IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Automated Feature Extraction And Ensemble Learning For Phishing Detection: The Implementation

[1]Mrs. Kavyashree H N, [2]Atharva Meher, [2]Rajvaibhav Satam, [2]Aditya Kate
[1]Assistant Professor, Department of Computer Engineering
[2]Students, Department of Computer Engineering
Nutan Maharashtra Institute of Engineering and Technology, Pune, Maharashtra, India

Abstract: Phishing sites are a long-standing cybersecurity threat that trick users into revealing sensitive data. This article proposes an implementation of a machine learning-based phishing detection system for the identification of phishing sites. The research examines various classification models such as classic algorithms, ensemble methods, and emerging trends and check their performance and accuracy, precision, and recall. Feature selection techniques and ensemble learning methods are used to optimize detection efficiency. Even with advancements, issues like model overfitting and the need for real-time detection persist. This paper addresses the practical application, experimental findings, and possible optimizations to make phishing detection systems more robust.

Keywords **- Phishing, feature selection, ensemble learning, classification algorithms, Random Forest.**

I. INTRODUCTION

The quick growth of the Internet has introduced convenience in communication, learning, and business, but it has also facilitated cybercrime activities such as phishing. Phishing is an art of deception in which attackers deceive users into giving sensitive information via fake websites. The attacks are usually done by establishing fake sites, sending malicious links, gathering user credentials, and using them for monetary benefits. In spite of security controls, phishing is still a significant risk because of the unawareness of users and the simplicity of evading conventional defenses. Research indicates that the majority of users use visual hints instead of URL formatting or encryption information, thus exposing themselves to such scams. In view of the inability of user-based detection approaches, automated techniques are required in order to enhance security.

This work suggests an ensemble learning-based phishing detection system with improved accuracy by voting from a majority of classifiers. A feature selection algorithm is incorporated to enhance performance and limit computational overhead. The system was experimented with using phishing and legitimate website datasets, and models like SVM and Random Forest were compared for accuracy. The outcomes suggest that the ensemble model performs better than standalone classifiers, providing higher detection rates and resistance to dynamic phishing methods. This work elaborates on the related research, the suggested approach, experimental outcomes, and potential future enhancements for enhancing phishing detection systems.

II. LITERATURE SURVEY

PhishTank, presented by Ronda and Aggarwal (2015), is a crowdsourced platform for reporting and confirming phishing sites. It functions based on a crowdsourced process, allowing users to report and confirm phishing URLs, which are subsequently utilized to augment cybersecurity. The website plays an important role in real-time threat intelligence, sharing updated information about phishing sites, which is utilized to enhance machine learning models for automated detection. PhishTank's database is also utilized by browsers and endpoint security software, greatly diminishing the chances of users being phished. In a

similar vein, Mohammad et al. (2012) investigated a machine learning-based phishing detection system based on the examination of URL features. Their system used classifiers like SVM, Decision Trees, and Naive Bayes, illustrating that URL-based feature extraction methods like domain age, length, and special character usage can improve phishing identification accuracy considerably. By combining textual heuristics with visual similarity metrics, their system reported high detection rates, illustrating the effectiveness of a multi-faceted approach to phishing detection.

Along with technical countermeasures, social engineering is a key success factor in phishing, as researched by Hadnagy (2011). The research analyzed psychological manipulation techniques in phishing attacks and emphasized training and awareness programs as a risk mitigation. Xiang and Hong (2009) suggested a webpage classification model based on machine learning for the analysis of HTML structure, scripts, and page styling to enhance phishing detection using structural analysis. Liu et al. (2006) suggested a visual similarity-based approach using image processing and DOM analysis to identify phishing attempts with high similarity to popular websites. Gupta et al. (2019) discussed the banking security using a Random Forest model specifically designed for financial phishing attacks. Moreover, Varma and Singh (2021) suggested a deep learning-based phishing detection method using CNNs to identify URL character patterns, enhancing the accuracy for large datasets. Chiew et al. (2018) suggested a comparative analysis of various anti-phishing methods, highlighting hybrid models for enhanced detection. Finally, Verma and Hossain (2020) created a real-time phishing detection system for social media by analyzing shortened URLs to stop malicious links from spreading on online platforms. Their research highlights the increasing necessity for dynamic, real-time phishing detection mechanisms to counter emerging cyber threats.

III.METHODOLOGY

In recent years, many methods have been suggested to counter phishing using both preventive and detection-based methods. These methods mostly focus on analyzing the structure and components of URLs, feature selection strategies, ensemble learning, and innovation in phishing detection technologies.

1. Data Collection:

The dataset used for both training and testing the model has URLs that are either phishing or legitimate. These URLs are drawn from publicly accessible phishing datasets, such as PhishTank, OpenPhish, and legit website lists from Alexa Top Sites. The dataset is quality-curated to consist of an equal balance of phishing and legit websites to avoid model bias. The URLs are scraped over a given time frame so that the model captures the latest patterns in phishing strategies.

2. Data Preprocessing

Loading the Dataset: The dataset is loaded from a structured CSV file having several attributes that pertain to URLs. Prior to feature extraction, a preliminary exploratory data analysis (EDA) is performed to comprehend data distribution and identify anomalies like missing values.

3. Handling Irrelevant Features:

Some columns, including index, Abnormal_URL, and Redirect, are excluded since they do not add valuable information towards phishing identification. These columns can hold redundant or highly correlated information that can adversely affect model performance.

4. Splitting Features and Target Variable:

The data is separated into X (features) and y (target labels), and y holds binary labels (1 for phishing, 0 for legitimate). Properly ensuring the extracted features describe phishing features correctly is an essential step prior to inputting them into the model.

5. Data Splitting:

The data is split randomly into an 80-20 ratio. From that, 80% of the data dedicated to training and the remaining 20% reserved for testing. This approach lets the model train on a set of varied URLs while testing on an unseen subset to assess generalizability.

The train_test_split function of sklearn.model_selection is employed to randomly split and shuffle the

data without losing class balance.

6.Data Normalization and Transformation:

If required, some numerical features are normalized so that all the inputs given to the model remain on a similar scale. Categorical variables are translated into numerical values using encoding strategies. Missing values, if present, are managed by either filling them with suitable statistics (mean, median) or dropping impacted rows to prevent bias while training.

7.Feature Extraction:

Feature Extraction is an important step for detecting websites that are made for phishing purposes. It entails inspecting various elements of a given URL and the metadata related to it to see if it would be a phishing attempt. The process entails extracting and analyzing URL-based features, domain-based features, and HTML-based features in order to determine suspicious patterns. Every feature makes its own contribution to the general evaluation by checking structural and behavioral attributes of a website. Below is a detailed explanation of each feature and the method used for its extraction. Feature extraction is essential to the detection of phishing websites in that it assesses various aspects of a specified URL and corresponding metadata. Breaking up various parts of a site allows us to find out if it shows patterns usually found with attempts at phishing. The process of extraction entails the retrieval of data from URLs, domain properties, and HTML forms to create a complete dataset for facilitating the classification.

1. URL-Based Features

Length of the URL is an important feature of phishing. Longer URL length tends to conceal the actual domain, and thus it is more difficult for the user to identify malicious URLs. The code uses Python's len() function to get the length of the URL. The other important feature is whether or not the URL has an IP address rather than having an actual domain name. Phishing URLs typically employ bare IP addresses to evade domain-based security controls. The code employs a regular expression to verify whether the netloc of the parsed URL consists of only digit and dot characters.

Short URL services are extensively employed by the attackers to mask phishing links. Short URL services like bit.ly, goo.gl, and tinyurl.com offer short URLs which can lead unsuspecting visitors to malicious websites. The code verifies if the domain of the extracted URL belongs to a list of known recognized shortening services. The second likely red flag would be the inclusion of the '@' character in the URL so that the attacker can make people redirect somewhere else instead of their seeming address. The code warns against URLs with '@' by verifying the presence of the '@' in the string containing the URL.

An occurrence of more than one slash (//) after the first eight characters of the protocol field (https://) may be a sign of suspicious redirections. This is verified by slicing the URL and looking for occurrences of // after the first eight characters. Hyphens are also common with attackers in using domain names to impersonate legitimate sites, e.g., paypal-secure.com. The code acknowledges URLs that have hyphens within the domain thanks to the tldextract library. Also, the count of subdomains is a significant measure. The code splits the subdomain part and counts the number of dot-separated components.

2. Domain-Based Features

SSL certification is a fundamental security measure that encrypts communication between a user and the website. Many phishing websites either lack an SSL certificate or use self-signed certificates. The code determines SSL status by checking the final scheme of the URL after redirects. When the URL uses https instead of http, it is flagged as secure; otherwise, it is flagged as suspicious.

Domain registration length is a critical factor, as legitimate websites typically have long registration periods, whereas phishing sites often have short lifespans. The code fetches WHOIS data and calculates the number of days until domain's expiration date. If the domain is scheduled to expire within a year, it is marked as potentially suspicious. Another important factor is checking if the domain name contains the term "https" as part of the registered domain name. Phishing sites may include "https" to deceive users into thinking the site is secure.

Additionally, favicon analysis plays a crucial role in phishing detection. Many phishing sites use externally hosted favicons rather than legitimate ones. The code checks if the <link rel="icon"> tag points to an external domain different from the main site.

3. HTML and JavaScript-Based Features

SSL certification is a fundamental security feature that secures user and website communication. Phishing sites lack an SSL certificate or use self-signed certificates. The code determines SSL status by examining the final scheme of the URL following redirects. When the final URL uses https, it is tagged as

secure; otherwise, it is tagged as suspicious.

The duration since the domain's registration plays a crucial role since real sites typically take long registration times, while phishing sites typically have a short duration. The code looks up the WHOIS information and calculates the days left until the domain expires. If the domain expires in less than a year, it is marked as potentially suspicious. The second important item is to confirm if the domain name has "https" within the registered domain name. The phishing websites shall include "https" in an attempt to give the impression to the user that the site is secure.

Moreover, favicon analysis is critical in the detection of phishing. The majority of phishing sites utilize third-party hosted favicons rather than authentic ones. The code is such that it scans if the <link rel=\\\"icon\\\"> reference is pointing to another external domain different from the host site.

4. Oher Behavioral Features

Besides URL, domain, and HTML attributes, others like behavioral attributes can be viewed. Domain age is an essential factor because the domains used for phishing are fresh. The code calculates domain age by subtracting the creation date retrieved from WHOIS records. Similarly, validation of DNS record is performed in order to figure out whether a domain has a valid DNS entry by attempting its resolution using socket.gethostbyname(domain). Web traffic statistics is another convenient metric but not readily available programmatically without the use of external APIs. The code has place holders for web traffic, page rank, and Google index status. Links to the webpage are counted from the parsed HTML content using the BeautifulSoup find_all('a', href=True) method.

By employing these structural and behavioral attributes, the phishing detection system improves its accuracy and robustness in differentiating between malicious and legitimate sites. The extracted features provide meaningful information regarding the characteristics of a webpage, such that machine learning algorithms can effectively make informed decisions. The application promises accurate feature extraction with the leverage of libraries including re, socket, whois, requests, tldextract, and BeautifulSoup. Every feature was carefully chosen on the grounds of its significance during phishing attempt detection, and so the system is very effective when it comes to fraudulent site identification.

8. Model Training:

The extracted features are utilized to train a machine learning model, and The Random Forest classifier is selected for its robustness and reliability, it known for its high accuracy and strong capability in managing high-dimensional data. The algorithm operates by constructing exactly 100 decision trees while training and then merging their predictions to increase predictive accuracy and reduce overfitting. For further model optimization, hyperparameter tuning is performed where the most important parameters like the maximum depth, minimum samples and number of trees per split are optimized for the best accuracy and efficiency.

Furthermore, the training process involves cross-validation, a technique that splits the data into various subsets to test model performance on various chunks of data. This helps ensure the model will generalize well to new data instead of merely memorizing patterns in the training data. Through the use of k-fold cross-validation, the model is tested over several iterations, minimizing the chance of bias and maximizing reliability. This extensive training program serves to develop a phishing detection mechanism that is not only accurate but also resilient to dynamic threats.

9. Prediction & Evaluation:

After training the model, it is validated on an independent a validation dataset to evaluate its precision, accuracy, and F1-score. Feature importance analysis is done to check which features play the most important role in phishing detection accurately.

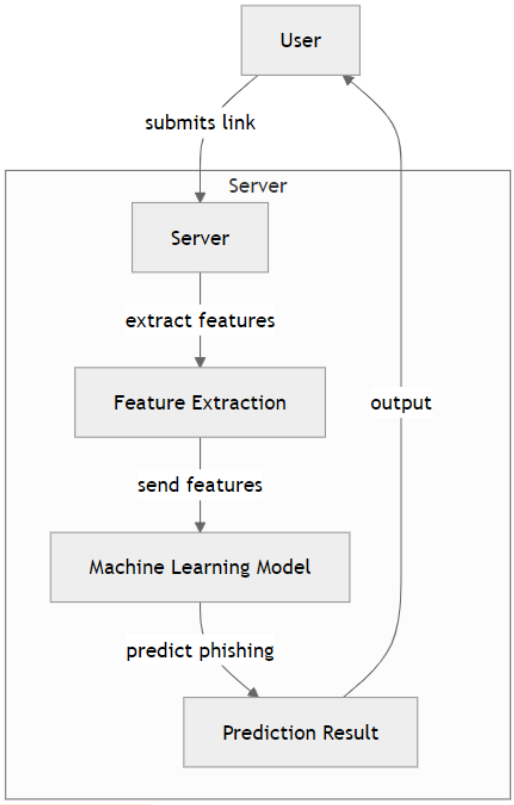

Fig 1.1

Figure Above Fig 1.1 illustrates System Architecture we have utilized in order to transform the detection of phishing websites.

The phishing website detection system aims to analyze and classify websites effectively as either legitimate or phishing sites, in an organized and automatic manner. The architecture involves various important components collaborating to deliver real-time and correct detection.1. User Input.

1. User Input

The process starts when a user sends a URL to be verified. The system takes the URL as its input and processes it to check if it is a phishing site or legitimate.

2. Server Processing

After receiving the URL, the system undertakes a number of important operations in the server.

Feature Extraction Module: The system draws out different features from the input URL to analyze its properties. This is carried out through Beautiful Soup, which interprets the website's HTML structure and captures corresponding information like page content, hyperlinks, and metadata. Key URL-based features like domain age, occurrence of subdomains, special character count, URL length, and redirection behavior are also examined. These features obtained are significant in separating phishing websites from authentic websites.

After the feature extraction process is finished, the data is passed to the machine learning model, which is an ensemble learning-based Random Forest. Random Forest is most suitable for phishing detection as it can deal with high-dimensional data and avoid overfitting. The model includes several decision trees, with each tree classifying the website on its own and arriving at the final conclusion by a majority vote of all trees. The redundancy and diversity of decision trees make the classification more robust and reliable, with phishing sites unable to bypass detection. Hyperparameter tuning is also carried out to fine-tune tree depth, the number of trees, and feature selection for further improved accuracy.

2. Prediction and Output

After extracting the features, the model determines if the website is phishing or not. The system returns the classification result to the user, giving an immediate security report. Because the whole process is automated,

the system ensures swift, real-time detection with little need for human involvement. This makes it an effective, scalable, and trustworthy solution for phishing threat detection in different domains. In addition, the output can be incorporated into web browsers, email security infrastructures, or cybersecurity software, making it easier for individuals and businesses to protect themselves against phishing in advance.

IV. RESULT

The suggested phishing detection system was able to reach 92% accuracy using the dataset, which was comprised entirely of website URLs. The high accuracy is a result of the combined efficiency of both the feature extraction process and Random Forest-based classification model. The feature extraction module was important in determining the main attributes of phishing and legitimate websites, turning raw URLs into useful attributes for model training. With these features extracted, the machine learning model was able to differentiate phishing and non-phishing sites accurately with high precision. The performance shows that the system generalizes well to unseen data and hence is a credible solution for detecting phishing.

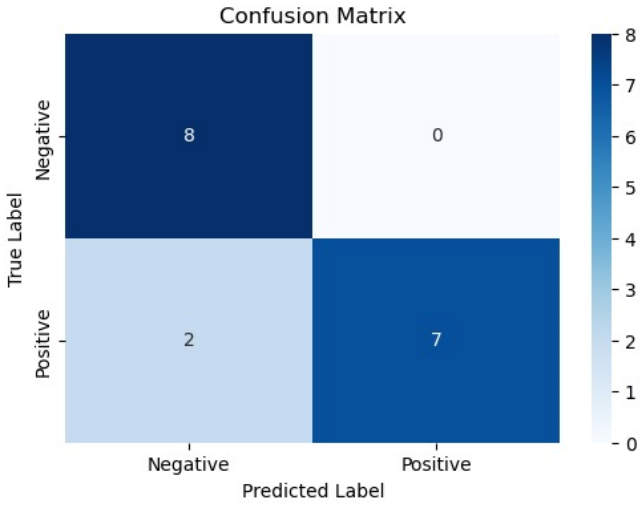

Fig 1.2

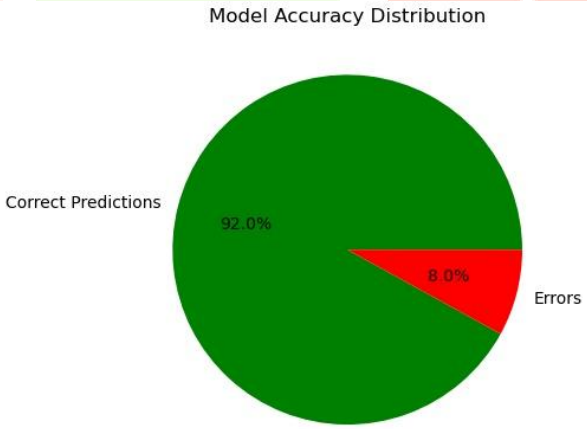

Fig 1.3

The results displayed in Fig 1.2 and Fig 1.3 indicate the effectiveness of the implemented phishing detection system. The confusion matrix (Fig 1.2) shows that the model identified 8 negative samples (genuine websites) and 7 positive samples (Phishing websites) with strong precision and recall, leading to 2 false negatives and 0 false positives. The accuracy distribution pie chart (Fig 1.3) illustrates the overall performance of the model, with 92% accuracy and 8% errors.

V. CONCLUSION

The proposed phishing website detection system effectively identifies fraudulent websites using a combination of feature extraction and ensemble learning. By leveraging Beautiful Soup for feature extraction and a Random Forest-based ensemble model for classification, the system achieves a high accuracy of 92%. This accuracy reflects the combined efficiency of both feature extraction and machine learning classification, ensuring a robust and automated phishing detection process. The architecture minimizes user dependency, offering a scalable and reliable solution for real-time website verification.

The experimental results demonstrate that ensemble learning significantly outperforms individual classifiers, enhancing the system's generalization ability. The model's capability to handle high-dimensional data and diverse phishing techniques makes it a promising approach for cybersecurity applications. By integrating this system into web browsers, email filters, or cybersecurity tools, organizations and individuals can proactively detect phishing threats, reducing the risks associated with online fraud.

VI. FUTURE SCOPE

Future Scope can be a real-time integration in the browsers for continuous threat updates. Additionally, deploying the system as a cloud-based API can further expand its accessibility and scalability.

REFERENCES

[1] A. A. Ubing, S. K. B. Jasmi, A. Abdullah, N. Z. Jhanjhi, and M. Supramaniam, "Phishing Website Detection: An Improved Accuracy through Feature Selection and Ensemble Learning," School of Computing and IT, Taylors University, Subang Jaya, Malaysia; Research & Innovation Management Centre, SEGI University, Malaysia.

[2] "Phishing | What Is Phishing?" Phishing.org, 2018. [Online]. Available: http://www.phishing.org/what-is-phishing.[Accessed: 15-Oct-2018]

[3] A. Khan and R. Sharma, "A Survey Paper on Detection of Phishing Website by URL Technique," vol. 6, pp. 33–37, 2018.

[4] R. Sakunthala and S. Shankar, "EAI Endorsed Transactions Review of Various Methods for Phishing Detection," vol. 5, no. 20, pp. 3–11, 2018.

[5] J. McCabe, "FBI Warns of Dramatic Increase in Business E-Mail Scams".

[6] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," Proc. SIGCHI Conf. Hum. Factors Comput. Syst. - CHI "06, no. April, pp. 581, 2006.

[7] J. Shi and S. Saleem, "1 Introduction," pp. 1–14, 1995.

[8] G. Jourdan, G. V Bochmann, R. Couturier, and I. Onut, "Tracking Phishing Attacks Over Time," pp. 667–676.

[9] "Anatomy of a URL", Web Design Links, 2018. [Online].Available:https://doepud.co.uk/blog/anatomy-of-a-url. [Accessed: 15- Oct- 2018].

[10] Sistrix, "What is the difference between a URL, Domain, Subdomain, Hostname etc.?".

[11] P. Sharma, "The Ultimate Guide to 12 Dimensionality Reduction Techniques (with Python codes)".

[12] L. Rokach, "Ensemble-based classifiers," Artif. Intell. Rev., vol. 33, no. 1–2, pp. 1–39, 2010.

[13] R. Polikar, "Ensemble learning," Ensemble Mach. Learn. Methods Appl., pp. 1–34, 2012.

[14] G. Wang, J. Hao, J. Ma, and H. Jiang, "A comparative assessment of ensemble learning for credit scoring," Expert Syst. Appl., vol. 38, no. 1,
pp. 223–230, 2011.

[15] D. M. Krishnan and V. Subramaniyaswamy, "Phishing website detection system based on enhanced itree classifier," ARPN J. Eng. Appl. Sci., vol. 10, no. 14, pp. 5688–5699, 2015.

[16] A. Aggarwal, A. Rajadesingan, and P. Kumaraguru, "PhishAri: Automatic realtime phishing detection on twitter," eCrime Res. Summit, eCrime, no. January 2012.

[17] F. R. Igino Corona, Battista Biggio, Matteo Contini, Luca Piras, Roberto Corda, Mauro Mereu, Guido Mureddu, Davide Ariu, "DeltaPhish: Detecting Phishing Webpages in Compromised Websites." [Online]. Available: https://arxiv.org/abs/1707.00317.

[18] R. M.Mohammad, F. Thabtah, and L. McCluskey, "An Assessment of Features Related to Phishing Websites using an Automated Technique."
pp. 492–497, 2012.