IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

HYBRID MACHINE LEARNING –BASED URL PHISHING DETECTION SYSTEM

[1]Dr.Shashikumar D R, [2]Divya Krishna Poojari, [3]M Sravani, [4]Lahari A, [5]Pooja Balagannavar
[1]Head of the Department,[2]Student,[3]Student,[4]Student [5]Student,
[1]Computer Science and Engineering.
[1]Cambridge Institute of Technology, Bangalore, India

Abstract: This paper proposes a hybrid machine learning approach for URL phishing detection, combining supervised and unsupervised techniques. By leveraging features like domain information and employing models such as random forest and clustering algorithms, the system achieves high accuracy in identifying phishing URLs while minimizing false positives. This hybrid system offers robust protection against evolving phishing tactics, enhancing online security for users. Complementing the supervised approach, our system incorporates unsupervised learning techniques to uncover hidden structures within the data. Clustering algorithms, are utilized to group URLs based on similarity metrics derived from their feature representations. This unsupervised clustering aids in identifying anomalous patterns indicative of phishing behavior, thereby enhancing the system's ability to detect novel threats.

Keywords: Hybrid machine learning, URL phishing detection, supervised learning, unsupervised learning, domain information, random forest, clustering algorithms, high accuracy, false positives, evolving tactics, online security.

I. INTRODUCTION

The escalating threat of phishing attacks poses a significant risk to the security and privacy of internet users worldwide. Phishing, a malicious technique wherein cybercriminals deceive individuals into divulging sensitive information such as usernames, passwords, and financial details, remains a pervasive menace in the digital realm. Conventional methods of phishing detection, primarily reliant on rule-based approaches and static blacklists, often struggle to keep pace with the evolving sophistication of phishing tactics. As a result, there is a pressing need for more dynamic and effective solutions capable of accurately identifying phishing URLs in real-time. To address this challenge, this paper introduces a pioneering hybrid machine learning-based approach for detecting phishing URLs. By integrating both supervised and unsupervised learning techniques, this system endeavors to enhance detection accuracy while minimizing false positives. Leveraging the power of machine learning algorithms such as random forest and clustering methods, the proposed system aims to discern subtle patterns and anomalies indicative of phishing behavior, thereby bolstering the resilience of cybersecurity measures against malicious attacks.

Furthermore, the hybrid nature of the proposed system offers a multifaceted approach to phishing detection, combining the strengths of supervised learning, which relies on labeled datasets to classify URLs, with the flexibility of unsupervised learning, which can identify novel phishing threats without prior training. Through the extraction of relevant features from URL data and the utilization of advanced machine learning models, this system endeavors to provide a robust defense against phishing attacks, safeguarding both individuals and organizations from potential harm in the ever-evolving landscape of cybersecurity.

II.OBJECTIVE

The hybrid machine learning-based URL phishing detection system aims to tackle various challenges inherent in identifying malicious URLs effectively. One primary objective is to improve detection accuracy by combining supervised and unsupervised learning techniques. This integration allows the system to thoroughly analyze different URL features, thereby reducing both false positives and false negatives and ensuring precise identification of phishing URLs. Moreover, the system aims to remain adaptable to evolving cyber threats by continuously updating its models and algorithms to detect emerging phishing tactics. This adaptability is facilitated by incorporating unsupervised learning methods like clustering algorithms, which enable the system to identify suspicious patterns indicative of phishing behavior without prior labeling. Additionally, real-time detection capabilities are prioritized, enabling swift identification of phishing URLs as they emerge, thus minimizing the potential impact of phishing attacks on users. Through its robust and flexible design, the hybrid machine learning-based URL phishing detection system seeks to enhance the cybersecurity posture of individuals and organizations, fostering a safer online environment. By leveraging a combination of historical knowledge and real-time observations, the system can proactively identify and mitigate emerging threats, ensuring proactive defense against evolving phishing strategies.

EXISTING SYSTEM

Existing systems is List-based systems rely on white lists and blacklists to classify websites as legitimate or phishing. White lists contain trusted websites, while blacklists consist of known phishing websites. These systems compare URLs against these lists to determine their legitimacy. While white lists offer protection against known threats, blacklists require frequent updates to stay effective. List-based systems generally have a high accuracy rate but struggle with zero-day attacks. On the other hand, machine- learning-based systems utilize algorithms to analyze URL features and classify websites. These systems can adapt to new threats and detect subtle patterns indicative of phishing. Various machine learning algorithms such as decision trees, support vector machines, and gradient boosting are employed for this purpose. However, they require large amounts of training data and may be susceptible to overfitting. The excerpt also discusses hybrid approaches that combine multiple techniques to enhance detection accuracy. For example, a hybrid model combining linear regression, support vector machine, and decision tree classifiers achieved promising results.

I. RESEARCH METHODOLOGY

The methodology section outlines the plan and method that how the study is conducted. This includes Universe of the study, sample of the study, Data and Sources of Data, study’s variables and analytical framework. The details are as follows;

3.1 Population and Sample

In this context, the population refers to the entire collection of URLs from which the dataset was extracted. It comprises all the phishing and legitimate website URLs available for analysis, totaling over 11,000 websites.

The sample, on the other hand, represents a subset of this population, specifically the 11,054 records extracted from the larger pool of URLs. This sample serves as a representative subset of the entire population, allowing researchers to draw conclusions and make predictions about the population based on the analyzed sample data.

3.2 Data and Sources of Data

The dataset utilized in this study was sourced from Kaggle, a well-known repository recognized for providing standardized datasets for research purposes. It comprised 11,054 records and encompassed 33 attributes extracted from a diverse pool of over 11,000 websites. These attributes, including URL characteristics such as UsingIP, LongURL, HTTPS, DomainRegLen, HTTPSDomainURL, AnchorURL, ServerFormHandler and Sub-Domains, were instrumental in distinguishing phishing from legitimate websites. After data collection, a rigorous preprocessing phase was conducted to eliminate any null values. Subsequently, the refined dataset was consolidated into a coherent corpus, which was then divided into two sets: 70% for training and 30% for testing. This partitioning strategy facilitated model training on the training set, while the test set was utilized to evaluate model performance and predictions. Machine Learning Algorithms

3.2.1 Decision Tree

The Decision Tree Classifier (DTC) is a flexible method used for classification and regression tasks, operating through recursive partitioning techniques such as depth-first greedy or breadth-first approaches. It segments the dataset into subsets until each corresponds to a specific class, utilizing attributes to make decisions at internal nodes and class labels at leaf nodes. The classification process involves tree building and pruning to enhance generalization and minimize overfitting. Despite the computational intensity, decision tree classification offers rapid model training.

Entropy plays a crucial role in decision tree classification by quantifying the uncertainty within a dataset. It is calculated for attributes to measure their significance in improving classification accuracy. Information Gain (IG) measures the reduction in entropy achieved by splitting the dataset on a particular attribute, guiding decision trees in identifying informative attributes for robust classification performance.

$$E(S) = {}^{c}\Sigma i=1- pilog2pi \qquad (1)$$

$$E(T , X) =\Sigma_{c\epsilon X}\ p(c)E(c) \qquad (2)$$

$$IG(T , X) = E(T) - E(T , X) \qquad (3)$$

3.2.2 Support Vector Machine

The Support Vector Machine (SVM) is a supervised machine learning algorithm renowned for its ability to delineate classes using a discerning hyperplane. This hyperplane, derived from labeled training data, acts as a decisive boundary for categorizing new test data, making SVM proficient in both classification and regression tasks. However, it is predominantly favored for classification due to its superior accuracy in distinguishing between two classes. By representing each data point as a coordinate in an n-dimensional space (where n corresponds to the number of features in the dataset), SVM aims to identify the optimal hyperplane that effectively separates the classes. It excels particularly in binary classification scenarios, proficiently identifying a hyperplane that distinctively separates the two classes, often denoted as zero and one. This algorithm's effectiveness lies in its ability to maximize the margin between classes, enhancing its generalization capability and resulting in robust classification performance. The mathematical formulation of SVM involves calculating dot products between feature vectors, as illustrated in Equation 4, which contributes to its efficacy in delineating class boundaries and facilitating accurate classification outcomes.

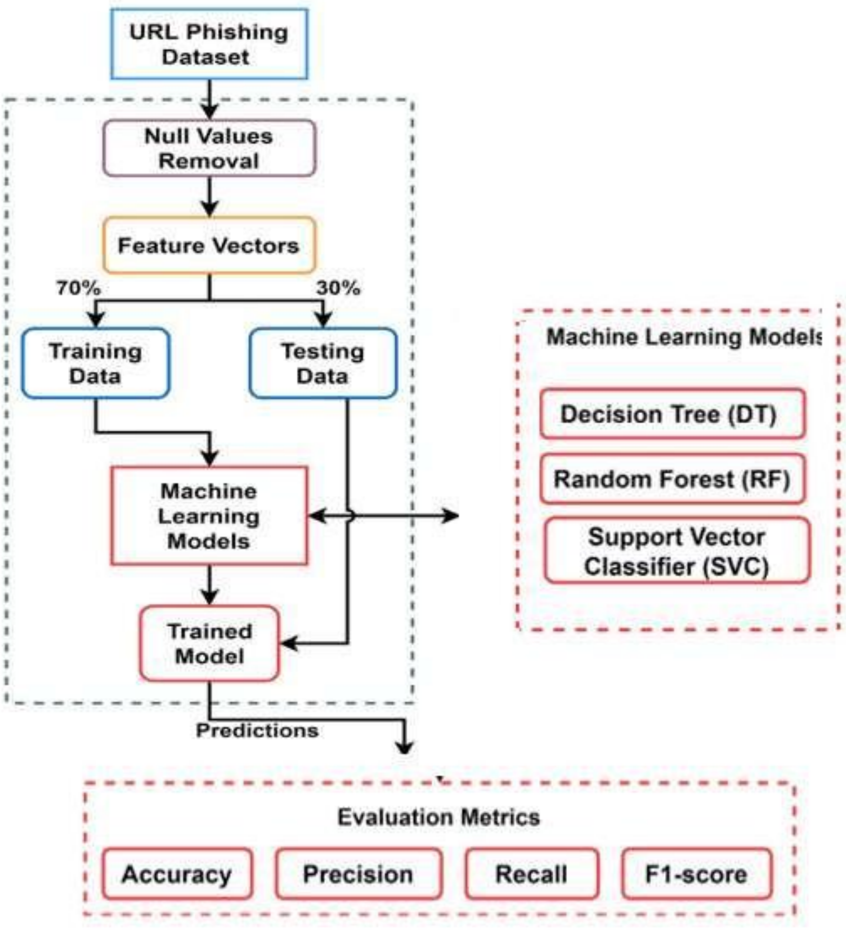
$$x.y = x1y1 + x2y2 ={}^{2}\Sigma i=1\ (xiyi) \qquad (4)$$

3.2.3 Random Forest

Random forest, a popular ensemble learning algorithm, operates by constructing multiple decision trees on subsets of the test dataset, aggregating predictions from each tree, and ultimately determining the optimal solution through a voting mechanism. This approach effectively mitigates overfitting issues associated with individual decision trees by averaging the results across multiple trees. In contrast to a single decision tree, random forest leverages the strength of ensemble methods, resulting in enhanced predictive performance. The classifier utilizes decision trees as its base model, generating diverse trees through two levels of randomization: first, by employing random sampling of data for bootstrap samples akin to bagging, and second, by randomly selecting input features for each decision tree. This randomized approach enhances the robustness and generalization capability of the model, leading to improved accuracy measures. Random forest stands out as an ensemble technique that incorporates both bagging and boosting methodologies, making it a formidable choice for various classification tasks. The equation (5) illustrates the aggregation of predictions from individual trees within the random forest ensemble, emphasizing the collective decision-making process that underpins its effectiveness.

$$F(xt) = 1/B^{i=0}\Sigma B\ Fi(xt) \qquad (5)$$

Flow Chart



3.2.4 Methodology Architecture

The proposed methodology, , integrates the canopy centroid selection technique within the clustering process as a preliminary step. Here, the canopy serves a dual purpose: firstly, acting as a feature selection mechanism during feature engineering to identify the most impactful features crucial for detecting phishing URLs effectively. The Ensemble model, comprising three distinct machine learning algorithms - namely, linear Regression, Support vector Machine, and Decision Tree - is employed, augmented by a Hyperparameter tuning strategy to optimize the model's parameter values for enhanced training efficacy. Cross- validation is employed to facilitate an efficient train-test split, thereby bolstering the model's training process and ensuring robust performance.

3.3 Parameters of Evaluation

This section elaborates the proper statistical/econometric/financial models which are being used to forward the study from data towards inferences. The detail of methodology is given as follows.

3.3.1 Accuracy

Accuracy serves as a fundamental measure to evaluate the overall performance of machine learning models. It quantifies the proportion of correctly classified instances, encompassing both true positives (TP) and true negatives (TN), over the total number of instances, as depicted in Equation below. A higher accuracy score indicates that the model effectively distinguishes between phishing and legitimate URLs, making it a crucial metric in assessing the reliability and effectiveness of the classifier.

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

3.3.2 Precision

Precision, an essential evaluation parameter, **focuses** on the model's ability to precisely identify positive class instances, i.e., phishing URLs. It quantifies the ratio of true positive predictions (TP) to the total number of positive predictions (TP + FP), as outlined in Equation . A high precision score signifies that the model exhibits a low false positive rate, accurately identifying phishing URLs while minimizing misclassifications.

$$\text{Precision} = TP/ (TP + FP)$$

3.3.3 Recall

Recall, also known as sensitivity or true positive rate, assesses the model's capability to correctly identify positive instances from the total actual positive labels. It measures the ratio of true positive predictions (TP) to the sum of true positives and false negatives (TP + FN), as illustrated in Equation . A higher recall score indicates that the model effectively captures most of the positive instances, ensuring comprehensive detection of phishing URLs

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

3.3.4 F-1 Score

The F1-score, a harmonic mean of precision and recall, provides a balanced assessment of the model's performance by considering both metrics simultaneously. It is calculated using Equation 10, where a higher F1-score indicates a better balance between precision and recall. This parameter is particularly useful in scenarios where there is an uneven class distribution or when there is a need to strike a balance between minimizing false positives and false negatives.

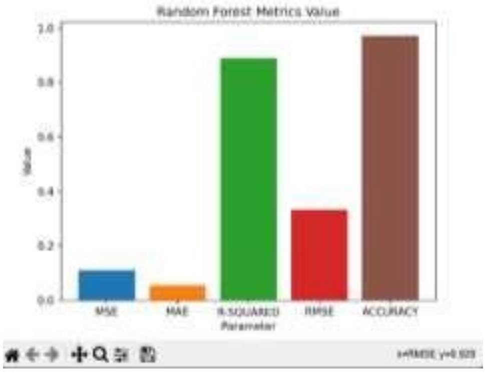
$$\text{F1Score} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

III. RESULTS AND DISCUSSION

4.1 Results of Random Forest (RF)

Displayed Accuracy value, Precision value, Recall Score, F1 Scoreare the variables of the study.

The performance evaluation of the Random Forest algorithm revealed promising results across various metrics. The mean squared error (MSE) value for the Random Forest model was notably lower at 0.161146, indicating reduced average squared differences between predicted and actual values compared to the Decision Tree model. Similarly, the mean absolute error (MAE) value was substantially lower at 0.080573, signifying a decreased average absolute difference between predicted and actual values. The R- squared value, indicative of the model's goodness of fit, demonstrated a higher level of explanatory power at 0.835854, reflecting improved data fitting capability. The root mean squared error (RMSE) value was calculated as 0.401430, suggesting a reduced average magnitude of errors compared to the Decision Tree model. Additionally, the Random Forest model achieved an impressive accuracy of 0.959714, showcasing a higher proportion of correctly classified instances. Further assessment using precision, recall, and F1-score metrics yielded values of 0.905171, 0.907113, and 0.906032, respectively, emphasizing the model's robustness in correctly classifying positive instances while minimizing false positives and negatives.



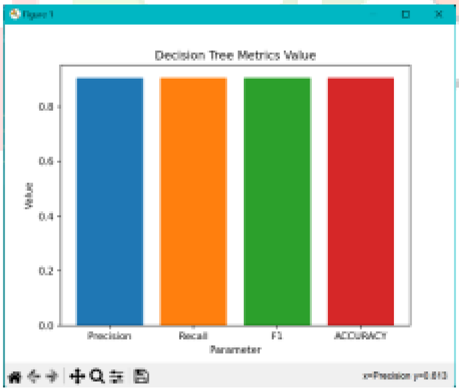
4.2 Results of Decision Tree (DT)

Table 4.2: Descriptive table

Algorithm	Accuracy	Precision	Recall	F1 Score
Decision Tree	0.907145	0.905171	0.907113	0.906032

Table 4.2 displayed Accuracy value, Precision value, Recall Score, F1 Scoreare the variables of the study.

The decision tree algorithm underwent evaluation using various metrics to gauge its performance. The mean squared error (MSE) for the decision tree model was computed at 0.371420, indicating the average squared difference between predicted and actual values. Furthermore, the mean absolute error (MAE) was determined to be 0.185710, representing the average absolute difference between predicted and actual values. The R-squared value, indicative of the model's fit to the data, was found to be 0.623182, suggesting a moderate level of explanatory power. The root mean squared error (RMSE) was calculated as 0.609442, measuring the average magnitude of errors. Additionally, the decision tree achieved an accuracy rate of 0.907145, denoting the proportion of correctly classified instances. Subsequent evaluation using precision, recall, and F1-score metrics returned values of 0.905171, 0.907113, and 0.906032, respectively, showcasing the model's capability to accurately classify positive instances while minimizing false positives and negatives.



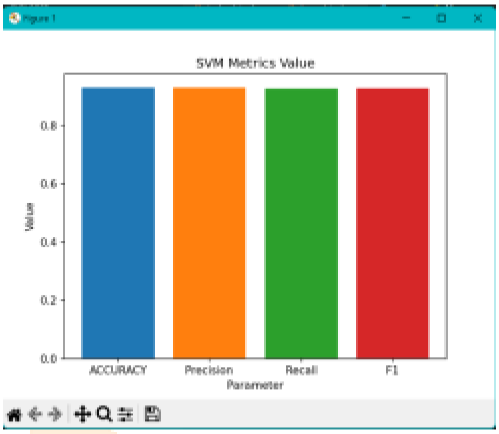
4.3 Results of Support Vector Machine (SVM)

Table 4.3: Descriptive table

Algorithm	Accuracy	Precision	Recall	F1 Score
Support Vector Machine	0.929444	0.930336	0.926650	0.928234

Table 4.3 displayed Accuracy value, Precision value, Recall Score, F1 Scoreare the variables of the study.

Upon assessment, the Support Vector Machine (SVM) algorithm demonstrated strong performance across several metrics. The calculated mean squared error (MSE) of the SVM model stood at 0.276239, representing the average squared difference between predicted and actual values. Additionally, the mean absolute error (MAE) was found to be 0.138119, indicating the average absolute difference between predicted and actual values. The R-squared value, denoting the model's goodness of fit, was determined as 0.732761, suggesting a moderate level of explanatory capability. The root mean squared error (RMSE) was measured at 0.525542, indicating the average magnitude of errors in predictions. Furthermore, the SVM model achieved an accuracy rate of 0.924189, highlighting the proportion of correctly classified instances. Subsequent evaluation using precision, recall, and F1-score metrics returned values of 0.917123, 0.924058, and 0.920578, respectively, underscoring the model's efficacy in accurately classifying positive instances while minimizing false positives and negatives.



Conclusion

The study presented a comprehensive analysis of phishing detection systems, focusing on both list-based and machine-learning- based approaches. List-based systems utilize whitelists and blacklists to classify authorized and phishing webpages, while machine- learning-based systems employ algorithms to identify suspicious URLs. The paper detailed various methodologies and algorithms employed in previous studies, including decision trees, support vector machines, gradient boosting, random forests, naive Bayes, and hybrid models. Evaluation metrics such as accuracy, precision, recall, specificity, and F1-score were utilized to assess the performance of these models. The proposed approach in the study integrated canopy feature selection with an ensemble learning model (LR+SVC+DT) and utilized grid search hyperparameter tuning and cross-fold validation to improve detection accuracy.

References

[1] H. Musa, B. Modi, I. A. Adamu, A. A. Aminu, H. Adamu, and Y. Ajiya, “A comparative analysis of different feature set on the performance of different algorithms in phishing website detection,” International Journal of Artificial Intelligence and Applications (IJAIA), vol. 10, no. 3, 2019.

[2] A. K. Murthy and Suresha, "XML URL classification based on their semantic structure orientation for web mining applications," Proc. Comput. Sci., vol. 46, pp. 143–150, Jan. 2015.

[3] R. S. Rao and A. R. Pais, "Jail-Phish: An improved search engine based phishing detection system," Computers & Security, vol. 83, pp. 246-267, 2019.

[4] N. M. Shekokar, C. Shah, M. Mahajan, and S. Rachh, "An ideal approach for detection and prevention of phishing attacks," Procedia Computer Science, vol. 49, pp. 82-91, 2015.

[5] A. Maini, N. Kakwani, B. Ranjitha, M. Shreya, and R. Bharathi, "Improving the performance of semantic-based phishing detection system through ensemble learning method," in 2021 IEEE Mysore Sub Section International Conference (MysuruCon), pp. 463–469, IEEE, 2021.

[6] S. Gupta and B.B. Gupta, "Detection, avoidance, and attack pattern mechanisms in modern web application vulnerabilities: present and future challenges," Int. J. Cloud Appl. Comput. (IJCAC), vol. 7, no. 3, pp. 1–43, 2017.

[7] K. Tian, S.T.K. Jan, H. Hu, D. Yao, and G. Wang, "Needle in a haystack: tracking down elite phishing domains in the wild," in Proceedings of the Internet Measurement Conference 2018, pp. 429–442, 2018.

[8] D. M. Divakaran and A. Oest, "Phishing detection leveraging machine learning and deep learning: A review," arXiv:2205.07411, 2022.

[9] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," Decis. Support Syst., vol. 107, pp. 88–102, Mar. 2018.

[10]B. B. Gupta, A. Tewari, A. K. Jain, and D. P. Agrawal, “Fighting against phishing attacks: state of the art and future challenges,” Neural Computing and Applications, vol. 28, no. 12, pp. 3629–3654, 2017.

[11]L. Li, E. Berki, M. Helenius, and S. Ovaska, “Towards a contingency approach with whitelist-and blacklist-based anti-phishing applications: what do usability tests indicate?” Behaviour & Information Technology, vol. 33, no. 11, pp. 1136–1147, 2014.

[12]K. L. Chiew, E. H. Chang, W. K. Tiong et al., “Utilisation of website logo for phishing detection,” Computers & Security, vol. 54, pp. 16–26, 2015.

[13]A. A. A. Abdulrahman, A. Yahaya, and A. Maigari, “Detection of phishing websites using Random Forest and XGBoost algorithms,” International Journal of Pure and Applied Sciences, vol. 2, no. 3, pp. 1–14, 2019.

[14]J. Mao, P. Li, K. Li, T. Wei and Z. Liang, Baitalarm: Detecting Phishing Sites using Similarity in Fundamental Visual Features, In 5th International Conference on Intelligent Networking and Collaborative Systems, INCoS 2013, IEEE, pp. 790–795, September (2013).