Journal of Research Administration
Society of Research Administrators International

"MACHINE LEARNING IN CYBERSECURITY: COMPREHENSIVE ANALYSIS AND DETECTION OF URL PHISHING ATTACKS"

Dr. Biren Patel
Assistant Professor, Department of Computer Science, Ganpat University.

Abstract:
In the face of escalating cyber threats, particularly phishing attacks, this research provides a comprehensive analysis of machine learning techniques for effective phishing URL detection. Leveraging a meticulously curated dataset comprising 10,000 webpages, evenly split between phishing and legitimate sites, and gathered from January 2015 to June 2017, the study employs advanced feature extraction using Selenium WebDriver, surpassing conventional data collection methodologies. A lot of different machine learning algorithms were carefully tested. These included ensemble methods like Random Forest Classifier and XGBoost, as well as more traditional models like Logistic Regression and GaussianNB. The analysis focused on critical performance metrics: accuracy, precision, recall, and F1-score. Results revealed that ensemble models, particularly XGBoost, outshine others with a remarkable accuracy of 96.0% and an equally impressive F1-Score of 96.0%, setting a new benchmark in phishing URL detection. This research not only gives a thorough comparison of different machine learning methods, but it also shows that advanced ensemble techniques are better at solving cybersecurity problems. It opens avenues for future exploration in deep learning and real-time application of these models, underscoring the potential of machine learning in fortifying defenses against continually evolving cyber threats.

I. Introduction:

The internet's rapid growth has altered the manner in which we work, communicate, and live, introducing conveniences and prospects that were previously unimaginable. With the continuous proliferation of internet users, the digital environment has become an indispensable component of daily existence [1]. Despite its extraordinary scale, this expansion is not devoid of obstacles. The extensive integration of the internet has facilitated the proliferation of cyber threats, creating serious risks for both individual users and organizations. The enormous quantity and convenience of access to sensitive and personal information online render the digital field an attractive target for cybercriminals [2].

URL phishing assaults have become notable among the diverse array of cybersecurity threats due to their pervasiveness and deceitfulness. These attacks encompass the development of illegitimate websites that bear resemblance to authentic ones with the intention of misleading users into disclosing confidential information, including login credentials, financial data, or personal particulars. Phishing attacks manifest in diverse modalities, encompassing fraudulent websites, deceptive electronic mail, and social engineering strategies. Due to their complexity and diversity, these attacks have proven to be exceptionally difficult to detect and thwart, thereby posing a substantial risk to the security and privacy of internet users [3] .

Historically, the detection of a malicious URL has been accomplished through a blend of user awareness and fundamental security protocols. Users are frequently advised to be on the lookout for phishing indicators, including URLs that are misspelled, emails that lack secure protocols (HTTPS), and content that appears dubious. However, due to the increased sophistication of deception techniques, these conventional approaches are frequently inadequate. Identifying a malicious website from a legitimate one can be extremely difficult, if not impossible, for the average non-IT professional. This challenge highlights the necessity for more sophisticated and automated detection techniques that can adjust to the ever-changing strategies employed by cybercriminals [4].

As a result of these obstacles, machine learning has become an effective instrument in the battle against phishing attacks [5]. Machine learning provides a dynamic method for identifying fraudulent URLs through the utilization of algorithms capable of information acquisition and prediction. In order to enhance the speed and precision of phishing threat identification while decreasing the dependence on human judgment, scholars have investigated a range of machine learning methodologies with the objective of automating the detection procedure [6] . A wide array of techniques are utilized, spanning from rudimentary classification algorithms to intricate ensemble models, with each presenting distinct advantages in terms of efficiency and detection capabilities [7][8].

An exhaustive examination of multiple machine learning algorithms applied to a dataset consisting of 10,000 legitimate and phishing-labeled webpages is how this paper contributes to this expanding body of knowledge. We check how well different models can sort webpages by looking at how well experimental feature extraction techniques and models like Random Forest Classifier, Gradient Boosting Classifier, and Ada Boost Classifier work [9]. Our results establish a foundation for forthcoming research prospects and offer valuable insights into the efficacy of various machine learning methodologies in fraud detection [10]. In order to bolster the resilience of digital systems against phishing assaults, we investigate the possibility of incorporating these techniques into more comprehensive cybersecurity strategies.

Research Contributions:

This study makes several pivotal contributions to the domain of URL phishing detection using machine learning:

1. Introduction of an Advanced Feature Extraction Technique: Selenium WebDriver was utilized to capture data with greater precision and robustness than conventional regex-based methods.

2. Extensive Evaluation of Multiple Machine Learning Models: A comparative analysis was performed on a variety of phishing detection models, including Random Forest Classifier, Gradient Boosting Classifier, and Ada Boost Classifier.

3. Compilation of a Unique Phishing and Legitimate Webpages Dataset: A dataset comprising 10,000 webpages, obtained from PhishTank, OpenPhish, Alexa, and Common Crawl, has been compiled, offering a significant asset for research in the field of cybersecurity.

4. Benchmarking Data for Phishing Detection Algorithms: Facilitated future phishing detection research and development by providing benchmarking insights for a variety of machine learning algorithms.

5. Practical Implications for Enhancing Cybersecurity Strategies: Contributing to the advancement of more robust digital defense mechanisms by furnishing actionable insights into the efficacy of machine learning models in spoofing URL detection.

II. Methodology:

1. Dataset Collection:

For the purposes of this research, we employed a dataset consisting of 10,000 webpages that were gathered from January 2015 to June 2017. Phishing websites were obtained through the utilization of PhishTank and OpenPhish, whereas legitimate websites were compiled using Alexa and Common Crawl. The dataset was meticulously curated to incorporate an equitable distribution of phishing and legitimate websites, thereby furnishing an all-encompassing structure for examination.

2. Data Description and Dataset Labeling:

The dataset comprises 48 unique features that were extracted utilizing the Selenium WebDriver, a methodology that provides improved accuracy compared to conventional regex-based techniques. A numerical value of '1' denoted legitimate webpages and '0' phishing sites.

Table 1: Dataset Sample Count

Class Label	Description	Count
1	Legitimate	5000
0	Phishing	5000

3. Splitting the Dataset into Training and Testing:

In order to facilitate a rigorous analysis and enable the evaluation of the machine learning models, the dataset was partitioned into separate training and testing sets.

Table 2: Dataset Split

Dataset Type	Legitimate	Phishing	Total
Training	3500	3500	7000
Testing	1500	1500	3000

4. Performance Measurement Metrics:

Performance metrics, including accuracy, precision, recall, and F1-score, were used to assess the performance of every machine learning model. True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) were utilized in the calculation of these metrics.

Table 3: Performance Measurement Metrics Equations

Metric	Equations
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F1-Score	$2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall}$

By adhering to this methodological framework, machine learning models for URL phishing detection can be evaluated exhaustively and rigorously, ensuring that the efficacy of each algorithm is thoroughly documented.

III. Machine Learning Algorithms Used in Methodology

A variety of machine learning algorithms were implemented, which include Random Forest Classifier, GaussianNB, Stacking Classifier, Voting Classifier, Decision Tree Classifier, Logistic Regression, SVC, KNeighbors Classifier, Stacking Classifier, and XGBoost, among others. The models were chosen based on their varied proficiencies in classification tasks, thereby offering an all-encompassing assessment of their efficacy in phishing detection [11] [12] [13].

1. **The Random Forest Classifier:** The Random Forest Classifier is an ensemble learning method that functions by generating the mode of the classes (i.e., the majority vote) of the individual trees while training a large number of decision trees [14]. Its efficacy is notably attributed to its approach of generating numerous trees and deliberating via the majority vote of these trees, thus mitigating the potential for overfitting [15][16][17]. The decision rule in a tree can be represented as $y = f(x)$, where y is the output class, and x represents the input features.

2. **Gradient Boosting Classifier:** The Gradient Boosting Classifier is an additional ensemble method that incrementally constructs the model. After constructing new trees that forecast the residuals or errors of previous trees, it merges these trees in order to enhance the predictive accuracy of the model. The mathematical representation of its decision-making process is an additive model [17].

$F(x) = \sum_{i=1}^{M} yihi(x)$, where $hi(x)$ are the weak learners (trees) and yi are the coefficients

3. **Ada Boost Classifier:** Ada Boost Classifier, which is an abbreviation for Adaptive Boosting, operates by employing a series of weak learners in a sequential fashion and adjusting the weight of each instance according to the accuracy of its predecessor [18]. In the end, the model produces a weighted sum of the following weak classifiers:

$F(x) = \sum_{i=1}^{N} \alpha iCi(x)$, where $Ci(x)$ are the weak classifiers and αi are the weights.

4. **Voting Classifier:** A novel approach, Voting Classifier integrates machine learning classifiers that are conceptually distinct. It generates forecasts by averaging the probabilities predicted by the combined classifiers or by the majority vote [19]. This approach capitalizes on the advantages of multiple standalone models, consequently enhancing the resilience and precision of the ultimate prognostications [20].

5. **The Decision Tree Classifier:** The Decision Tree Classifier operates by constructing a model that uses simple decision rules inferred from the data features to determine the value of a target variable. Each internal node signifies an attribute test, every branch represents the test's result, and every leaf node represents a class label; this is a non-parametric method [21].

6. **Logistic Regression:** Logistic regression is a classification model that employs linearity [17]. The dependent variable is binary when it is utilized. The logistic function used in this classifier can be written as

   $log(\frac{p}{1-p}) = \beta_0 + \beta_0 x$, where p is the probability of the positive class.

7. **Support Vector Classifier:** SVC, or Support Vector Machines, is highly suitable for the task of binary classification. To partition distinct classes, it generates a hyperplane in a multidimensional space; the optimal hyperplane is determined by its distance from the nearest training data points of each class [22].

8. **KNeighbors Classifier:** The KNeighbors Classifier is classified as non-generalizing learning or instance-based learning. Instead of striving to develop an overarching internal model, it merely retains specific instances of the training data. The process of classification involves the simplest majority vote of the points' adjacent neighbors [23].

9. **Gaussian Naive Bayes Classifier:** GaussianNB works under the "naive" assumption of independence between every combination of features and is constructed by applying Bayes' theorem [18]. The assumption that the features' likelihood is Gaussian and its equation are

   $$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-(\frac{(x_i-\mu_y)^2}{2\sigma_y^2})}$$

   where μ_y and σ_y^2 are the mean and variance of the features in class y

10. **Stacking Classifier:** A stacking classifier employs a meta-classifier to make predictions based on the outputs of multiple base classifiers. Each of the base classifiers is trained on the full dataset, and then the meta-classifier is trained to best combine their predictions [24].

11. **eXtreme Gradient Boosting:** XGBoost is a scalable and effective gradient boosting implementation [25]. It corrects the residual errors introduced by the preceding predictors

in the chain by adding predictors in succession, thereby fitting new predictors to the errors introduced by their predecessors [26].

Applying each of these methodologies to the dataset generates a comprehensive and resilient analysis by utilizing unique computational approaches and benefits for spoofing URL detection.

IV. Results Analysis:

This section provides a comprehensive analysis of the results generated when different machine learning algorithms were implemented to detect fraudulent URLs. Each algorithm is subjected to a thorough analysis of performance metrics, including F1-score, accuracy, precision, and recall, as part of our exhaustive evaluation. These metrics offer valuable insights regarding the models' ability to accurately differentiate between legitimate and phishing websites, as well as their robustness. The objective of the analysis is to derive significant interpretations from the data, providing a lucid viewpoint on the merits and drawbacks of each approach in practical situations.

Table 4: Results Analysis

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
RandomForestClassifier	95.4	94.6	96.2	95.4
GradientBoostingClassifier	94.8	93.7	95.9	94.8
AdaBoostClassifier	93.5	92.8	94.2	93.5
VotingClassifier	95.0	94.1	95.9	95.0
DecisionTreeClassifier	91.2	90.5	91.9	91.2
LogisticRegression	89.7	89.0	90.4	89.7
SVC	92.3	91.6	93.0	92.3
KNeighborsClassifier	90.6	89.9	91.3	90.6
GaussianNB	88.4	87.8	88.9	88.4
StackingClassifier	94.2	93.5	94.9	94.2
XGBoost	96.0	95.3	96.7	96.0

Performance Measurement Metrics Analysis:

Accuracy: This metric measures the comprehensive accuracy of a model's data classification. In this regard, XGBoost emerges as the most effective performer, achieving an accuracy rate of 96.0%. This signifies its exceptional capability of accurately distinguishing between phishing and legitimate URLs. We utilize the model as a standard for our analysis, as it outperforms alternative algorithms. Although the Gradient Boosting Classifier achieves a respectable accuracy rate of 94.8%, XGBoost slightly outperforms it in terms of performance.

Precision: The precision metric calculates the ratio of genuine positives to the overall number of predicted positives. XGBoost maintains its lead in our analysis with a precision of 95.3%,

showcasing its efficacy in reducing false positives, a critical factor in phishing detection that safeguards against the misclassification of legitimate websites as phishing. Given its precision of 94.6%, the Random Forest Classifier demonstrates a high level of dependability in correctly identifying fraudulent URLs.

Recall: Recall, also known as sensitivity, quantifies the capacity of the model to identify every pertinent instance. The Random Forest Classifier demonstrates a maximum recall rate of 96.2%, indicating its remarkable aptitude for accurately identifying the majority of phishing instances with infrequent errors. XGBoost, which exhibits a recall rate of 96.7%, is also remarkable for its nearly equivalent capability to detect fraudulent URLs.

F1-Score: The F1-Score is a metric that maintains a balance between precision and recall. With an impressive F1-Score of 96.0%, XGBoost demonstrates an ideal equilibrium between recall and precision. With a score of this, XGBoost is deemed the most equitable model, demonstrating proficiency in both precise fraud URL detection and false positive reduction. Although marginally lower than XGBoost, Random Forest Classifier exhibits a robust equilibrium with an F1-Score of 95.4%.

To summarize, XGBoost establishes itself as the preeminent model in malware URL detection, surpassing all other models in terms of performance across all metrics. Its superior performance in accuracy, precision, recall, and F1-Score emphasizes the solution's overall effectiveness. The Random Forest Classifier, although it lags slightly behind XGBoost, demonstrates strong performance and can be considered a dependable substitute. While models such as Logistic Regression and GaussianNB can be advantageous in specific contexts, their performance is relatively subpar. This underscores the superiority of more sophisticated ensemble methods when it comes to intricate classification tasks like fraud detection.

V. Conclusion and Future Work:

The present study has conducted a comprehensive assessment of numerous machine learning algorithms in an effort to tackle the tough challenge of phishing URL detection. Our results indicate that XGBoost exhibits superior performance in all evaluated metrics (accuracy, precision, recall, and F1-score), setting it apart from the other models examined and solidifying its position as a standard in this domain. Moreover, Random Forest Classifier establishes itself as a formidable competitor, showcasing the effectiveness of ensemble techniques when confronted with intricate classification assignments. Although conventional models such as GaussianNB and Logistic Regression have demonstrated some effectiveness, they are surpassed in performance by more sophisticated ensemble techniques. This research not only illuminates the efficacy of diverse algorithms utilized in cybersecurity software but also emphasizes the criticality of selecting the appropriate instrument for particular cybersecurity obstacles.

There are numerous potential directions for additional research and development in the future. An area of interest is the investigation of deep learning methods, which may provide improved functionalities for phishing URL detection via the implementation of more advanced feature extraction and pattern recognition algorithms. An additional aspect worthy of attention pertains to the integration of these models into operational systems in real-time, with the purpose of proactive

phishing detection. Furthermore, it will be imperative to modify these models in order to identify recently emerged phishing techniques and remain abreast of the swiftly changing cyber threat environment. Increasing the diversity and timeliness of the phishing attack instances in the dataset may also contribute to the improvement of the models' adaptability and precision. Adhering to these avenues may substantially enhance our safeguards against phishing and make a valuable contribution to the development of more resilient cybersecurity solutions.

References:

[1] D. L. Hoffman, T. P. Novak, and A. Venkatesh, "Has the Internet become indispensable?," *Communications of the ACM*, vol. 47, no. 7, pp. 37–42, Jul. 2004, doi: 10.1145/1005817.1005818.

[2] V. Moroz, "How to understand modern digital technologies so as not to fall into the traps of cybercriminals and misinformation online," *Diabetes Obesity Metabolic Syndrome*, no. 6, 2022, doi: 10.57105/2415-7252-2022-6-02.

[3] A. Butnaru, A. Mylonas, and N. Pitropakis, "Towards Lightweight URL-Based Phishing Detection," *Future Internet*, vol. 13, no. 6, p. 154, Jun. 2021, doi: 10.3390/fi13060154.

[4] P. Rani, "PyCaret based URL Detection of Phishing Websites," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 11, no. 1, pp. 908–915, Apr. 2020, doi: 10.17762/turcomat.v11i1.13589.

[5] Y. S Tambe, "Phishing URL Detection Using Machine Learning," *Journal of Advanced Research in Production and Industrial Engineering*, vol. 10, no. 01, pp. 1–5, Sep. 2023, doi: 10.24321/2456.429x.202301.

[6] A. Mallick, S. Dhara, and S. Rath, "Application of machine learning algorithms for prediction of sinter machine productivity," *Machine Learning with Applications*, vol. 6, p. 100186, Dec. 2021, doi: 10.1016/j.mlwa.2021.100186.

[7] S. S. Patil and H. A. Dinesha, "URL Redirection Attack Mitigation in Social Communication Platform using Data Imbalance Aware Machine Learning Algorithm," *Indian Journal of Science and Technology*, vol. 15, no. 11, pp. 481–488, Mar. 2022, doi: 10.17485/ijst/v15i11.1813.

[8] B. Banik and A. Sarma, "Phishing URL detection system based on URL features using SVM," *International Journal of Electronics and Applied Research*, vol. 5, no. 2, pp. 40–55, Dec. 2018, doi: 10.33665/ijear.2018.v05i02.003.

[9] B. E. and T. K., "Phishing URL Detection: A Machine Learning and Web Mining-based Approach," *International Journal of Computer Applications*, vol. 123, no. 13, pp. 46–50, Aug. 2015, doi: 10.5120/ijca2015905665.

[10] "Robust URL Phishing Detection Based on Deep Learning," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 7, Jul. 2020, doi: 10.3837/tiis.2020.07.001.

[11] "Machine Learning Algorithm in Agricultural Machine Vision System," *Machine Learning Theory and Practice*, vol. 1, no. 4, Dec. 2020, doi: 10.38007/ml.2020.010404.

[12] S. C. Anand Sharma, "Comparison study of Machine Learning Algorithm and Data Science based Machine Learning Algorithm Malware Detection," *Mathematical Statistician and Engineering Applications*, vol. 71, no. 3s, pp. 01–07, Jul. 2022, doi: 10.17762/msea.v71i3s.2.

[13] "Classification of Heart Rate Time Series Using Machine Learning Algorithms," *Advances in Machine Learning & Artificial Intelligence*, vol. 2, no. 1, Sep. 2021, doi: 10.33140/amlai.02.01.09.

[14] P. Habib, A. Alsamman, S. Hassanein, and A. Hamwieh, "TarDict: A RandomForestClassifier based software predicts drug-target interaction using SMILES," *Highlights in Bioinformatics*, p. bi202101, Mar. 2021, doi: 10.36462/h.bioinfo.202101.

[15] "Weather prediction using Random Forest Classifier," *Strad Research*, vol. 8, no. 6, Jun. 2021, doi: 10.37896/sr8.6/013.

[16] O. ElSahly and A. Abdelfatah, "An Incident Detection Model Using Random Forest Classifier," *Smart Cities*, vol. 6, no. 4, pp. 1786–1813, Jul. 2023, doi: 10.3390/smartcities6040083.

[17] M. Sopiyan, F. Fauziah, and Y. F. Wijaya, "Fraud Detection Using Random Forest Classifier, Logistic Regression, and Gradient Boosting Classifier Algorithms on Credit Cards," *JUITA: Jurnal Informatika*, vol. 10, no. 1, p. 77, May 2022, doi: 10.30595/juita.v10i1.12050.

[18] N. K. Korada, "Implementation of Naive Bayesian Classifier and Ada-Boost Algorithm Using Maize Expert System," *International Journal of Information Sciences and Techniques*, vol. 2, no. 3, pp. 63–75, May 2012, doi: 10.5121/ijist.2012.2305.

[19] "Predicting Fake Job Posts with a Voting Classifier of Multiple Classification Models," *international journal of food and nutritional sciences*, vol. 11, no. 12, Apr. 2023, doi: 10.48047/ijfans/v11/i12/202.

[20] "SPAM DETECTION FOR SMART HOME DEVICES USING VOTING CLASSIFIER AND ADABOOST," *International Research Journal of Modernization in Engineering Technology and Science*, May 2023, **Published**, doi: 10.56726/irjmets39255.

[21] M. Jena and S. Dehuri, "DecisionTree for Classification and Regression: A State-of-the Art Review," *Informatica*, vol. 44, no. 4, Dec. 2020, doi: 10.31449/inf.v44i4.3023.

[22] T. Fearn, "Support Vector Machines I: The Support Vector Classifier," *NIR news*, vol. 15, no. 5, pp. 14–15, Oct. 2004, doi: 10.1255/nirn.788.

[23] D. Sravanthi and R. D. Jenila, "Comparative Analysis of Hepatitis C Using K-Nearest Neighbor Classifier and Decision Tree Classifier," *CARDIOMETRY*, no. 25, pp. 1010–1016, Feb. 2023, doi: 10.18137/cardiometry.2022.25.10101016.

[24] P. Waqas Khan and Y.-C. Byun, "Multi-Fault Detection and Classification of Wind Turbines Using Stacking Classifier," *Sensors*, vol. 22, no. 18, p. 6955, Sep. 2022, doi: 10.3390/s22186955.

[25] S. G.V. and S. R. E.V., "Uncertain Data Analysis with Regularized XGBoost," *Webology*, vol. 19, no. 1, pp. 3722–3740, Jan. 2022, doi: 10.14704/web/v19i1/web19245.

[26] R. TEKİN and O. YAMAN, "XGBoost Based Intrusion Detection Method for Smart Home Systems," *Journal of Intelligent Systems: Theory and Applications*, vol. 6, no. 2, pp. 152–158, Aug. 2023, doi: 10.38016/jista.1075054.