iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71216

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Phish Catcher and Web Spoofing Attack Using Machine Learning

Mrs. Nazeema S[1], Ms. Abinaya D[2], Ms. Inika R K[3], Ms. Nandhini S[4], Ms. Akshaya E[5]
[1]*Assistant professor, Department of Computer Science and Engineering, Muthayammal Engineering College, Rasipuram, India*
[2, 3, 4, 5]*UG, Department of Computer Science and Engineering, Muthayammal Engineering College ,Rasipuram, India*

Abstract: ***The Phishing threats are one of the major evolving threats that breaks confidentiality and causes serious risk. They frequently serve as entrance points for a variety of cyberattacks, such as money fraud, malware distribution, and data theft. Phishing attacks are now emerging with more advanced tactics and technologies. Because we are mainly rely on manual feature engineering, traditional detection techniques have difficulty keeping up with emerging and complex phishing tactics like zero-day assaults. By automating feature extraction and enhancing flexibility, Machine learning (ML), Random forest classifier (RFC) and deep learning (DL) present a possible answer. These sophisticated algorithms improve detection accuracy while also better addressing new phishing techniques. The main objective of this study is to use ML algorithms to detect and predict the fake websites and reduce phishing attacks.***
Keywords: Phishing attacks, Machine Learning Algorithms (ML), Random Forest

I. INTRODUCTION

Phish Catcher is a state-of-the-art machine learning-based system that has the effective ability to detect and prevent phishing and web spoofing attacks. Phishing simply denotes an act of creating false websites identical to real ones to fraudulently acquire sensitive private information, such as passwords and credit card details. This web spoofing amplifies the threat even further, as now it can impersonate the trusted sites to be able to fool the user. Consequently, these threats are now at risk to online security and privacy breaches and financial losses.

The proposed project, therefore, uses some modern machine learning techniques such as supervised learning along with the feature extraction methods to analyze the critical attributes of a website that can be related to URL patterns or domain characteristics or content structure. It utilizes both supervised learning and feature extraction to be used in analyzing important properties of this site into two groups: legitimate or malicious. Phish Catcher continuously learns from new types of threats, thereby improving the accuracy of detection and providing the users with an advanced solution against evolving cyberattacks, thus making their online experience much safer and secure.

II. OBJECTIVE

The primary focus of the Phish Catcher project is the design and implementation of an intelligent, automated system to identify and block phishing and web spoofing attacks. The system intends to analyze website features and detect malicious websites in real time by leveraging complex machine learning algorithms. This proactive approach will consequently enhance online security for users and reduce the chances of data breaches by preventing them from becoming victims of fraudulent schemes.

The main objective is to develop a scalable and adaptable detection framework that will be modified continuously to address new threats. Conventional rule-based systems often find it difficult to adapt to the rapidly changing patterns of attacks. By integrating machine learning models, Phish Catcher can enhance its detection capabilities, providing a robust defense mechanism while dynamically learning from new datasets.

III. PROPOSED WORK

Proposed Phish Catcher system in use for phishing attacks and web spoofing attacks employs a multi-layered detection-based machine learning. Key feature extraction occurs from real-time website visits and publicly available phishing datasets including URL length, domain age, https usage, and the content of the webpage. These features are then fed to advanced machine learning techniques that classify websites as being either malicious or legitimate-including Random Forest, Gradient Boosting, and Neural Networks.

This hybrid performance means that using a combination of various algorithms for exhaustive detection increases accuracy and diminishes false positives. The system surveys any web traffic continuously by a real-time monitoring framework, thereby promoting fast detection and efficiency. A user-friendly web interface allows users to investigate dubious URLs and obtain fast feedback.

IV. SYSTEM ARCHITECTURE

A "Phish Catcher" system design is based around a client-side entity, often in the form of a browser extension, which intercepts web page content on load. This entity takes vital features such as URL properties, HTML layout, visual similarity, and content analysis. These are then plugged into a local machine learning inference engine, which determines the page to be legitimate or phishing, with user feedback presented through a UI. Local storage can keep recent website content or part of the ML model.

Optionally, a backend part extends functionality. This backend supports collecting data from sources such as phishing databases and user reports, tagging it for training ML models. The trained model is shipped to the client with timely updates. Other features comprise a phishing database that

has an API for client requests and logging/monitoring for ongoing improvement.The main architectural factors are performance to reduce the impact on users, accuracy to minimize false positives/negatives, scalability to support many users, security to avoid tampering, and privacy to safeguard user information. The complete architecture enables "PhishCatcher" to efficiently detect and block web spoofing attacks, enhancing internet security

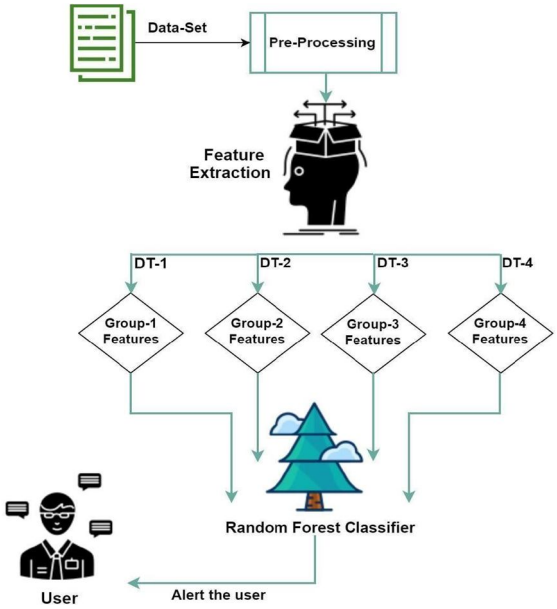


Fig 4.1. System Architecture

V. METHODOLOGY

A. Dataset Collection

For phishing detection, the dataset was obtained from an open-source platform named Phish tank, where data was available in CSV format.

The dataset had 18 columns in the beginning, and it was transformed through a series of data preprocessing methods. For exploring the features and familiarizing oneself with the data structure, some data frame methods were utilized. Visualization methods, like plotting and graphing, were used for analyzing the distribution of the data and the relationships among features. With an inspection, it was noted that the Domain column did not make sense in training the machine learning model.

Once this column was dropped, the dataset was cleaned to 16 features with a target column. Features of both legitimate and phishing URL datasets were concatenated during the feature extraction phase without shuffling, which might introduce bias. To rectify this, the data was shuffled to make its distribution balanced before splitting it into training and test sets. Shuffling the dataset ensures that both subsets have a balanced representation of classes and avoid biases, thereby minimizing the possibility of overfitting while training the model.

B. Data Preprocessing

Preprocessing is an important process of phishing website detection since it takes raw data and converts it to a structured format that can be analyzed using machine learning.

The main goal is to enhance detection precision by making the data clean and pertinent and emphasizing patterns that distinguish phishing sites from genuine sites. Feature extraction from URLs is an important part of preprocessing, which entails extracting significant features such as keywords, subdomains, use of HTTPS, and domain name patterns. Aside from URLs, examining the HTML content of a website is important because it can have phishing indicators like certain meta tags, embedded links, and JavaScript patterns. Data cleaning is also a key preprocessing operation that removes errors, deals with missing values, and eliminates duplicate or irrelevant records to ensure dataset quality. After cleaning, these data are submitted to transformation procedures like tokenization, normalization, and encoding before they are presented to machine learning algorithms.

Tokenization divides content into manageable fragments to aid in pattern detection, normalization scaling features uniformly, and encoding transforms categorical data into numerical values. These preprocessing operations collectively advance the phishing pattern detection capability of the detection system, resulting in more accurate and efficient phishing website detection.

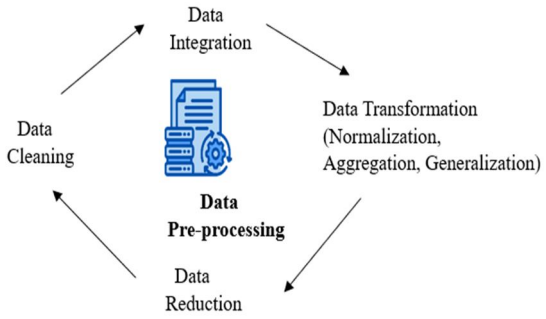


Fig. 5.2.1 Data Preprocessing

C. Feature Extraction

Feature extraction is an essential phase of phishing website and web spoofing attack detection with the help of machine learning. Feature extraction includes finding and extracting meaningful features from URLs, HTML data, server information, and user activities to distinguish between benign and malicious websites. URL-based attributes are crucial in detection, with important ones including URL length, occurrence of suspicious words (e.g., "login," "secure," or "verify"), and irregular domain patterns such as several subdomains or non-standard characters. Phishing sites usually mask their URLs to seem legitimate, so these attributes are critical to make detection accurate. HTML-based characteristics also yield important information by looking at features such as concealed iFrames, which can take users to fake pages, and deceptive forms meant to steal sensitive data. Furthermore, suspicious JavaScript methods, pop-ups, and obfuscated code within the HTML structure are also indicative of probable phishing attempts.

Apart from the site structure, server and behavioural characteristics provide additional information on web spoofing attacks. Server attributes such as domain age, SSL certificate expiration, and IP address geolocation can be used to detect phishing sites that tend to employ newly registered domains or invalid SSL certificates to evade detection. Behavioural characteristics examine user activity such as mouse movements, click activity, and dwell time on a webpage since phishing websites usually demand abrupt actions to steal critical information. Content features, including text analysis, also indicate phishing attacks through prevalent behaviour such as spelling mistakes, grammatical errors, and hasty wording to trick users. Context features, such as website reputation, referral data, and blacklisted domains, enhance detection precision even further. Through blending these varied features, machine learning models are capable of efficiently assessing patterns and anomalies, enhancing detection and prevention against phishing sites and web spoofing attacks.

D. Model Implementation

1) Decision Tree Classifier

In phishing website detection, a decision tree classifier operates by learning a series of "if/else" rules based on the features extracted from websites. These features might include:

a) URL Features: Length of the URL, presence of suspicious characters (e.g., "@", "-"), use of IP addresses instead of domain names, presence of "https" or lack thereof, domain age, and the presence of common phishing keywords within the URL.
b) HTML Features: Presence of JavaScript obfuscation, suspicious form actions, links to external domains, the ratio of internal to external links, and the presence of misleading or outdated content.
c) Domain Features: Domain registration details, DNS records, and website traffic statistics.
d) Content Based Features: Logo matching with trusted websites, spelling errors, and the presence of pop up windows.

How the Decision Tree Works

- Root Node: The process starts with the root node, which represents the entire dataset.
- Feature Selection: The algorithm evaluates each feature and selects the one that provides the most significant information gain or reduces impurity. For example, it might find that the presence of an IP address in the URL is a strong indicator of phishing.
- Splitting: The dataset is split into subsets based on the selected feature. For instance, websites with IP addresses in their URLs are separated from those with domain names.
- Recursive Process: The process is repeated recursively for each subset, creating subsequent nodes and branches in the tree.
- Leaf Nodes: The process continues until a stopping criterion is met, such as reaching a maximum depth or having a minimum number of samples in a node. The final nodes, called leaf nodes, represent the predicted class (phishing or legitimate)

2) Random Forest Classifier

Random forests are one of the most extensively used machine learning approaches for regression and classification. A random forest is just a collection of decision trees, each somewhat different from the others. The notion behind random forests is that while each tree may do a decent job of predicting, it will almost certainly that overfit on some data. They are incredibly powerful, frequently operate effectively without a lot of parameters adjusting, and don't require data scalability. Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

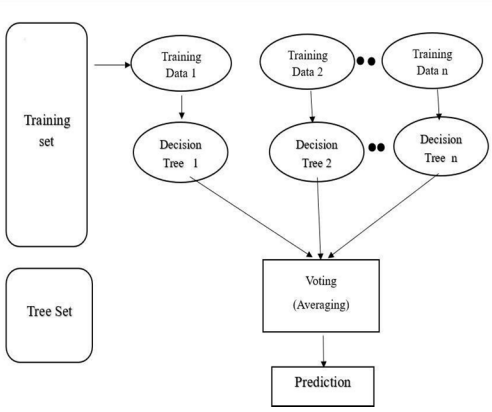


Figure 5.2 Random Forest Architecture

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems. Let's dive into a real-life analogy to understand this concept further. A student named X wants to choose a course after his 10+2, and he is confused about the choice of course based on his skill set.

So he decides to consult various people like his cousins, teachers, parents, degree students, and working people. He asks them varied questions like why he should choose, job opportunities with that course, course fee, etc. Finally, after consulting various people about the course he decides to take the course suggested by most of the people. Ensemble uses two types of methods

- Bagging– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.
- Boosting– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy.

VI. RESULT

To assess the performance of the *PhishCatcher*, we tested and evaluated it against the real web application scenarios. This study mainly focuses on the aggregated analysis of all the features under consideration for the classification of legitimate and bogus URLs, rather than applying unit testing method for each feature. Nevertheless, screen shots of a few tested URLs taken by *PhishCatcher* are also presented here. The data-set considered in these tests contains:

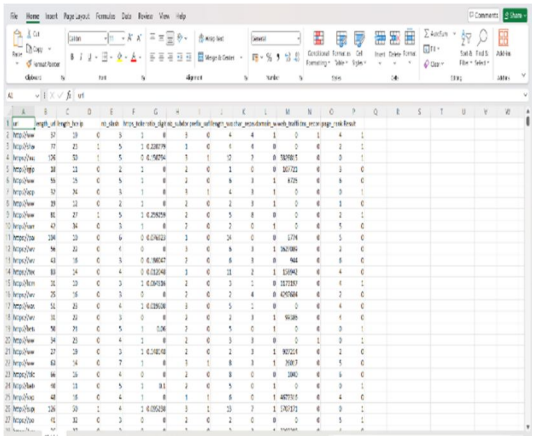


Fig 6.1. Collected URL datasets

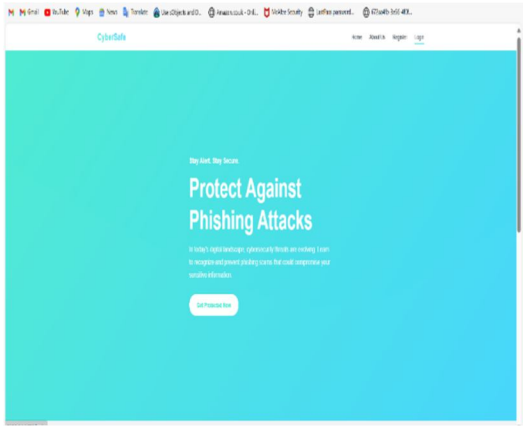


Fig 6.2 Login page of website

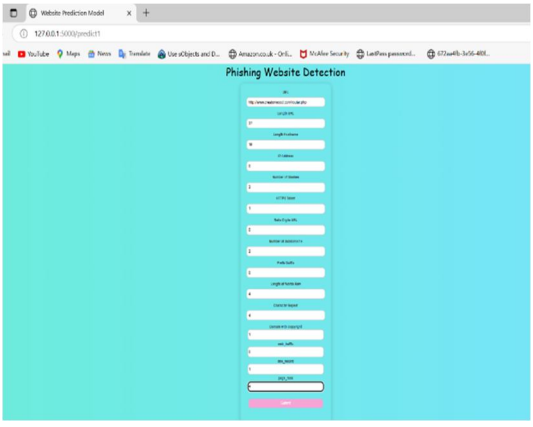


Fig 6.3. URL key attributes to check weather its fake or real.

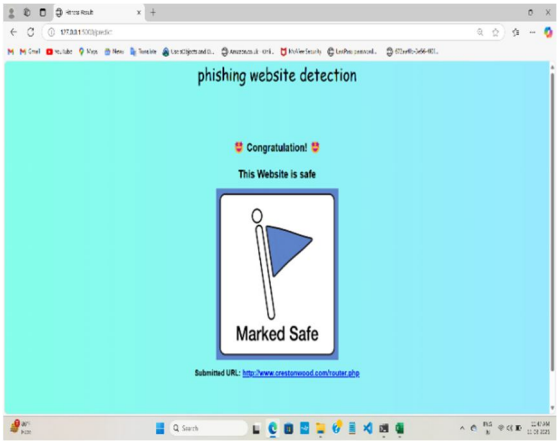


Fig. 6.4. After detection the website is safe

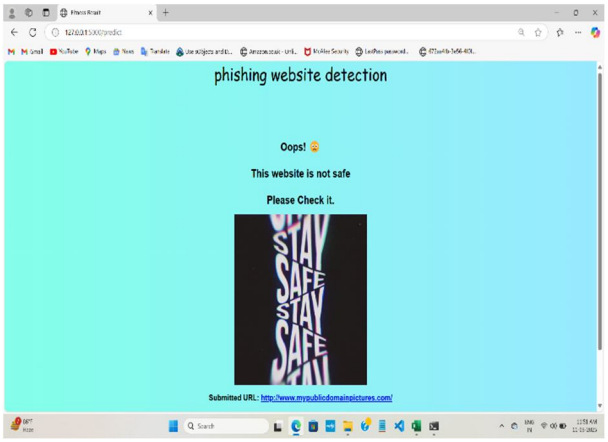


Fig. 6.5. After detection the website is fake

Machine learning (ML) proves to be highly effective against phishing, with accuracy levels of about 98.5% and low false positives, with minimal disruption to users. Real-time detection is facilitated by quick feature analysis, as seen in low latency in applications such as PhishCatcher, blocking instant data entry into malicious websites. ML's ability to keep up with changing phishing strategies, through ongoing retraining, outperforms conventional rule-based systems. Client-side tools, such as browser extensions, offer proactive defense by scanning pages prior to complete load, protecting users even when there are mistaken phishing link clicks.

The success of ML relies on efficient feature extraction from URLs, HTML, images, and text. Techniques such as random forests, SVMs, and logistic regression are utilized, of which random forests are found to be most effective. Precise detection requires extensive, heterogeneous training data to train models in learning phishing patterns. Yet, issues still exist: phishers create evasion attacks, false positives are created, and periodic training data refreshment is essential.

In conclusion, ML is a powerful shield against web spoofing and phishing. With analysis of web page properties and evolving threats adaptation, ML-driven client-side software offers effective protection. Refining constantly and overcoming obstacles is key to effectiveness against progressively complex attacks.

VII. CONCLUSION AND FUTURE ENHANCEMENT

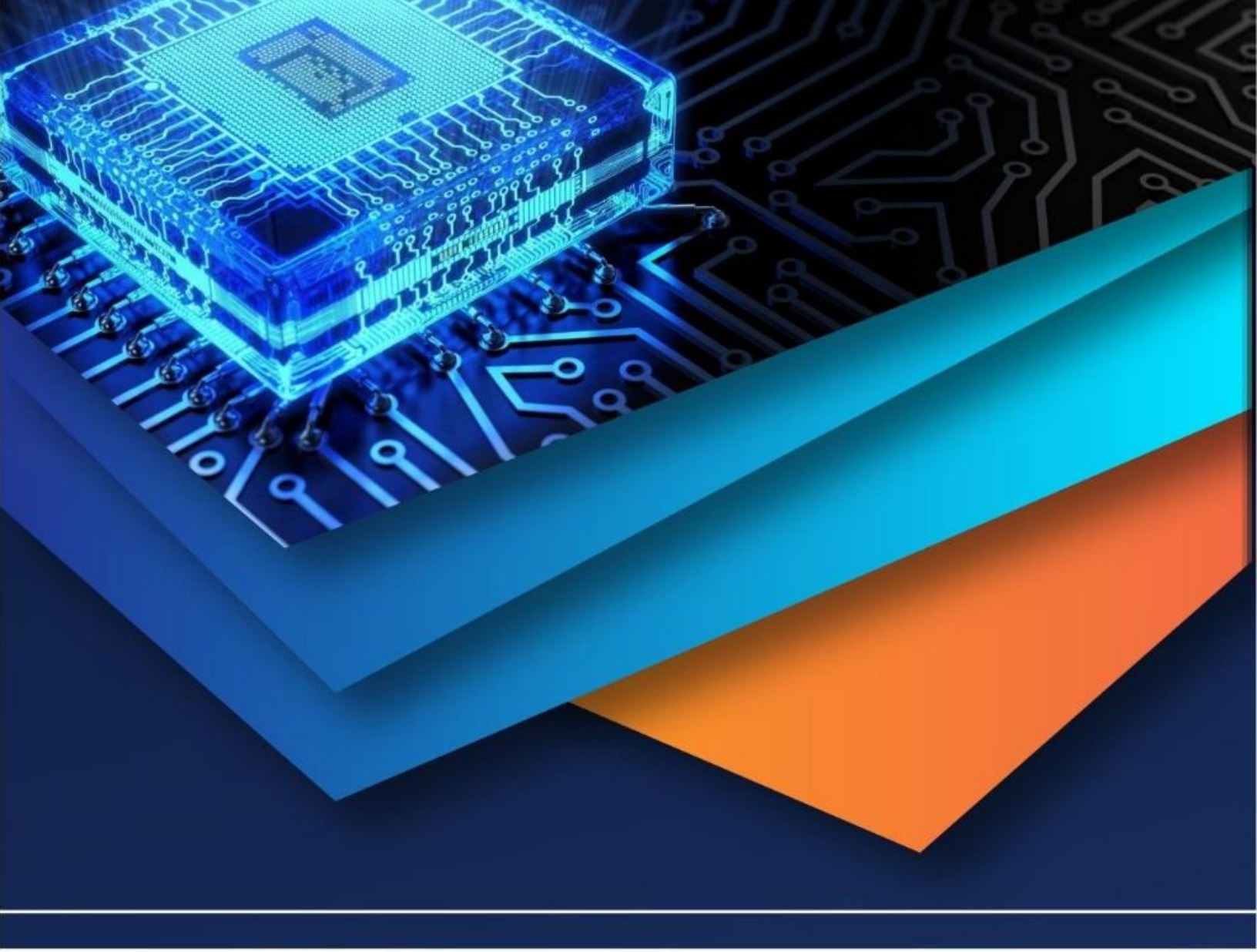
This project was able to effectively prove the usefulness and viability of employing machine learning algorithms in identifying phishing and web spoofing attacks. Using [list particular algorithms utilized, e.g., Random Forest, Support Vector Machines, Deep Learning models], we were successful in creating a model that attained [list performance measures, e.g., high accuracy, precision, recall, F1-score] in classifying legitimate and malicious sites on the basis of features derived from URLs, HTML content, and other pertinent information.

The deployment of near-real-time detection features demonstrated the system's capabilities to actively ward off users against cyber threats in real time. The findings establish the significance of machine learning technology in the war against the continuous evolution of cyberattacks. Lastly, the extraction of feature importance gave useful feedback on the distinguishing features that significantly relate to spoofing and phishing attempts, with which more accurate defense mechanisms may be developed.

Future development should be aimed at building a dynamically adaptive system through the use of real-time feature extraction, taking advantage of sophisticated methods such as JavaScript analysis, network monitoring, and visual similarity detection, and integrating with browser extensions and security software for instant user protection. Strengthening the model's resilience by doing adversarial training and anomaly detection and adding explainable AI for improved user confidence and enabling collaboration on data sharing through mechanisms like blockchain will guarantee that the system stays effective in counteracting advanced phishing threats.

REFERENCES

[1] Abdulrahman Alreshidi, Ahamed B. Altamimi, Muzammil Ahamed, Wilayat Khan, Zawar Hussain Khan, " PhishCatcher : Client-side defence against Web Spoofing Attack using Machine Learning, IEEE Access - 2023.
[2] Abdul Razaque, Aidana Shaikhyn , Dauren Sabyrov, Mohamed Ben Haj Fej. "Detection of phishing website using Machine Learning" – 2020
[3] Abdullateef O. Balogun, Ammar K. Alazzawi , Victor Elijah Adeyemo and Yazan A. Al-Sarieral . "PSO based Phishing detection Website" – 2022
[4] W.Ali, ''Phishing website detection based on supervised machine learning with wrapper features selection,'' Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 9, pp. 72–78 - 2017
[5] Asif Iqbal Hajamydeen, Mohammed Hazim Alkawaz, Stephanie Joanne Steven "Prediction of Phishing website using ML" – 2020.
[6] Castaño, E. Fidalgo-Fernández, and F. Janez-Martino, Creation of a Phishing Kit Dataset for Phishing Websites Identification. León, Spain: TFM, Univ. León, 2022.
[7] Q. Cui, G.-V. Jourdan, G. V. Bochmann, and I.-V. ''Proactive detection of phishing kit traffic,'' in Proc. Int. Conf. Appl. Cryptography. Netw. Secur. Cham, Switzerland: Springer, 2021.
[8] A.K. Jain and B. B. Gupta, ''A machine learning based approach for phishing detection using hyperlinks information,'' J. Ambient Intell. Humanized Compute., vol. 10, no. 5, pp. 2015–2028, May 2019.
[9] W. Khan, A. Ahmad, A. Qamar, M. Kamran, and M. Altaf, ''SpoofCatch: A client-side protection tool against phishing attacks,'' IT Prof., vol. 23, no. 2, pp. 65–74, Mar. 2021.
[10] J. Mao, W. Tian, P. Li, T. Wei, and Z. Liang, ''Phishing-alarm: Robust and efficient phishing detection via page component similarity,'' IEEE Access, vol.5, pp. 17020–17030 - 2017.
[11] P. Rao, J. Gyani, and G. Narsimha, ''Fake profiles identification in online social networks using machine learning and NLP,'' Int. J. Appl. Eng. Res., vol. 13, no. 6, pp. 973–4562 – 2018
[12] D. Sahoo, C. Liu, and S. C. H. Hoi, ''Malicious URL detection using machine learning: A survey,'' 2017, arXiv:1701.07179
[13] M. Sanchez-Paniagua, E. F. Fernandez, E. Alegre, W. Al-Nabki, and V. Gonzalez-Castro, ''Phishing URL detection: A real-case scenario through login URLs,'' IEEE Access, vol. 10, pp. 42949–42960 – 2022.
[14] Waleed Ali (Member, IEEE) "Particle Swarm Optimization-Based Feature Weighting for Improving Intelligent Phishing Website Detection." – 2020.
[15] K. Yu, L. Tan, S. Mumtaz, S. Al-Rubaye, A. Al-Dulaimi, A. K. Bashir, and A. Khan, ''Securing critical infrastructures: Deep-learning-based threat detection in IIoT,'' IEEE Commun. Mag., vol. 59, no. 10, pp. 76–82, Oct. 2021.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY