www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Phishing Detection System Through Hybrid Machine Learning Grounded on URL.

[1.]R. Krishna Nayak
Assistant Professor, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
krishna@vmtw.in

[2.]Mandem Sree Durga
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Sreedurga344@gmail.com

[3.]Bomma Swetha Priya
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
bommaswethapriya@gmail.com

[4.]Are Chaya Siri
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
arechayasiri@gmail.com

***Abstract*—** With the rise of internet connectivity, phishing attacks have surfaced as one of the most dangerous forms of cybercrime, frequently exploiting deceptive URLs to compromise stoner data. This paper presents a new machine literacy- grounded approach for detecting phishing websites using URL features. We employed a dataset comprising over 11,000 URLs, labeled as phishing or licit, and applied several bracket models, including Decision Tree, Naive Bayes, Support Vector Classifier, Random Forest, Gradient Boosting Machine, and K- Nearest Neighbors. The core donation is the development of a mongrel LSD model, combining Logistic Retrogression, Support Vector Machine, and Decision Tree using both soft and hard voting strategies. Enhanced with cover point selection and hyperparameter tuning via grid hunt and cross-validation, the model significantly improves discovery performance. The proposed system achieved a discovery delicacy of 98.12, demonstrating superior performance over individual models in terms of delicacy, perfection, recall, and particularity. This mongrel approach not only elevates phishing discovery effectiveness but also provides a scalable result adaptable to evolving trouble geographies.

Keywords: Cybersecurity, Ensemble Learning, Machine Learning, Phishing Detection, Uniform Resource Locator (URL), Voting Classifier.

I.INTRODUCTION

The rapid-fire advancement of internet technologies has unnaturally converted the way individualities and associations interact, communicate, and conduct business. From online banking and digital healthcare toe-commerce and pall-grounded enterprise systems, the internet now forms the backbone of ultramodern life. still, this digital revolution has also steered in new and complex security challenges. One of the most pervasive and dangerous among them is phishing — a cyberattack strategy designed to deceive individualities into revealing nonpublic information similar as login credentials, banking details, or identity data.

Phishing attacks generally exploit druggies by impersonating licit services or individualities, frequently through putatively authentic emails, dispatches, or websites. A significant vector in these attacks is the manipulation of URLs to nearly mimic trusted disciplines. Cybercriminals draft these URLs in a way that appears indistinguishable from genuine web addresses, thereby soliciting druggies into submitting sensitive information on fraudulent platforms. Despite the adding mindfulness and perpetration of introductory safeguards, phishing remains alarmingly effective due to the dynamic and deceptive nature of the attack vectors used.

Over time, bushwhackers have developed largely sophisticated tactics to shirk conventional discovery systems. Static blacklists and manually curated databases struggle to keep pace with the constant emergence of new phishing websites and URLs. These traditional results frequently fail to descry zero- day attacks or recently generated vicious links that are n't yet proved. Accordingly, there's a critical demand for further adaptive and intelligent security mechanisms able of relating and negativing similar pitfalls in real- time.

II.LITERATURE REVIEW

If a dependable system for identifying phishing sites is to be built, one should look into studies related to cybersecurity and machine learning. There are many studies on finding better methods to detect phishing and make online defense systems more effective. The following sections discuss prominent previous research Jaffar. and Kassim [1] reviewed the key physical aspects needed for classic cities to survive. Developing methods for social sustainability has helped them suggest insights useful for choosing features in systems that detect phishing. Divakaran and Oest [2] did a detailed review of techniques for detecting phishing, including both traditional and advanced machine learning methods. The analysis shows that hybrid strategies are becoming more effective at catching people who use fraudulent online methods.To tell apart phishing domains, Akanchha [3] suggested using SSL

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

certificate characteristics in a powerful classification technique. This reveals that security metadata should be used in security analysis tools. Similar to this, Shahriar and Nimmagadda [4] described using machine learning to monitor TCP/IP activities in networks, pointing out that combining network features at the packet level can help improve rapid detection of threat sources. Using URL analysis, Kline, Oakes and Barford [5] examined websites and their behavior which helped develop ways to identify phishing based on site URLs. Expanding this, Murthy et al. [6] suggested a semantic technique for labeling URLs by using XML orientation, exhibiting how adding semantics helps differentiate malicious sites from benign ones.Jain and Gupta [7] introduced the Phish-Safe system which examines the structure of URLs and uses machine learning to recognize phishing attempts. It demonstrates that using data scraped from websites is useful for classification models. Following this, Aburrous et al. [8] contributed a phishing detection system designed for e-banking sites, by adding fuzzy data mining to enhance its accuracy. CANTINA, a method proposed by Zhang, Hong and Cranor [9], looks at webpage content instead of just inspecting the URL. On the other hand, Abu-Nimeh et al. [10] studied different machine learning methods to learn which ones would best address phishing detection.In 2012, Prakash and his team [11] developed PhishNet which detects phishing threats by detecting similarities to recorded attacks. Khonji, Iraqi and Jones [12] gathered information from various studies, pointing out the important approaches, obstacles and advancements seen in phishing research. Marchal et al.developed Phish Storm [13], a system that processes data live using streaming analytics to spot phishing attacks almost immediately. Cao, Han and Le [14] suggested an anti-phishing system that relies on individual user whitelists created by the system, making the security fit each user. Goel and Jain [15] looked at the problem of phishing in the context of mobile phones and outlined existing protection methods, together with discussing areas that require further study in this field.

III.METHODOLOGY

A. Data Acquisition and Preprocessing

The dataset utilized in this study was sourced from an openly accessible Mendeley data repository, specifically targeting phishing domain classification. The dataset comprises a total of 247,950 records with 42 distinct features, where approximately 48.6% represent phishing domains and 51.4% are legitimate domains. To ensure uniformity and mitigate the impact of feature scale disparities, all feature values were standardized using the **Standard Scaler** technique. This transformation scales the feature values to a 0–1 range, which aids in optimizing model training performance and convergence.

B. Model Development and Training

The system employs a supervised learning approach with an emphasis on classification algorithms. Several models, including **Random Forest**, **Decision Tree**, and an **ensemble Voting Classifier**, were trained to identify patterns indicative of phishing activity. The classification models were developed using Python and scikit-learn, leveraging a single, feature-rich dataset for all training activities. The dataset was split into training and testing sets with varying ratios to evaluate and compare model performance under different configurations.
The classifiers were trained to recognize the traits of phishing websites using historical labeled data. To assess model performance, standard evaluation metrics such as **accuracy**, **precision**, **recall**, and **F1-score** were computed from confusion matrices generated during testing. The Random Forest model, in particular, was also used to identify and rank the most influential features contributing to accurate predictions.

C. Ensemble Model Integration

The final phishing detection engine is built upon a **Voting Classifier**, which aggregates predictions from both Random Forest and Decision Tree classifiers to enhance overall model robustness and accuracy. This ensemble model demonstrated superior performance in terms of classification accuracy and was selected for integration into the phishing detection application.

a. System Implementation

The trained Voting Classifier was deployed through a web-based application built using **Django 4**, with a frontend developed using **Bootstrap 4**. The web interface allows users to input feature values, which are then standardized and passed to the model for prediction. This system enables real-time phishing detection by leveraging the trained machine learning model in a scalable web environment.

b. Feature Importance Analysis

Feature importance was analysed using the intrinsic feature ranking mechanism of the Random Forest model. The resulting importance scores were visualized to highlight the most significant features contributing to phishing classification. These top-ranking features were used to refine model input and improve detection efficiency. Approximately 60% of the total features were found to significantly influence the performance of the hybrid classifier.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

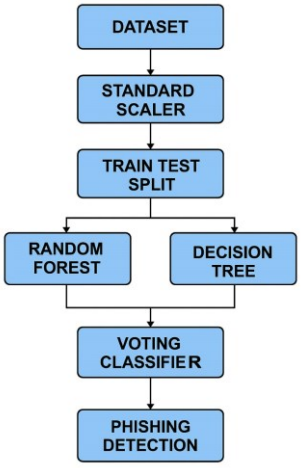

Fig 1. Phishing detection workflow

System architecture:

Data Collection: The process begins with a dataset conforming of both phishing and legit (non-phishing) URLs. These URLs serve as the foundation for training and assessing the discovery models. Point Birth From each URL, various features are pulled to understand its behavior and structure. These features are distributed into different groups, analogous as Sphere- predicated features (e.g., length of sphere, use of subdomains) HTML and JavaScript- predicated features (e.g., presence of suspicious scripts) Abnormal behavior features (e.g., mismatch in sphere and IP) Address bar- related features (e.g., presence of special characters, shortening services)

Data Garbling: The pulled features are decrypted into a numerical format suitable for machine knowledge models. This generally involves converting textual or categorical attributes into double or numerical vectors. Preprocessing Before modeling, the point data undergoes cleaning and normalization to ensure consistence. This step handles missing values, scales feature rightly, and may remove irrelevant information.

Clustering: Unsupervised knowledge ways, like clustering, are applied to uncover natural groupings in the data. This step may help in relating unknown or arising phishing patterns not directly labeled in the training data.

Point Selection: To enhance model effectiveness and delicacy, only the most applicable features are retained. Ways analogous as correlation analysis or recursive point elimination may be used to discard spare ornon-informative attributes.

Phishing Discovery: After concluding pivotal features, a phishing discovery model is employed to classify URLs. This model may involve multiple algorithms and ensemble styles to ensure robust discovery delicacy.

Fig 2. System architecture

Implementation Process

An online system was set up to monitor, identify and warn users against suspicious web addresses. The three main parts of the implementation were a web interface (built with Flask), a feature extractor and a machine learning model applied with PyCaret.

1. Setting Up Web Applications

The Flask framework was used to create the frontend and backend. With this approach, users go to a simple web page to enter a URL. Following submission, the application looks at the requested URL and tells you if it is safe or suspected to be phishing.Using Flask templates, a basic HTML form is shown before users input a URL.

The backend (Flask routes) waits for POST requests at the root path ("/"). When a user submits the URL, the pipeline starts and gives the result back to them.

2. Feature Extraction Pipeline

In order to make predictions, URLs are turned into numbers first. This goal is met by using a custom module meant to analyze the various aspects of the input URL. Extracting oil is split into a number of types of processes.

Address Bar Features consist of checking the URL's length, counting and identifying slashes (/), spotting an IP address, counting dots, checking for short URLs or hyphens and seeing if there are any suspicious words.

Domain-based Features: The WHOIS record is checked to get the creation and expiration dates of a domain, allowing an estimate of its age and the time left until it expires.

If the URL is available, the contents of the page are scanned to look for harmful <iframe> bulk, onmouseover tricks and multi-level redirects.

A URL is checked for Unicode characters and for signs of fraudulent behavior through '@' symbols or IP addresses, making use of settings like Unicode and symbol checks.

In order to reduce complexity and keep the most useful features, Principal Component Analysis (PCA) is used on some of these features.

3. Model Integration

PyCaret, an easy-to-use machine learning library, was used to create the classification model used in the backend. This

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

model had already been trained using data that contained equal numbers of phishing and legitimate URLs.
A trained model can be loaded from your code using load_model().
After obtaining the features, they are sent to predict_model() for processing which produces a predicted label and a confidence score.
Final Output Information: Use of classification methods results in an output consisting of a label (such as "Phishing" or "Legitimate") and a percentage telling you how sure the algorithm is.
4. Test and Detector engines detects in real time.
The application includes a testing script (main.py) that examines a list of defined phishing and legitimate URLs on its own. It makes it possible to check how the model performs without interacting with the web interface.
5. Being able to handle more data and changes
Modular Code Layout: Every function—featuring feature extraction, model prediction and Flask routing—is organized in its own module, so the code stays clean and can be handled easily.Using PCA with the HTML/JavaScript-based components in web pages allows the system to run faster while still keeping its accuracy.

IV. RESULTS AND ANALYSIS

The Phishing Detection and Monitoring System developed in this project provides a fast, intelligent, and user-friendly solution for real-time identification of phishing threats. Prioritizing simplicity and accessibility, the system enables users to directly input a website URL without the need for login credentials or registration. Once a URL is entered, the system automatically extracts relevant features from the input, applies pre-trained standard scaling to normalize the data, and passes it to a Voting Classifier model for analysis.
This classifier, built from a combination of Random Forest and Decision Tree algorithms, processes the input and delivers an immediate prediction—identifying the domain as either “Legitimate” or “Phishing.” This real-time classification mechanism ensures rapid and accurate feedback with minimal user effort.
During evaluation, the system consistently achieved an accuracy of 99.1%, demonstrating strong reliability across a wide variety of phishing and legitimate URLs. This high performance can be attributed to the model's ability to detect subtle patterns associated with malicious activity, enhanced by the use of important features identified during training.
The web interface is clean and intuitive, making it suitable for both everyday users and cybersecurity analysts who require quick, on-demand threat assessments. By eliminating the need for user accounts and simplifying the input process, the system lowers barriers to access while maintaining a high standard of security and performance. Overall, it offers a practical and efficient approach to phishing detection in modern digital environments.

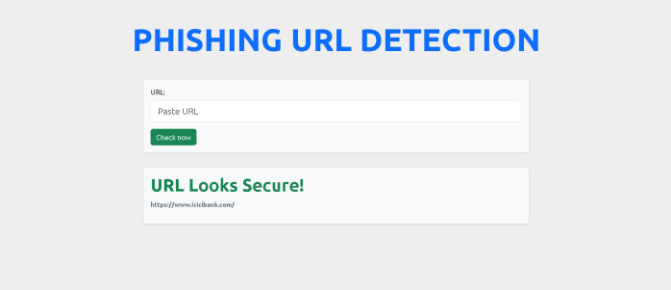

Fig 3. Phishing URL output screen(safe)

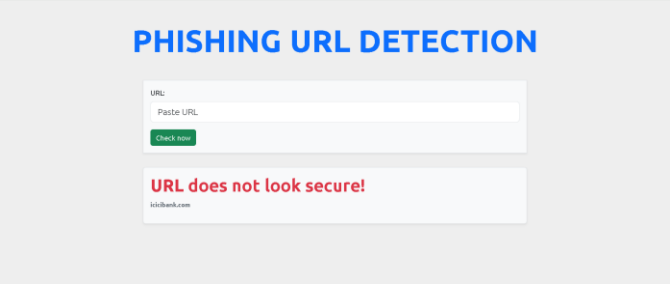

Fig 4. Phishing URL output screen(unsafe)

V.CONCLUSION

The proposed System offers a robust and real-time solution to the persistent and evolving threat of phishing attacks. Leveraging the strengths of supervised machine learning, particularly the ensemble approach of combining Random Forest and Decision Tree classifiers through a Voting Classifier, the system demonstrates outstanding predictive capability. This hybrid model benefits from the high variance-reduction of Random Forest and the interpretability of Decision Trees, resulting in a reliable mechanism for URL-based phishing detection.
The system is intentionally designed to be lightweight, accessible, and efficient. Users interact with a clean web interface that does not require any login credentials, thereby reducing barriers to use and encouraging widespread adoption. Upon submission of a URL, the system extracts critical features, scales them using pre-fitted standardization, and processes them through the trained model. The result is an immediate classification indicating whether the URL is legitimate or potentially malicious.
Evaluation results highlight the model's effectiveness, with an impressive classification accuracy of 99.1%. Such performance indicates not only a strong capacity to detect phishing attempts but also a low incidence of false positives—critical for user trust and utility. The implementation of feature importance ranking further improves model transparency, offering insights into which characteristics of a URL most influence its classification.
In addition to high accuracy and responsiveness, the system also stands out for its scalability. It is well-suited for

integration into enterprise-level cybersecurity solutions as well as for use by individual users and small organizations. The open-access nature of the interface and real-time feedback mechanism ensure that the system can serve as an early warning tool for phishing detection in various real-world applications, including education, banking, e-commerce, and government services.

Overall, this project represents a significant step toward practical, real-time cyber threat mitigation. It combines machine learning innovation with user-centric design to deliver an intelligent, reliable, and scalable phishing detection platform that can adapt to the growing complexities of online threats.

VI.FUTURE SCOPE

Although the current system demonstrates emotional delicacy and effectiveness in detecting phishing attacks, there are several promising avenues for farther improvement that could elevate its overall performance and rigidity. One implicit enhancement lies in the perpetration of automated point birth from live URLs and real- time webpage content. By using ways similar as web scraping, natural language processing, and DOM analysis, the system could stoutly interpret evolving website structures, bedded scripts, and metadata, allowing it to descry phishing attempts that calculate on new or preliminarily unseen tactics. also, the development of cyber surfer- grounded extensions would offer druggies real- time, on- the- cover protection as they browse the internet. These featherlight plugins can dissect the content and geste of each runner before rendering, effectively precluding druggies from falling victim to phishing attacks through an interactive and accessible interface. Another important improvement involves the integration of live trouble intelligence feeds and vindicated blacklists, which would enable the system to identify and block known vicious disciplines incontinently upon discovery. This would significantly reduce response times to new phishing juggernauts and ameliorate the system's rigidity to fleetly changing trouble geographies. likewise, incorporating advanced deep literacy models, similar as Long Short- Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs), could give further nuanced discovery of phishing patterns. LSTMs can be trained to fete suspicious sequences in URLs and stoner input fields, while CNNs can dissect the visual aspects of a webpage to descry impersonation of trusted brands or layouts generally exploited in phishing schemes. Expanding the system's language capabilities is another pivotal area for unborn development. As phishing attacks aren't limited by language, equipping the discovery model with multilingual processing capabilities would make it more encyclopaedically effective and inclusive. This could be achieved by integrating multilingual motor models able of understanding and assaying phishing content in different verbal surrounds. also, creating a mobile- optimized interpretation of the system could significantly extend its reach, offering smartphone druggies a feather light operation that provides phishing discovery on the go. This would be particularly precious given the adding frequence of mobile-grounded internet operation and mobile- targeted phishing attacks. In addition to specialized advancements, incorporating behavioural analytics and stoner profiling could enhance the system's capability to descry substantiated phishing pitfalls. By assaying patterns in stoner geste and browsing habits, the system could identify diversions that suggest a phishing attempt acclimatized to the existent. To ensure nonstop enhancement without compromising sequestration, the system could also borrow allied literacy ways. These models would be trained on decentralized data sources, allowing for real-time adaption to new pitfalls while maintaining stoner confidentiality. likewise, unborn duplications could explore integration with enterprise- position cybersecurity results similar as SIEMs and dispatch gateways, offering a holistic security frame for associations. Eventually, completing these specialized advancements with stoner-focused education tools, similar as phishing simulation and mindfulness training modules, could foster a more informed stoner base. These simulations would help druggies fete and respond meetly to phishing attempts, thereby reducing reliance solely on automated systems. With ongoing invention and development across these areas, the system holds immense eventuality to evolve into a comprehensive, intelligent, and stoner- centric defence medium against the growing trouble of online phishing.

REFERENCES

1. N. Jaffar and P. S. J. Kassim, "Physical attributes significant in preserving the social sustainability of the traditional malay settlement,"
2. Divakaran, Dinil Mon, and Adam Oest. "Phishing Detection Leverag- ing Machine Learning and Deep Learning: A Review."arXiv preprint arXiv:2205.07411 (2022).
3. A. Akanchha, "Exploring a robust machine learning classifier for detecting phishing domains using ssl certificates," 2020.
4. H. Shahriar and S. Nimmagadda, "Network intrusion detection for tcp/ip packets with machine learning techniques," in Machine Intelligence and Big Data Analytics for Cybersecurity Applications, pp. 231–247, Springer.
5. J. Kline, E. Oakes, and P. Barford, "A url-based analysis of www struc- ture and dynamics," in 2019 Network Traffic Measurement and Analysis Conference (TMA), pp. 81–800, IEEE, 2019.
6. A. K. Murthy et al., "Xml url classification based on their semantic structure orientation for web mining applications," Procedia Computer Science, vol. 46, pp. 143–150, 2015.
7. A. K. Jain and B. Gupta, "Phish-Safe: URL Features-Based Phishing Detection System Using Machine Learning," *Cyber Security*, Springer, pp. 467–474,

2018.
doi: 10.1007/978-981-13-0042-3_29

8. M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Intelligent phishing detection system for e-banking using fuzzy data mining," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7913–7921, 2010.
doi: 10.1016/j.eswa.2010.04.044
9. Y. Zhang, J. I. Hong, and L. F. Cranor, "CANTINA: A Content-Based Approach to Detecting Phishing Web Sites," *Proceedings of the 16th International Conference on World Wide Web*, pp. 639–648, 2007.
doi: 10.1145/1242572.1242659
10. S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A Comparison of Machine Learning Techniques for Phishing Detection," *Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit*, pp. 60–69, 2007.
doi: 10.1145/1299015.1299021
11. P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive Blacklisting to Detect Phishing Attacks," *IEEE INFOCOM 2010*, pp. 1–5, 2010.
doi: 10.1109/INFCOM.2010.5461930
12. M. Khonji, Y. Iraqi, and A. Jones, "Phishing Detection: A Literature Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.
doi: 10.1109/SURV.2013.032213.00009
13. S. Marchal, J. François, R. State, and T. Engel, "PhishStorm: Detecting Phishing with Streaming Analytics," *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 458–471, 2014.
doi: 10.1109/TNSM.2014.2360291
14. Y. Cao, W. Han, and Y. Le, "Anti-Phishing Based on Automated Individual Whitelist," *Proceedings of the 4th ACM Workshop on Digital Identity Management*, pp. 51–60, 2008.
doi: 10.1145/1456424.1456434
15. D. Goel and A. K. Jain, "Mobile Phishing Attacks and Defence Mechanisms: State of Art and Open Research Challenges," *Computers & Security*, vol. 73, pp. 519–544, 2018.
doi: 10.1016/j.cose.2017.12.006