model achieves the highest accuracy at 97.47% with a commendably low false positive rate of 3.76%. However, limitations include the performance being restricted to the specific dataset used, which may not generalize to all phishing detection scenarios. Future directions include expanding the ET classifier's application to larger datasets capable of identifying spam URLs and enhancing its ability to predict multiple URLs simultaneously.

III. Datasets

The papers reviewed employ diverse datasets to advance the detection and classification of phishing and legitimate websites using machine learning techniques.

A. JChris Machine Learning Github Repository

- Dataset Details: Contains 400,000 entries consisting of website URLs paired with their status (good or bad). This dataset is essential for training the Random Forest algorithm to differentiate between legitimate and fake websites.[8]

B. DS1-30 and DS2-48

- Dataset Details:
  DS1-30 includes internal and external features derived from webpage content and third-party sources. DS2-48 consists solely of internal features.The study focuses on optimizing feature selection using statistical methods like variance inflation factor (VIF) and p-values to enhance detection accuracy while minimizing computational overhead.

C. Kaggle's Website Phishing Data repository

- Dataset Details: Contains 1,353 observations categorized into malicious and legitimate classes, each described by 10 features. Features include URL length, presence of pop-up windows, age of domain, and other indicators crucial for distinguishing phishing from legitimate URLs.

D. MillerSmiles archive and Phish Tank archive

- Dataset Details: Features 30 parameters identified through data mining algorithms, specifically chosen for their relevance to phishing detection. This dataset emphasizes human interaction in identifying phishing-related features critical for assessing vulnerability to phishing attacks.

E. Phishing and legitimate URLs from phishtank.com

- Dataset Details: Focuses on 9 key features crucial for determining the legitimacy of URLs, using a dataset designed to train a Random Forest model. This dataset helps in classifying URLs effectively, enhancing user security against phishing attacks.[7]

F. UCI Machine Learning Repository dataset

- Dataset Details: Contains 11,055 entries labeled '-1' for phishing URLs and '1' for legitimate URLs, featuring 30 comprehensive features. This dataset facilitates research into developing robust algorithms for accurately distinguishing phishing URLs from legitimate ones, contributing significantly to cybersecurity efforts.[9]

These datasets play a pivotal role in advancing machine learning techniques for phishing detection, leveraging diverse features and methodologies to enhance accuracy and effectiveness in identifying malicious websites.

IV. Pre-processing Techniques

The papers reviewed employ diverse preprocessing on datasets.

A. Data Cleaning

Data Cleaning involves removing duplicates, handling missing values through imputation or removal of rows with missing data, and ensuring data consistency.

B. Feature Engineering

Feature Engineering includes:

- Extracting relevant features from URLs.
- Converting categorical features into numerical values.
- Transforming raw URL data into meaningful numerical values suitable for machine learning.

C. Data Preparation

Ensuring the data is in a suitable format for machine learning models, potentially involving:

- Data scaling (normalization or standardization).
- Merging datasets based on common features.
- Data partitioning.

V. Models

The six papers on phishing detection using machine learning algorithms utilize various models to detect phishing URLs.

A. Random Forest (RF)

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (phishing or legitimate) as the prediction. It excels in handling large datasets and is robust against overfitting.

B. Gradient Boosting Machine (GBM)

A Gradient Boosting Machine (GBM) is an ensemble learning technique that builds models sequentially, each new model attempting to correct the errors made by the previous ones. It combines the predictions of multiple weak learners, typically decision trees, to produce a strong predictive model. The method uses gradient descent to minimize a loss function, enhancing the model's accuracy over iterations.

C. K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) is a simple, non-parametric algorithm used for classification and regression tasks. It classifies data points based on the majority class among its 'k' nearest neighbors in the feature space, typically determined by a distance metric such as Euclidean distance. KNN is intuitive and effective but can be computationally expensive, especially with large datasets, as it requires calculating the distance from each point to all others.

D. Decision Tree Classifiers

Decision Tree Classifiers are a type of supervised learning algorithm used for classification tasks. They split the data into subsets based on the value of input features, creating a tree-like model of decisions where each node represents a feature, each branch represents a decision rule, and each leaf node represents an outcome. This method is easy to interpret and visualize, but can be prone to overfitting, especially with complex trees.

E. Linear SVC

Linear Support Vector Classification (Linear SVC) is a type of Support Vector Machine (SVM) that seeks to find the best linear boundary (hyperplane) that separates data into classes. It works by maximizing the margin between the classes, which is the distance between the hyperplane and the nearest data points from each class. Linear SVC is efficient for high-dimensional datasets and linearly separable data, but may not perform well with non-linear relationships unless combined with kernel tricks.

F. One-Class SVM

One-Class SVM is an unsupervised learning algorithm used primarily for anomaly detection. It models the normal data distribution by finding a boundary that encapsulates the majority of the data points and identifies points that lie outside this boundary as anomalies. This technique is particularly useful for applications where the goal is to identify outliers or rare events in datasets.

G. Extra Trees (ET) Classifier

The Extra Trees (Extremely Randomized Trees) Classifier is an ensemble learning method that builds multiple decision trees using randomly selected subsets of the training data and features. Unlike other tree-based methods, it introduces more randomness by selecting split points at random for each feature, rather than choosing the best possible split. This approach helps in reducing variance and often results in robust models that are less prone to overfitting.

VI. Evaluation Metrics

Various evaluatuion metrics were used in these papers.

A. Accuracy

Accuracy measures the overall correctness of the model, indicating the proportion of correctly classified instances (both true positives and true negatives) out of the total instances.

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

B. Precision

Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive. It indicates how precise the model is in predicting positives.

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

C. Recall

Recall measures the proportion of actual positive instances that were correctly predicted by the model. It indicates the model's ability to identify all positive instances.

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

D. F1-score

F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. It is especially useful when there is an uneven class distribution.

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

The evaluation metrics of these papers are given in Table I.

TABLE I
Comparison of Phishing Detection Methods

Paper Name	Acc.	Rec.	Prec.	F1
Extra Trees	96.85	98.31	96.12	-
Eff. ML Approach	**1C SVM**			
	48.56	-	-	-
	Linear SVM			
	92.69	-	-	-
	KNN			
	93.53	-	-	-
	DT			
	96.05	-	-	-
	RF			
	96.87	-	-	-
KNN URL	87.82	-	-	-
Feat. Sel. ML Phishing	**DRF**			
	98.5	98.9	98.1	-
	GMB			
	98.8	99.2	98.3	-
Feat. Sel. ML Phishing	**LR**			
	92	92	93	93
	DT			
	94	94	94	94
	RF			
	96	97	97	97
	GB			
	94	95	95	95
URL Detection	86	-	-	-

VII. Research Gap

By observing these papers on phishing detection using machine learning algorithms, several research gaps emerge that could guide future studies. "Fake Website Prediction Using Random Forest" focuses on real-time detection but lacks functionality to block or alert users about identified fake websites, suggesting a need for enhanced user protection mechanisms. "Feature Selection for Machine Learning-based Phishing Websites Detection" optimizes feature selection but does not address real-time feature extraction feasibility, essential for practical deployment. The study on "K-Nearest Neighbor Based URL Identification Model for Phishing Attack Detection" highlights the need for expanding datasets and exploring other algorithms like SVM and decision trees to improve model robustness and generalizability. "Phishing website detection based on effective machine learning approach" identifies challenges with high false positive rates and the requirement for extensive labeled data, suggesting a need for methods that minimize training data dependency and enhance detection speed. "Phishing Website Detection Based on URL" emphasizes the manual entry of URLs into the system and the challenge of predicting unseen URLs, indicating a need for automated URL detection and broader feature analysis. Finally, "Phishing Detection using Extra Trees Classifier" discusses limitations in model generalizability beyond specific datasets, calling for research into broader application scenarios and improved prediction capabilities for diverse phishing threats. Addressing these gaps could lead to more effective and adaptive phishing detection systems, enhancing cybersecurity measures against evolving online threats.

VIII. Conclusion

In conclusion, this study has delved into various machine learning approaches, including Random Forest, K-Nearest Neighbor (KNN), and Extra Trees (ET), for detecting phishing URLs, showcasing their effectiveness and highlighting areas for improvement. Each algorithm demonstrated promising results: Random Forest excelled in real-time detection with a 96% accuracy rate, emphasizing its suitability for immediate threat mitigation. Meanwhile, optimized feature selection techniques enhanced model efficiency, achieving a notable 93.7% accuracy, underscoring the importance of selecting

pertinent features for effective phishing detection systems. KNN showed potential with an 87.82% accuracy, albeit with challenges related to dataset size and generalizability, suggesting avenues for future exploration into algorithmic diversity and dataset expansion. Despite these advancements, challenges such as high false positives and the reliance on extensive labeled data remain prevalent in current phishing detection methodologies. These shortcomings highlight critical research gaps that demand innovative solutions. Future efforts should focus on developing algorithms that minimize false positives, reduce dependency on large datasets through enhanced feature extraction techniques, and adapt to evolving phishing tactics in real-time. Moreover, integrating advanced neural network models and expanding feature analysis to encompass newer phishing vectors like clickable images and QR codes could further bolster detection accuracy and system resilience. By addressing these challenges, future phishing detection systems can better protect users' digital assets and privacy, fortifying cybersecurity measures against increasingly sophisticated threats in the digital landscape.

IX. Contribution of Group Members

Musaddique wrote the abstract, introduction, and sections on preprocessing techniques. Tabia was responsible for the literature review and the conclusion. Tafhim focused on the evaluation metrics and identified the research gap. Raiyan described the datasets and models used in our project.

References

[1] Antony M, Ashna. "Detecting Phishing Websites Using Datamining-International Research Journal of Engineering and Technology (IRJET) June 2020". International Research Journal of Engineering and Technology, vol. 7, no. 6, 2020.

[2] SanthanaLakshmi V, Vijaya MS. "Efficient prediction of phishing websites using supervised learning algorithms-International Conference on communication Technology and System Design 2011". International Conference on communication Technology and System Design, 2011.

[3] Patil, Sagar, Yogesh Shetye, and Nilesh Shendage. "Detecting Phishing Websites Using Machine Learning", International Research Journal of Engineering and Technology Volume: 07 Issue: 02 — Feb 2020. International Research Journal of Engineering and Technology, vol. 7, no. 2, 2020.

[4] Kulkarni, Arun, and Leonard L. Brown. "Phishing Websites Detection using Machine Learning", Article Published in International Journal of Advanced Computer Science and Applications (IJACSA), Volume 10 Issue7, 2019. International Journal of Advanced Computer Science and Applications (IJACSA), vol. 10, no. 7, 2019.

[5] Abutaira, Hassan Y. A., and Abdelfettah Belghith. "Using Case-Based Reasoning for Phishing Detection",The 8th International Conference on Ambient Systems, Networks and Technologies (ANT2017). The 8th International Conference on Ambient Systems, Networks and Technologies (ANT2017), 2017.

[6] Pujara, P., & Chaudhari, M. B. (2018). Phishing Website Detection using Machine Learning: AReview. Phishing Website Detection using Machine Learning: A Review, 2018.

[7] Mohammad, Rami M., Fadi Thabtah, and Lee McCluskey. "Phishing Websites dataset." University of California, Irvine Machine Learning Repository. Accessed January.

[8] Jcharis/Machine-Learning-In-Julia-JCharisTech. "urldata.csv". Github. https://github.com/Jcharis/Machine-Learning-In-Julia-JCharisTech/blob/master/urldata.csv. Accessed June 2021.

[9] Tan, C. L. "Phishing Dataset for Machine Learning: Feature Evaluation," Mendeley Data, 24-Mar-2018. https://data.mendeley.com/datasets/h3cgnj8hft/1. Accessed June 2021.