International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 1, Jan-Feb 2025
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Detection of Phishing Website Using ML

Sanjana K M, Dhanalakshmi S, Ranjitha K R , Chaithra U ,Mrs. Divya Tyagi
Information Science and Engineering, AMC College of Engineering, Bangalore, Karnataka, India

----------**********----------

Abstract:

Phishing websites pose a severe threat to cyber security, tricking users into divulging sensitive information. This research introduces the framework for machine learning, to accurately detect phishing websites. The proposed system utilizes the Random Forest algorithm, and NLP tools such as Word Cloud and NLTK for feature extraction and visualization. Python libraries like SK learn and pandas are used for prepossessing and model evaluation. The results indicate that Logistic Regression achieves an accuracy of around 90%, making it a robust solution for real-world phishing detection.

Keywords— **Accuracy, NLTK, Pandas, SK Learn**

----------**********----------

I. Introduction

Individuals, businesses, and even nations are falling victim to phishing, making it the biggest issue today. In recent years, the Web's growth has been expedited by the availability of several services, including electronic banking, educational software installing, and social networking. Consequently, there is a continual influx of enormous data being downloaded and sent to the Internet. Social engineering involves sending phishing emails that seem to originate from reputable companies or organizations in order to lure individuals into accessing harmful websites and providing sensitive information (such as login credentials). Installing spyware on computers is one technical approach used to steal credentials directly. Systems are frequently utilized to intercept users' login credentials for online accounts.

The proliferation of online services—such as electronic banking, e-commerce, social networking, and remote education—has exponentially increased the volume of data exchanged over the internet. This surge in online activity presents an attractive target for cybercriminals employing sophisticated phishing techniques. Traditional phishing strategies include the use of spoofed emails, malicious URLs, and the installation of spyware to harvest credentials directly from users. These methods frequently exploit human weaknesses instead of technical flaws, making them both pervasive and difficult to counter.

Some challenges and limitations encountered during the process of the project development include:

Data Collection: Collecting a sufficient volume of high-quality, labelled data for phishing and legitimate websites is crucial for training machine learning models. Datasets may often be incomplete, unbalanced, or outdated.

Real time Detection Constraints: Phishing detection systems need to operate in real-time to stop users from visiting malicious websites

Scalability and Resource Limitations: Phishing detection systems must process a large volume of URLs in real-time, which demands significant computational resources. High traffic environments, such as enterprise firewalls or email gateways, require scalable solutions.

Machine learning models, especially those involving ensemble methods Random Forest and Gradient Boosting, demand significant memory and processing power.

Balancing False Positives and False Negatives:Striking an optimal trade-off between false positives (flagging legitimate URLs) and false negatives (missing phishing URLs).
High false positive rates diminish user trust in the

detection system, leading to potential bypassing of alerts. False negatives pose a significant security risk, exposing users to phishing attacks.
Models tuned for high recall may sacrifice precision, increasing false positives, and vice versa.

Integration with Existing Systems:
Integrating the detection model with existing platforms, such as browsers, email filters, or enterprise security systems. Compatibility issues with different system architectures or protocols.

II. Literature Survey

The paper "Detection of Phishing Website using Machine Learning" by Mrs. Keerthana Shankar, U. Rithika, Sathya M., Vaishnavi Chitapur, and Tejaswini J. explores the challenge of detecting phishing websites, which manipulate users into exposing sensitive information like login credentials and financial data. To address the limitations of traditional methods like blacklisting, the authors propose a machine learning-based approach using the Gradient Boosting Classifier (GBC). This model derives features from website URLs, content, and visual elements to distinguish phishing websites from authentic ones. The approach involves collecting data from public phishing databases, preprocessing for consistency, and feature engineering to identify unique traits of phishing sites. The system, implemented using Python and tools like Scikit-learn, achieved an accuracy of 87.82%, highlighting its effectiveness. The authors emphasize the adaptability of their approach to evolving phishing techniques, demonstrating the potential for scalable and robust online security solutions.

The paper titled "Improving Auto-Detection of Phishing Websites using Fresh-Phish Framework", authored by Hossein Shirazi, Kyle Haefner, and Indrakshi Ray from Colorado State University, presents an enhanced machine learning framework, Fresh-Phish, to address the growing complexity of phishing attacks. Phishing, a malicious tactic to steal sensitive information by impersonating legitimate entities, has outpaced traditional detection methods like blacklisting. The Fresh-Phish framework is an open-source Python-based solution designed to create up-to-date datasets for training machine learning classifiers. It features two core modules: an Evaluation Module that measures 28 characteristics of websites, such as URL length and DNS anomalies, and an Experiment Module that evaluates classifier performance using the dataset. With data sourced from 2,500 legitimate websites (Alexa) and 2,500 phishing sites (Phish Tank), the study highlights the framework's ability to enhance detection accuracy through non-binary feature modeling and advanced classifiers like SVMs and TensorFlow-based neural networks.

The paper titled " "A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques," authored by Said Salloum, Tarek Gaber, Sunil Vadera, and Khaled Shaalan, explores the application of Natural Language Processing techniques for tracing phishing emails,provides a detailed review of 100 studies published between 2006 and 2022, focusing on the use of NLP techniques for detecting phishing emails. The authors explore key research areas, including feature extraction and selection, classification methods, and optimization techniques. The review underscores the prevalence of supervised machine learning algorithms, including Support Vector Machines (SVMs) being the most commonly used, followed by Naïve Bayes and Decision Trees. Popular NLP approaches such as TF-IDF, word embeddings, and latent semantic analysis (LSA), are commonly used to detect phishing patterns. The analysis also reveals a reliance on datasets like the Nazario phishing corpus and Spam Assassin Public Corpus. The findings aim to guide future research by presenting the effectiveness of existing approaches.

The paper titled "Phishing URL Detection with Neural Networks: An Empirical Study", authored by Hayk Ghalechyan, Elina Israyelyan, Avag Arakelyan, Gerasim Hovhannisyan, and Arman Davtyan, investigates the application of deterministic and probabilistic neural networks for phishing URL detection. The study highlights the increasing threat of phishing attacks, especially those involving URLs, and the shortcomings of traditional detection methods such as blacklisting.The authors developed a novel approach leveraging probabilistic neural networks (PNNs) to enhance detection accuracy, achieving a validation accuracy of 89% on datasets from sources like Alexa, Phish Tank, Open Phish, and private data from Easy DMARC. They compared the performance of

deterministic models with PNNs, showing that PNNs provide additional validation through uncertainty estimation. Their results indicate superior accuracy on both short and long URLs, a notable achievement considering the challenges with short URLs in phishing detection. The study also highlights the importance of real-world production data, demonstrating the model's effectiveness over six months of deployment. The authors conclude that integrating probabilistic models offers significant improvements in prediction reliability and supports the broader adoption of explainable AI in cybersecurity.

The paper 'Phishing URL Detection: A Real-Case Scenario Through Login URLs' on phishing URL detection highlights various approaches explored in recent research. Traditional methods, such as list-based approaches, use blacklists or whitelists to block or allow URLs, but these systems struggle with newly created phishing sites due to limited update frequencies. To overcome these limitations, researchers have focused on machine learning and deep learning models. Machine learning-based methods are divided into URL-based and content-based approaches. URL-based models extract features like URL length, special characters, and domain information, while content-based methods analyze web page source codes and external data like WHOIS records. Recent advancements include employing deep learning models such as CNNs, RNNs, and GRUs for automatic feature extraction and classification. Despite these advancements, most studies rely on outdated datasets or legitimate homepages rather than login URLs, limiting real-world effectiveness. This gap motivated the creation of the PILU-90K dataset, focusing on legitimate login URLs to enhance phishing detection accuracy in practical scenarios.

III. Proposed Solution

Linear Regression, is effective for predicting continuous values, is unsuitable for binary classification tasks because it can produce outputs outside the range from 0 to 1.

It also presumes a linear correlation between variables, which often does not come true in classification scenarios, resulting in inaccurate predictions. In contrast, Logistic Regression - specifically created for tasks involving classification and addresses these limitations by a sigmoid function to transform outputs to probabilities between 0 and 1, ensuring meaningful and interpretable results. Additionally Logistic Regression is effective in handling binary outcomes and offers a probabilistic framework that enables threshold adjustments based on specific application needs. It also includes regularization techniques (e.g., L1 or L2) to manage multicollinearity and improve generalization, which Linear Regression lacks inherently. These features make Logistic Regression more reliable and versatile for tasks like phishing detection, fraud prevention, and other classification problems where precise and bounded predictions are essential.

Linear Regression also presumes that the residuals (errors) follow a normal distribution, which isn't always the case, leading to inaccurate predictions. The model is sensitive to the scaling of input variables; if features have different units or magnitudes, it can distort the results, requiring normalization or standardization.

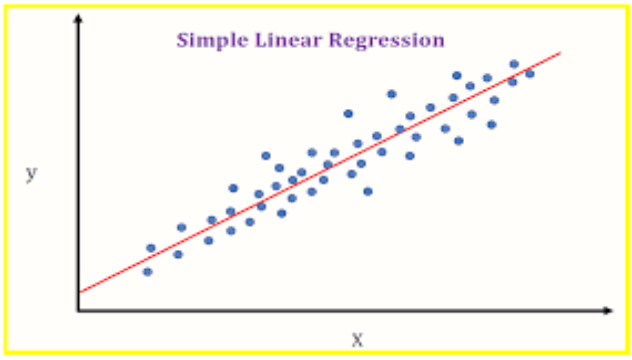

Fig.1: Linear Regression

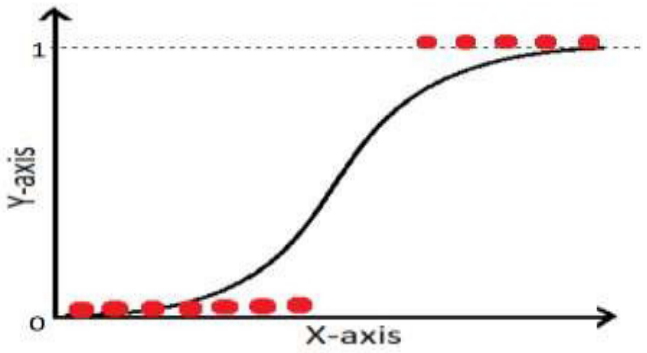

Fig.2: Logistic Regression

Logistic Regression is a popular algorithm for binary classification tasks, valued for its simplicity, interpretability, and efficiency.

By making use of a sigmoid function, Logistic Regression maps linear combinations of input features to probabilities from 0 to 1, ensuring meaningful and interpretable outcomes. This capability is specifically useful in fields like healthcare, finance, and cybersecurity, where understanding the likelihood of an event is as important as the classification itself.

Another significant advantage is its computational efficiency. Logistic Regression can handle large datasets effectively, making it a practical choice for real-time applications. It also works well when the relationship between features and also the target variable is approximately linear, providing reliable results without requiring complex transformations. Furthermore, Regularization techniques like L1 and L2 can be applied to mitigate overfitting and enhance model generalization, and identify the most impactful features.

Logistic Regression -versatile, extendable to multiclass problems using approaches like one-vs-rest or soft max, and serves as a robust baseline for more complex models. Its straightforward nature makes it easy to implement and interpret, offering transparency in understanding how predictions are made, which is critical in sensitive applications.

Few of the domain metrices used are:

A. Accuracy

Accuracy is the proportion of all classifications that were correct, whether positive or negative. In the spam classification example, accuracy measures the fraction of all emails correctly classified. An ideal model would have no false positives or false negatives, resulting in an accuracy of 1.0 or 100%.

Because it incorporates all four outcomes from the confusion matrix (TP, FP, TN, FN), given a balanced dataset, with similar numbers of examples in both classes, accuracy can serve as a coarse-grained measure of model quality. For this reason, it is often the default performance metric used for generic or unspecified models carrying out generic or unspecified tasks.

However, if the dataset is imbalanced, or where one kind of mistake (FN or FP) is more costly than the other, which is the case in most real-world applications, it's better to optimize for any one of the other metrics instead.

B. Recall

The true positive rate (TPR), or recall, quantifies the proportion of actual positives correctly identified as positive. The false negatives are the actual positives that were incorrectly classified as negatives, thus appearing in the denominator. In relation to the spam classification task, recall measures the ratio of actual spam emails that are correctly classified, identified as spam identified as spam also known as the possibility of detection. It essentially answers the question: "What proportion of spam emails are successfully detected by the model?" In a situation, where there are no errors, the recall (TPR) would be 1.0, representing a 100% detection rate. However, in imbalanced datasets, where the quantity of actual positive instances is extremely low (e.g., only 1-2 examples), recall may not be as informative or valuable as a performance metric, as it could overstate the model's effectiveness in handling rare positive cases.

C. Precision

Precision is the fraction of positive predictions made by the model that are actually correct. It is mathematically expressed as:
In regard with Phishing URL classification, precision measures the fraction of emails classified as spam that are indeed spam. A perfect model will result in zero false positives, resulting in a precision of 1.0, indicating that every instance labelled as positive is correctly classified.

In scenarios involving imbalanced datasets, where the count of actual positive instances is extremely low (e.g., only 1-2 examples), precision may not provide a comprehensive view of model performance, as it could be due to misleading the scarcity of positives.

Precision improves as false positives decrease, while recall improves when false negatives are minimized. However, adjusting the classification threshold influences both metrics: raising the threshold typically reduces false positives and increases false negatives, while lowering it has the opposite effect. Consequently, precision and recall often exhibit an inverse relationship, where enhancing one metric can lead to a decrease in the other.

D. F1 Score

The F1 Score is a metric that balances Precision and Recall, making it especially useful for evaluating classification models, particularly in situations with imbalanced datasets. It is derived as the harmonic mean of Precision and Recall, ensuring that both metrics are treated equally and balanced.

IV. Existing Methods:

Existing methods for phishing detection utilize a combination of traditional and advanced techniques to identify malicious activities. These include heuristic- based approaches that rely on predefined rules, blacklist and whitelist systems to block known phishing URLs, and machine learning models those analyze patterns in URLs, domain features, and webpage content.

1. The Random Forest algorithm is-robust and widely

used method in phishing detection, leveraging its ensemble learning capabilities to identify malicious patterns in URLs, webpage content, or network traffic. Using The Random Forest classifier is a collection of decision trees is trained on subsets of this data, each voting on whether a URL or webpage is phishing or legitimate. The majority vote from these trees is used to generate the final prediction in classification tasks, while averaging the outputs is employed for regression tasks which determines the final prediction, providing a binary classification output. During training, the model's performance is validated using measures like accuracy,
precision, recall, F1-score, and recall and support to ensure reliability. To enhance real-time applicability, the model is optimized and deployed in environments such as browser plugins, email filters, or security monitoring systems, where it can analyze and flag suspicious activities effectively By training multiple decision trees on various subsets of data and features, which helps improve the model's robustness and reduces the risk of overfitting using random feature selection, the algorithm ensures that no single feature or data subset dominates the learning process.

2. The Gradient Boosting Classifier (GBC) is a robust machine learning algorithm often used in phishing detection because of its capability to handle complex data and improve prediction accuracy through sequential learning and deliver high accuracy. It works by sequentially building an ensemble of decision trees, where each tree corrects the errors of the previous ones. In phishing detection, features such as URL length, URL style, presence of special characters, and content attributes are extracted from websites or emails. Initially, a weak decision tree is trained to determine whether a sample is phishing or legitimate. The algorithm calculates the errors (residuals) from this prediction and uses them to create another decision tree, focusing on minimizing these residuals. This process is repeated, with each new tree learning from the mistakes of the previous ones, and the predictions are combined using a weighted sum to produce the final output It operates using a boosting technique, which combines multiple weak learners—typically decision trees—into a strong predictive model. Unlike bagging methods such as Random Forest, where multiple trees are constructed independently, Gradient Boosting Classifier (GBC) creates trees in a sequential manner, with each tree learning from the errors of the previous model.The algorithm optimizes a loss function by adding trees that correct prediction errors through gradient descent. This makes GBC particularly effective in handling imbalanced datasets, noisy data, and complex feature interactions. Additionally, it supports various hyperparameter tuning options, such as learning rate, maximum depth of the trees, and the number of
3. estimators, allowing for tailored model performance based on specific tasks.

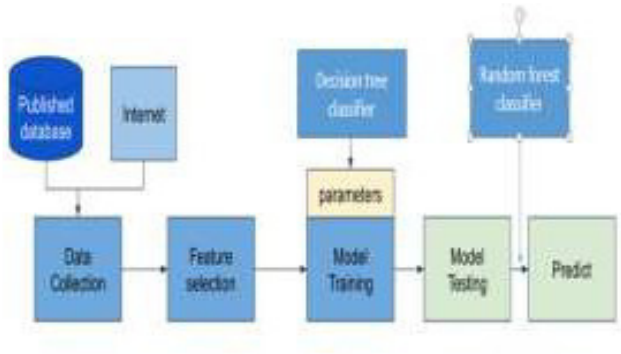

Fig .3. Working of Random Forest Algorithm

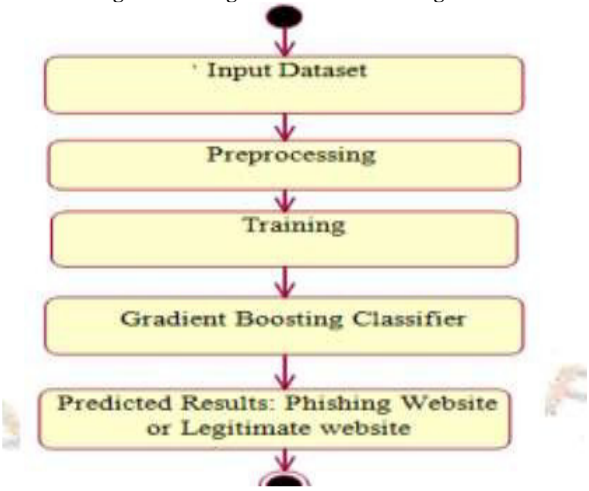

Fig.4.Working of Gradient Boosting Algorithm

It operates using a boosting technique, which combines multiple weak learners—typically decision trees—into a strong predictive model. Unlike bagging methods like Random Forest, where multiple trees are built independently, GBC builds trees sequentially, with each one learning from the previous model's mistakes. The algorithm optimizes a loss function by adding trees that correct prediction errors through gradient descent. This makes GBC particularly effective in handling imbalanced datasets, noisy data, and complex feature interactions. Additionally, it supports various hyperparameter tuning options, such as learning rate, maximum depth of the trees, and the number of estimators, allowing for tailored model performance based on specific tasks.

V. Conclusion

In conclusion, detecting phishing websites is an essential task in safeguarding users from online threats, and Logistic Regression offers a robust and efficient solution for this problem. By analyzing features such as URL

structure and suspicious patterns in content, Logistic Regression can effectively identify websites as phishing or legitimate. Its simplicity, interpretability, and ability to provide probabilistic outputs make it ideal for phishing detection systems. Although Logistic Regression may need extra feature engineering and regularization methods to enhance its performance on complex datasets, its ability to handle binary classification tasks with high accuracy remains a key advantage. As phishing tactics continue to evolve, integrating Logistic Regression with other machine learning models or ensemble methods can further improve its detection capabilities, ensuring a reliable defense against phishing threats tasks such as setting reminders, making phone calls, or ordering products.

VI. Future work

The future scope of this phishing detection project includes several promising avenues for enhancing the system's effectiveness and accuracy. By accessing a well-structured dataset of phishing attempts, the detection speed can be significantly increased. Combining multiple classifiers can further improve the system's accuracy, harnessing the advantages of various algorithms.

Additionally, exploring other phishing strategies that utilize features such as Lexical, Network, Content, Webpage, and HTML/JavaScript can provide a more comprehensive defense against phishing attacks. Future research can focus on refining URL feature extraction techniques and integrating them with diverse classifiers to create a robust and adaptable anti-phishing mechanism.

These improvements will not only enhance real-time phishing detection but also contribute to building a more secure online environment. These systems can analyse substantial amount of data and identify phishing attempts in real-time, improving response times and preventing potential attacks before they cause harm. Moreover, incorporating user behaviour analytics could help in identifying phishing attempts that are customized to specific individuals or organizations, adding an additional layer of detection. Another important area for future development is the use of blockchain technology to create a decentralized and transparent system for verifying websites and URLs, which could reduce the likelihood of phishing. Collaborative efforts between academia, industry, and government entities could result in the development of standardized anti-phishing protocols, ensuring a more unified and effective defense across different platforms. Additionally, continuous updates and maintenance of detection models are essential to adjust to the changing nature of phishing attacks, ensuring the system remains one step ahead of attackers.

References

[1] Hossein Shirazi, Kyle Haefner, Indrakshi Ray: Improving auto detection of phishing website using fresh-phish Framework

[2] Abdelhakim Hannousse, Salima Yahiouche.: Towards Bench mark datasets for machine learning based website phishing detection published on 27th October,2020.

[3] Furkan Colhak, Mert Ilhan Ecevit, Bilal Emir Ucar: Phishing website detection through multi-model analysis of HTML Content issued on 10th July 2024.

[4] Ammar Odeh, Ismail Keshta, Eman Abdelfattah: Machine Learning techniques for detection of website phishing on 2021

[5] Swathi Kakarla: A blog that contains decision tree, Phishing URL detection with python and ml published on 11th February 2021.

[6] Asif Ejaz, Adnan Noor Mian and Sanuallah Manzoor: Scientific report on life-long phishing attack detection using continual learning published on 17th July,2023.

[7] Keerthana Shankar, U Rithika, Sathya M, Vaishnavi Chitapur, Tejaswini: Detection of phishing website using Machine Learning issued on January 2024.

[8] Ayan Mahmood, Vishal Pandey ,Rohit Raj, Gouri Shankar Mishra: Detection of phishing sites using machine learning techniques issued on February 2024.

[9] Mohammed Hazim Alkawaz , Stephanie Joanne Steven: Detection of Phishing Website Using Machine Learning published on February 2020.

[10] B Amani Alswailem, Bahayr Alabdullah, Norah Alrumayh, Aram Alsedrani: Detection of Phishing Website using Machine Learning. DOI:10.1109/CAIS.2019.8769571 on May 2019.

[11] Article on Phishing attacks published on 9th March 2021 by Zainab Alkhalil, Chaminda Hewage, Liqaa Nawaf,Imtiaz Khan

[12] Mahajan Mayuri Vilas, Kakade Prachi Ghansham , Sawant Purva Jaypralash , Pawar Shila: Detection of phishing website using ML approach published on December 2019.

[13] Rishikesh Mahajan , Irfan Siddavatam: Phishing website detection using ML algorithms published on October 2018 DOI:10.5120/ijca2018918026

[14] Marwa AI saedi, Nahla Abbas Flayh: Phishing website detection using Machine Learning on June 2023. DOI:10.31185/wjps.145

[15] Alexey Natekin , Alois Knoll : Gradient Boosting Machine published on December 2013 DOI:10.3389/fnbot.2013.00021

[16] Arshid Ali, Muhammad Nouman Khan Ndeem Javed: Gradient boosting machine learning Algorithm published on December 2023 DOI:10.13140/RG.2.2.31609.65123

[17] Alabhya Farkiya , Prashanth Saini , Shubham Sinha: Natural Language Processing using NLTK and WordNet published in IJCSIT on 2015

[18] Frank Hutter, Lars Kottoff, Joaquin Vanschoren book on Automated Machine Learning

[19] B. Smith, “An approach to graphs of linear forms,” unpublished manuscript.

[20] E. H. Miller, “A note on reflector arrays,” IEEE Transactions on Antennas and Propagation, accepted for publication.

