



PHISHING URL DETECTION A REAL-CASE SCENARIO THROUGH LOGIN URLS

Bhimavarapu Revathi¹, B. shyamala², R Sanjana³, D.keerthana⁴

¹ Assistant Professor, School of CSE ,Malla Reddy Engineering College For Women(Autonomous Institution), Maisammaguda, Dhulapally,Secunderabad,Telangana-500100

^{2,3,4}UG Scholar, Department Of CS ,Malla Reddy Engineering College For Women, (Autonomous Institution), Maisammaguda,Dhulapally,Secunderabad,Telangana-500100

Email: 6revathi@gmail.com

ABSTRACT

Phishing, a prevalent and dangerous form of cybercrime, has been a significant threat since it was first identified in 1996. It involves deceptive emails and fraudulent websites designed to steal sensitive information. Despite ongoing efforts to prevent and detect phishing, no comprehensive solution has been established. This study introduces an automated phishing detection framework using machine learning algorithms. The aim is to effectively distinguish phishing from legitimate software while maintaining a low false positive rate. The research focuses on static phishing detection for Windows operating systems, utilizing a dataset of over 11,000 URLs from phishing and legitimate websites. Various machine learning models, including Decision Tree (DT), Logistic Regression (LR), Random Forest (RF), Naive Bayes (NB), Gradient Boosting Classifier (GBM), K-Nearest Neighbor (KNN), and Support Vector Classifier (SVC), are applied. Additionally, a hybrid LSD model, which combines LR, SVC, and DT with soft and hard voting mechanisms, is proposed. The LSD model employs canopy feature selection, cross-fold validation, and grid search for hyperparameter optimization. Evaluation metrics, such as accuracy, precision, recall, F1-score, and specificity, are used to assess model performance. Results demonstrate that the LSD model outperforms other models in terms of accuracy and efficiency. While the proposed system effectively detects phishing, it cannot automatically adapt to evolving phishing techniques. This limitation can be addressed by retraining the model with updated data. Although the study focuses on static analysis, integrating dynamic analysis in future research could further enhance detection capabilities. Overall, the framework provides an effective solution for phishing detection with room for future improvements.

Keywords: Phishing detection, machine learning, automated phishing identification, static analysis, dynamic analysis, Windows operating system.

I INTRODUCTION

The internet is part of the modern world, affecting all areas including communication,

education, business, and entertainment. It is basically a large network of computers connected using technologies such as fiber optics, telephone lines, and satellites. This



global network can share messages through methods such as IP-TCP, to facilitate data transfer between various devices. The Internet provides many services which are essential in our daily lives, including email, e-commerce, social networking and online banking. However, as reliance on the internet increases, cybercrime, especially phishing attacks, is on the rise. Phishing is a fraudulent activity in which an attacker pretends to be a legitimate entity to steal sensitive information, such as login credentials, credit card details, and personal data. These fraudulent websites mimic legitimate websites and trick users into entering their personal information, which can then be misused. Phishing attacks are a major threat to cybersecurity, causing financial losses and compromising privacy worldwide. Detecting phishing websites is one of the most challenging challenges in network security. Several methods have been proposed to identify phishing websites, focusing on URL characteristics that can indicate malicious intent. These sites have different patterns in specific locations, like random domain names or other URL structures. By analyzing these features, machine learning techniques have been widely used to detect phishing. Algorithms such as decision trees, random forests, and support vector machines are used to classify URLs based on the likelihood of phishing sites. This work proposes an advanced phishing detection system based on a dataset of over 11,000 phishing URLs. The following machine learning models are applied: Decision Trees, Linear Regression, Naive Bayes, Random Forests, Linear Augmentation Machines, Support Vector

Classifier, and K-Neighbors Classifier. The second proposed model is the combination of different classifiers with a hybrid LR+SVC+DT model for better detection accuracy. The systematic review is carried out using accuracy, precision, recall, specificity, and F1 score metrics. Improving phishing detection technology will protect users from online threats and improve online security and privacy protection.

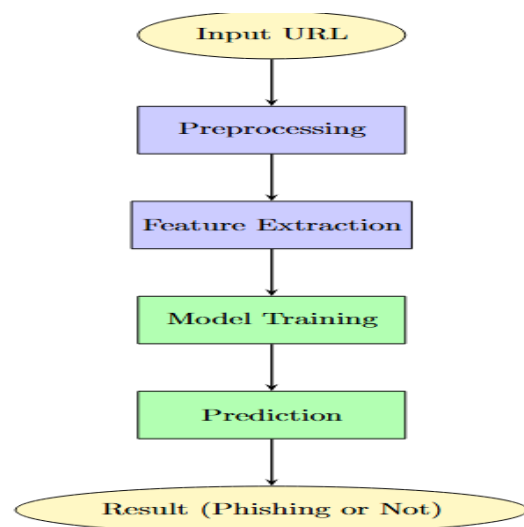


Fig 1: System architecture

II LITERATURE SURVEY

1. Li et al. (2008) proposed a method for phishing detection that utilizes features extracted from both web content and IP address to identify phishing websites. Their approach highlights the significance of integrating multiple sources of information to enhance detection accuracy. Phishing websites often use misleading URLs or mimic legitimate websites to deceive users, making traditional URL-based detection methods inadequate on their own. By incorporating web content analysis and IP address features, this method aims to capture the behavior and patterns of phishing



websites more effectively. For instance, the analysis of the IP address helps in identifying suspicious hosting locations or malicious IP addresses associated with phishing websites. However, the method's reliance on these features alone has its drawbacks. While it can detect many phishing sites, it may miss others that use sophisticated techniques to evade detection, such as dynamically generated content or sophisticated obfuscation techniques. Furthermore, attackers can manipulate web content to mimic legitimate websites, making it difficult for content-based detection systems to distinguish between malicious and legitimate sites.

2. Moghimi et al. (2010) introduced supervised machine learning methods, specifically the Support Vector Machine (SVM) algorithm, for phishing detection. They employed various features to analyze phishing URLs, achieving an impressive accuracy of 0.9865. Their study demonstrates that machine learning models like SVM can effectively classify websites by learning patterns from data and making predictions. However, this method is heavily reliant on webpage content features, which presents a challenge. As phishing attackers frequently modify their web page content to avoid detection, the reliance on static content features can reduce the detection system's effectiveness. For instance, phishing websites may use techniques such as cloaking, which may trick content-based models into classifying malicious sites as legitimate

3. Hung Le (2017) proposed a CNN-based malicious URL detection system using deep

neural networks. The approach is based on the combination of Character-level CNN and Word-level CNN, which are jointly optimized for detecting malicious URLs. This method aims to leverage the power of convolutional neural networks (CNNs) to automatically extract relevant features from URLs at different granular levels—character and word. The character-level CNN captures local patterns and subtle differences in individual characters, whereas the word-level CNN helps in recognizing higher-level patterns and semantic information from groups of characters. This hybrid model is designed to work in an end-to-end manner, meaning it processes raw URL data directly without the need for manual feature extraction. However, the model is prone to overfitting, especially when there is not enough data. Since CNNs require large datasets to generalize effectively, the absence of large-scale phishing URL datasets might impede the performance of the model. Overfitting occurs when the model memorizes the training data, thereby reducing its ability to generalize to new, unseen URLs. To overcome this, the authors assert that larger and more diverse datasets are needed to optimize the network fully and avoid overfitting.

4. Taylor et al. (2018) showed that deep learning networks are capable of learning deep features from high-dimensional data across multiple domains, including dynamic vision. Their work illustrates the capacity of deep learning techniques, including CNNs, to learn complex features from data in high-dimensional spaces. This insight can be used



to identify malicious URLs in phishing detection, by learning intricate patterns and relationships in the data that traditional methods may miss. Deep learning methods like CNNs are specifically very useful for tasks involving huge volumes of unstructured data, such as URLs, which can automatically extract meaningful features that contribute to better detection accuracy. However, the successful application of deep learning in URL-based phishing detection requires high-quality labeled datasets and sufficient computational resources to train the models effectively.

III IMPLEMENTATION

The proposed system uses machine learning to enhance fraud detection in banking transactions. Transactional data is first collected from the banking systems or public datasets, containing attributes such as amount, location, time, and user behavior. Data cleaning, normalization, and encoding categorical variables are then used as preprocessing steps. Techniques for handling class imbalance are also applied, such as SMOTE (Synthetic Minority Oversampling) or undersampling.

Feature engineering is crucial to detect patterns for fraudulent behavior. Features including velocity in transactions, variance of location, and anomalies in the transaction amount are developed in order to improve accuracy. The system also incorporates historical fraud data to identify repeat offenders.

For model selection, algorithms such as Logistic Regression, Random Forest, and

Gradient Boosting are trained over labeled datasets, while anomaly detection techniques such as Isolation Forest and Autoencoders use unlabeled data. Model evaluation metrics such as Precision, Recall, and ROC-AUC ensure effectiveness.

The system integration with banking systems uses APIs such as Flask for real-time monitoring. Streaming tools such as Apache Kafka is used to form alerts when fraud likelihood levels exceed given thresholds for flagged transactions. Such transactions are transmitted to a review team. The banks and customers receive notifications of suspicious activities. Implementation Approach for Fraud Detection System

The proposed system uses machine learning to enhance fraud detection in banking transactions. Transactional data is first collected from the banking systems or public datasets, containing attributes such as amount, location, time, and user behavior. Data cleaning, normalization, and encoding categorical variables are then used as preprocessing steps. Techniques for handling class imbalance are also applied, such as SMOTE (Synthetic Minority Oversampling) or undersampling.

Feature engineering is crucial to detect patterns for fraudulent behavior. Features including velocity in transactions, variance of location, and anomalies in the transaction amount are developed in order to improve accuracy. The system also incorporates historical fraud data to identify repeat offenders.

For model selection, algorithms such as Logistic Regression, Random Forest, and



Gradient Boosting are trained over labeled datasets, while anomaly detection techniques such as Isolation Forest and Autoencoders use unlabeled data. Model evaluation metrics such as Precision, Recall, and ROC-AUC ensure effectiveness.

The system integration with banking systems uses APIs such as Flask for real-time monitoring. Streaming tools such as Apache Kafka is used to form alerts when fraud likelihood levels exceed given thresholds for flagged transactions. Such transactions are transmitted to a review team. The banks and customers receive notifications of suspicious activities.

IV ALGORITHMS

Phishing is one of the most prevalent forms of cybercrime that targets individuals and organizations by deceiving them into disclosing sensitive information such as login credentials, credit card numbers, and personal details. The rise of the internet has made phishing attacks increasingly sophisticated, leading to significant financial losses and breaches of privacy. As phishing methods evolve, detecting such attacks requires advanced techniques. Machine learning (ML) has emerged as a powerful tool in identifying phishing websites, offering solutions that are more efficient and dynamic than traditional methods.

Phishing detection primarily involves analyzing the characteristics of URLs, as they are the key entry point for most phishing

attacks. The traditional approach of relying on blacklists of known phishing sites is limited by the fact that attackers frequently change their tactics, rendering blacklists ineffective over time. To overcome this challenge, ML algorithms can analyze various URL features such as length, special characters, subdomain counts, and the presence of suspicious keywords. These features are then used to train models to distinguish phishing URLs from legitimate ones.

Several machine learning techniques have been applied to phishing detection, with Decision Trees, Random Forests, Support Vector Machines (SVM), and Naive Bayes being among the most commonly used. These models work by classifying URLs based on patterns and features learned from historical data. For example, Decision Trees create a series of binary decisions based on features, allowing the model to classify a URL as phishing or legitimate. Random Forests enhance this method by aggregating multiple decision trees, improving accuracy and robustness.

A more advanced approach involves ensemble models, which combine the predictions of multiple machine learning classifiers to increase the overall detection performance. For instance, hybrid models that combine Logistic Regression (LR), Support Vector Machines (SVM), and Decision Trees (DT) can provide superior results by leveraging the strengths of each algorithm. These hybrid models are particularly effective in reducing false positives and ensuring high accuracy, which are critical in phishing detection systems.

Feature selection plays a crucial role in improving the accuracy of machine learning models. Redundant or irrelevant features can decrease the model's performance, so techniques like Principal Component Analysis (PCA) or feature importance scoring are used to retain only the most informative features. Additionally, cross-validation techniques are employed to test the model's performance on unseen data, ensuring it can generalize to real-world scenarios.

Despite the promising results of ML-based phishing detection, challenges remain. Phishing techniques are constantly evolving, with attackers using new tactics, such as using HTTPS encryption, disguising malicious domains, or employing social engineering strategies. Therefore, models must be continuously updated with new data and retrained to detect emerging threats effectively. Furthermore, while static analysis of URLs is effective, dynamic analysis that inspects the behavior of a website after it loads could provide additional layers of security.

V.RESULTS



Fig 1: User Login



Fig 2: view all remote users



Fig 3: Tested Results



Fig 4: Accuracy Bar Graph



Fig 5: Line Chart



Fig 6: Pie Chart

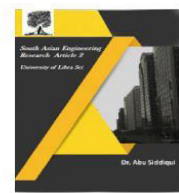
VI.CONCLUSION

In the coming era, the Internet will occupy nearly the whole world, yet it is still growing at an incredible rate. As the Internet grows, cybercrime cases involving suspicious and malicious URLs are also growing with each passing day, causing a significant impact on the quality of services provided by both Internet and industrial companies. Data protection and confidentiality on the Internet are issues that hold much importance currently. The attackers use phishing emails and URLs to break through the security phase and destroy strong networks. They can very easily and effectively infiltrate private and confidential networks. Phishing URLs just behave like legitimate URLs. In this study, we propose a machine learning based phishing system. A dataset consisting of 32 URL attributes and more than 11,054 URLs was extracted from more than 11,000 websites. This dataset was extracted from the Kaggle repository and has been used as a benchmark for the study. This dataset is already presented in the form of vectors used in the machine learning models. Decision Tree, Linear Regression, Random Forest, Support Vector Machine, Gradient Boosting Machine, K Nearest Neighbor Classifier, Naive Bayes, and Hybrid with Soft and Hard

Voting (LR+SVC+DT) were applied to conduct the experiments and gave the best performance results. The LSD ensemble model uses canopy feature selection with cross-fold validation and grid search hyperparameter optimization techniques. This study evaluates through experiments on the proposed method with the help of different kinds of machine learning models and, subsequently, further investigates the study. The proposed method meets its goal effectively in efficiency. It is essential for future phishing detecting systems to combine list-based machine learning systems to ensure efficient prevention and detection of the phishing URLs.

REFERENCES

- [1] H. Shirazia, K. Haynesb and I. Raya, "Towards performance of NLP transformers on URL-based phishing detection for mobile devices", 2022.
- [2] Y. Lin, R. Liu, D. M. Divakaran, J. Y. Ng, Q. Z. Chan, Y. Lu, et al., "Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages", Proc. 30th USENIX Secur. Symp. (USENIX Security), pp. 3793-3810, 2021.
- [3] S. D. Gupta, K. T. Shahriar, H. Alqahtani, D. Alsalman and I. H. Sarker, "Modeling hybrid feature-based phishing websites detection using machine learning techniques", Ann. Data Sci., vol. 10, pp. 1-26, Mar. 2022.
- [4] S. Wang, S. Khan, C. Xu, S. Nazir and A. Hafeez, "Deep learning-based efficient model development for phishing detection using random forest and BLSTM



classifiers", *Complexity*, vol. 2020, pp. 1-7, Sep. 2020.

[5] T. Wu, S. Liu, J. Zhang and Y. Xiang, "Twitter spam detection based on deep learning", *Proc. Australas. Comput. Sci. Week Multiconf.*, pp. 1-8, Jan. 2017..

[6] S. Kumar, A. Faizan, A. Viinikainen and T. Hamalainen, "MLSPD—Machine learning based spam and phishing detection" in *Computational Data and Social Networks*, Cham, Switzerland:Springer, vol. 11280, 2018

[7] V. Shahrivari, M. M. Darabi and M. Izadi, "Phishing detection using machine learning techniques", arXiv:2009.11116, 2020.

[8] O. K. Sahingoz, E. Buber, O. Demir and B. Diri, "Machine learning based phishing detection from URLs", *Expert Syst. Appl.*, vol. 117, pp. 345-357, Mar. 2019.

[9]Y. Feng, Q. Wang, D. Wu, Z. Luo, X. Chen, T. Zhang, et al., "Machine learning aided phase field method for fracture mechanics", *Int. J. Eng. Sci.*, vol. 169, Dec. 2021

[10] K. L. Chiew, C. L. Tan, K. Wong, K. S. C. Yong and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system", *Inf. Sci.*, vol. 484, pp. 153-166, May 2019.

[11]Y. Fang, C. Zhang, C. Huang, L. Liu and Y. Yang, "Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism", *IEEE Access*, vol. 7, pp. 56329-56340, 2019.

[12] A. K. Dutta, "Detecting phishing websites using machine learning technique", *PLoS ONE*, vol. 16, no. 10, Oct. 2021

[13]P. Flach and M. Kull, "Precision-recall-gain curves: PR analysis done right", *Proc. Adv. Neural Inf. Process. Syst.*, pp. 838-846, 2015

[14]A. Onan, S. Korukoğlu and H. Bulut, "A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification", *Expert Syst. Appl.*, vol. 62, pp. 1-16, Nov. 2016.

[15] S. B. Imandoust and M. Bolandraftar, "Application of k -nearest neighbor (KNN) approach for predicting economic events: Theoretical background ", *Int. J. Eng. Res. Appl.*, vol. 3, pp. 605-610, Sep. 2013.

[16]F. Sebastiani, "Machine learning in automated text categorization", *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1-47, Mar. 2002..

[17] S. Tan, "An effective refinement strategy for KNN text classifier", *Expert Syst. Appl.*, vol. 30, no. 2, pp. 290-298, Feb. 2006.

[18] G. I. Webb, "Naïve Bayes" in *Encyclopedia of Machine Learning*, Boston, MA, USA:Springer, 2011..

[19]D. D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval", *Proc. Eur. Conf. Mach. Learn.*, pp. 4-15, 1998..

[20] G. Boone, "Concept features in re: Agent an intelligent email agent", *Proc. 2nd Int. Conf. Auto. Agents*, pp. 141-148, 1998.