Machine Learning Based Approach for Phishing Attacks Detection and Prevention

Akshay M J, Pratheeksha N Hampole, Sowparnika K H, Sushmitha H S, T P Keerthi, Pavan M
Department of Information Science and Engineering
Jawaharlal Nehru New College of Engineering, Shivamogga

Abstract—**Phishing attacks remain a significant threat in the cybersecurity landscape, targeting individuals and organizations to steal sensitive information such as usernames, passwords, financial data, and confidential business records. These attacks are typically carried out through deceptive emails, websites, and messages that appear legitimate but aim to trick users into disclosing personal information. Given the sophistication and increasing frequency of phishing schemes, there is an urgent need for automated detection and prevention systems that can quickly identify and mitigate such threats. This work focuses on detecting and preventing phishing attacks using machine learning, specifically leveraging the XGBoost (Extreme Gradient Boosting) algorithm. XGBoost is a state-of-the-art supervised learning model known for its high predictive accuracy and ability to handle imbalanced datasets, which is often the case in phishing detection tasks. The proposed work explores the use of URL-based features, such as domain registration details, URL length, the presence of special characters, and other textual elements, as input to the XGBoost classifier.**

Keywords— Phishing, Legitimate, Classification, Feature extracton, XGBoost.

I. INTRODUCTION

Phishing attacks represent a significant cybersecurity challenge, leveraging social engineering tactics to manipulate individuals into divulging sensitive information. As these attacks become increasingly sophisticated, traditional detection methods often fall short, highlighting the need for advanced solutions. Machine learning (ML) has emerged as a powerful tool in the fight against phishing, offering innovative techniques for identifying and mitigating these threats. Machine learning (ML) offers a promising solution to enhance phishing detection and prevention efforts. By leveraging algorithms that can learn from data, ML models can identify patterns and anomalies that signify phishing attempts with greater accuracy and speed than conventional methods. Techniques such as supervised learning, unsupervised learning, and deep learning allow for the analysis of vast datasets, including email content, URLs, and user behavior. Thus incorporating ML into phishing attack prevention strategies not only improves the speed and accuracy of threat identification but also enables adaptive responses to evolving tactics used by attackers. This approach allows organizations to stay one step ahead of cybercriminals. XGBoost (eXtreme Gradient Boosting) is a powerful and efficient machine learning library designed for gradient boosting, a technique that combines the predictions of weak learners (typically decision trees) to create a strong predictive model. XGBoost implements an optimized gradient boosting algorithm based on decision trees, making it suitable for both classification and regression tasks. It includes advanced regularization, which helps prevent overfitting and enhances model generalization. Leveraging ML techniques allows for the analysis of large datasets, including email content, URLs, and website behavior, to identify and prevent phishing attempts more effectively than rule- based systems.

II. RELATED WORKS

This section discusses the papers and methods related to the detection and prevention of phishing attacks.

The research focuses on using machine learning (ML) models to identify and prevent phishing attacks, which are a significant cybersecurity threat [1]. It presents a comparative analysis of various ML techniques, including Support Vector Machines (SVM) and XGBoost, for detecting phishing websites. The study evaluates these models using performance metrics such as accuracy and precision. The XGBoost algorithm is highlighted for its superior accuracy, precision, and computational efficiency across multiple datasets. Through rigorous experimental analysis, the findings demonstrate that XGBoost outperforms the Support Vector Machines algorithm.

The research presents a novel approach that combines feature selection and hyperparameter tuning for the XGBoost algorithm using a modified genetic algorithm (GA) [2]. It addresses challenges in spam detection by reducing the dimensionality of large datasets while maintaining or improving classification performance. The proposed method achieves significant spam classification accuracy and a high geometric mean using less than 10% of the total feature space. The approach was validated against datasets that include web spam, which exploits social engineering to lure privileged users into logging into deceptive services.

The research explores the increasing prevalence and sophistication of spear-phishing attacks, which are highly targeted and context-specific compared to regular phishing [3]. It provides an extensive review of the literature on phishing and spear-phishing, differentiating between the two types of attacks and analyzing their processes. The study highlights the growing reliance on social engineering (SE) as an attack vector, focusing on targeting individuals rather than systems. It examines a real-world, advanced spear-phishing campaign targeting white-collar workers in 32 countries, illustrating the sophisticated tactics used by attackers, such as creating fake companies and job postings to lure victims into providing sensitive information.

The research presents a novel approach for detecting phishing URLs, particularly those generated by AI systems like DeepPhish [4]. The system is designed to detect both AI-generated and human-crafted phishing URLs. PhishHaven introduces several innovative techniques, including URL HTML Encoding for on-the-fly classification and a URL Hit approach to handle shortened URLs. The ensemble-based machine learning models are executed in parallel using a multi-threading approach, enabling real-time detection.

The paper introduces a novel solution for detecting phishing attacks by analyzing the visual similarity between web pages [5]. The authors argue that traditional phishing detection

methods, which rely on URL analysis or blacklists, are insufficient, as attackers can easily modify these features to evade detection.

This work focuses on leveraging machine learning techniques to detect phishing websites, which aim to steal sensitive user information through deceptive means [6]. The proposed approach employs supervised machine learning algorithms—Random Forest and Decision Tree—to classify URLs as either phishing or legitimate. The study utilizes a dataset of 10,000 URLs (5,000 phishing and 5,000 legitimate), with features extracted from three categories: URL-based attributes (e.g., length, depth), domain-based attributes (e.g., DNS records, domain age), and HTML/JavaScript-based attributes (e.g., iframe redirection, right-click disablement).

The research explores innovative approaches to phishing website detection using multimodal learning techniques [7]. It introduces an Adversarial Website Generation (AWG) framework, leveraging Generative Adversarial Networks (GANs) and transfer-based black-box attacks to simulate real-world phishing attacks. The study assesses 15 learning-based models, including machine learning (ML), deep learning (DL), ensemble learning (EL), and multimodal models (MM), focusing on their resistance to adversarial examples (AEs). It also proposes defense strategies such as adversarial training to enhance model robustness against phishing and adversarial websites.

The work presents a novel dataset specifically designed to address the increasing sophistication and prevalence of phishing attacks [8]. The authors emphasize that phishing kits, which are prepackaged software used to create phishing websites, pose a significant threat due to their ease of use and distribution. These kits allow attackers with minimal technical knowledge to launch effective phishing campaigns. The PhiKitA dataset aims to enhance the identification and understanding of phishing websites by providing a comprehensive collection of phishing kits and associated metadata.

The work addresses the challenge of detecting social semantic attacks, a subset of social engineering attacks [9]. These attacks exploit human behavioral and psychological vulnerabilities by creating deceptive elements such as URLs or webpages that mimic legitimate ones. The study focuses on detecting malicious URLs associated with four types of attacks: phishing, spamming, defacement, and malware.

This research investigates the issue of phishing attacks and evaluates the effectiveness of various machine learning algorithms in detecting them [10]. It systematically analyzes five algorithms— Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), Naive Bayes, and Extreme Gradient Boosting (XGBoost)—using a large dataset of URLs. The paper likely explores and evaluates various machine learning techniques for detecting phishing attacks. Phishing, a social engineering technique, tricks users into divulging sensitive information by mimicking trusted entities.

This work presents a novel hybrid two-level framework designed to enhance the detection of phishing websites by optimizing feature selection and XGBoost model performance [11]. The framework integrates advanced feature selection techniques to identify the most relevant attributes, reducing noise and computational overhead. Subsequently, it applies an iterative hyperparameter tuning process to fine-tune the XGBoost classifier for improved accuracy and robustness.

This research presents a multi-agent intelligent system for detecting and preventing phishing attacks and malicious scripts using machine learning [12]. It incorporates four agents: a monitoring agent for URL extraction, two decision-making agents utilizing classifiers (SVM and ANN), and an action-performing agent to block malicious pages or scripts. The system employs the Extensible Messaging and Presence Protocol (XMPP) for agent communication and integrates features like URL length, IP addresses, and DNS records for analysis.

This research explores the persistent cybersecurity threat of phishing attacks, detailing their various forms, including spear phishing, vishing, and smishing [13]. It explains how attackers exploit social engineering techniques to manipulate users into divulging personal information or installing malware. Specific attention is given to emerging threats like ransomware, banking trojans, and cryptojacking. The article also outlines the stages of phishing attacks, from preparation to execution.

This work investigates a novel adversarial hiding approach aimed at evading phishing detection mechanisms within the Ethereum blockchain ecosystem [14]. It explores how malicious actors exploit Ethereum's decentralized infrastructure and smart contract features to conceal phishing activities from detection systems. The proposed approach leverages adversarial techniques, such as obfuscation and manipulation of transaction patterns, to bypass traditional and machine learning-based phishing detection algorithms.

This study provides a comprehensive benchmarking and evaluation of phishing detection techniques, addressing their effectiveness in meeting modern security requirements [15]. It surveys state-of-the-art methodologies, highlighting their strengths, limitations, and practical applicability in real-world scenarios. By analyzing a wide range of phishing detection algorithms and systems, the study identifies critical gaps in existing research and offers insights into improving detection accuracy, scalability, and robustness.

This study explores the vulnerabilities of machine learning-based web phishing classifiers to evasion attacks, where adversaries manipulate inputs to bypass detection systems [16]. It provides an in-depth analysis of various evasion attack strategies, such as feature manipulation, adversarial examples, and mimicry techniques, which undermine the effectiveness of these classifiers. The study also reviews state-of-the-art defense mechanisms, including adversarial training, feature hardening, and robust model architectures, aimed at enhancing the resilience of phishing detection systems.

This work addresses the challenge of detecting new forms of phishing attacks that evade traditional filters [17]. It introduces SAFE-PC, a machine learning system for phishing email detection. SAFE-PC extracts and processes email features, leveraging techniques like Natural Language Processing and Named Entity Recognition. It uses an ensemble classifier for high detection accuracy.

The study investigates the vulnerabilities of machine learning (ML) models in detecting malicious advertisement URLs [18]. The study develops a framework leveraging lexical and webscrapped features for ML-based classification and clustering. It evaluates four ML models (Random Forest, Gradient Boost, XGBoost, and AdaBoost) for their detection accuracy and robustness against adversarial attacks, specifically the Zeroth Order Optimization (ZOO) attack.

III. PROPOSED APPROACH

The Proposed Approach mainly deals with two modules which are Feature Extraction of the URLs and Training Machine Learning Algorithms.

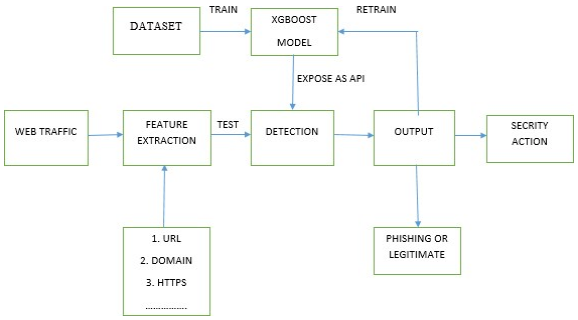

Fig. 1. Architecture of Proposed System

Figure 1 represents the architecture diagram. In this, the first algorithm is trained with base data set which is used as training data and the data which is taken from the web traffic acts as input for the feature extraction which is done mainly on three types of features URL based, domain-based, Html/JS-based features and this feature extracted data acts as testing data and this machine learning model is exposed to API and the prediction will be done and output is generated as phishing or legitimate. If it is phishing then we should block the website and if the output is legitimate then we should allow it.

A. Data Set Description

The data set consists of 88,647 URLs which is 30,647 are phishing and 58,000 are legitimate.Phishing URLs are collected from the Kaggle web source.
-All the phishing URLs are labeled as "-1".
-All the legitimate URLs are labeled as "1".

B. Feature Extraction

The Features of the data set has been extracted by using python programming language. This model is trained based on three kinds of features those are:

1. Domain-Based Features
    - DNS Record
    - Website Traffic
    - Age of Domain
    - End Period of Domain
2. HTML and JavaScript Based Features
    - IFrame Redirection
    - Status Bar Customization
    - Disabling Right Click
    - Website Forwarding
3. Address Bar Based Features
    - Domain
    - IP Address
    - "@" Symbol
    - Length
    - Depth
    - Redirection "//"
    - "HTTP/HTTPS" in Domain name
    - Using URL Shortening Services "Tiny URL"
    - Prefix or Suffix "-" in Domain

There are total of 30 features taken and the data set is shuffled and this data set is used to train the model.

Fig. 2 represents a correlation heat map that shows how the features of the data set are related to each other.

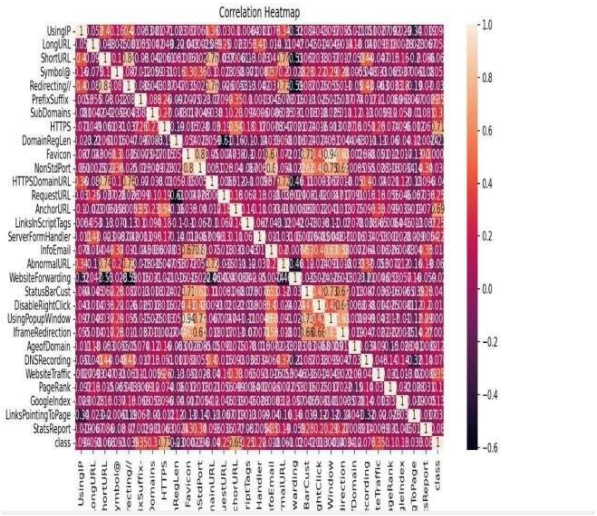

Fig. 2. Correlation Heat Map

C. Methodology

The proposed methodology begins with the collection of data which contains the information like 's' at the end of https, long URL, short URL and other related details that are required for detecting phishing in a website feature selection is then performed to identify the most relevant attributes in predicting the disease effectively, later the pre-processed data is splitted into training and testing of the data. 80% of the data is considered for training and 20%of data is considered for testing. The proposed methodology begins with the collection of data which contains the information like 's' at the end of https, long URL, short URL and other related details that are required for detecting phishing in a website, followed by data preprocessing preparing the raw data for analysis using various techniques like data completion, data noise reduction, data transformation and validation, feature selection is then performed to identify the most relevant attributes in predicting the disease effectively, later the pre- processed data is splitted into training and testing of the data. 80% of the data is considered for training and 20%of data is considered for testing. The proposed methodology begins with the collection of data which contains the information like 's' at the end of https, long URL, short URL and other related details that are required for detecting phishing in a website, followed by data preprocessing preparing the raw data for analysis using various techniques like data completion, data noise reduction, data transformation and validation, feature selection is then performed to identify the most relevant attributes in predicting the disease effectively, later the pre-processed data is splitted into training and testing of the data.

80% of the data is considered for training and 20%of data is considered for testing. The proposed methodology begins with the collection of data which contains the information like 's' at the end of https, long URL, short URL and other related details that are required for detecting phishing in a website, followed by data preprocessing preparing the raw data for analysis using various techniques like data completion, data noise reduction, data transformation and validation, feature selection is then performed to identify the most relevant attributes in predicting the disease effectively, later the pre- processed data is splitted into training and testing of the data. 80% of the data is considered for training and 20%of data is considered for testing.

The objective function in XGBoost is like a scorecard that helps the algorithm decide how good a model is. This scorecard has two main parts:

1. **Loss Function:** Measures how well the model's predictions match the actual data. This is a crucial part of the objective function, as it helps the algorithm understand how to improve its predictions.
2. **Regularization Term:** This part prevents the model from becoming too complex and overfitting (like memorizing the training data rather than learning to generalize). It's like adding a rule that you can't use too many darts to hit the bullseye; you need to hit it in fewer, well-placed throws.

•Loss Function = Measures prediction error.
•Regularization Term = Keeps the model simple to avoid overfitting.
•Objective Function = The combination of these two to guide the model building process.

IV. IMPLEMENTATION AND RESULTS

Python is a versatile and powerful programming language that offers a wide range of tools and libraries for various purposes, making it a popular choice among developers, data scientists, and researchers. Following are the libraries employed in our project. Pandas, NumPy, SciKit-Learn, Flask, re(Regular Expression), urllib.parse, socket, ipaddress, TfidVectorizer, Train_test_split are the most important tools that played majority role in implementation of the XGBOOST model.

With the help of these tools,it has become possible to classify features like longURL, shortURL, using IP or not, presense or absence of symbol @, number of redirecting //, Subdomains occurences, policies of Domain Registration length, presence of anchor URL, Website forwardings and many more into phishing or legitimate for the model development.

The feature Extracted data set is taken and it is divided into training and testing data with a ratio of 80-20 this training data is used to train the XGBOOST model and testing data is used to test and find the accuracy of the algorithm.

The diagram represents a workflow for a system involving user login, data extraction, and prediction phases. The process begins with the user registering or logging in, providing their login ID and password. The system verifies the credentials by looking them up in a database, allowing successful users to proceed. Verified users are redirected to a prediction page, where they can enter a URL for analysis. The system extracts feature from the URL and performs predictions, providing the results. The final output of the prediction is displayed to the user as the last step.

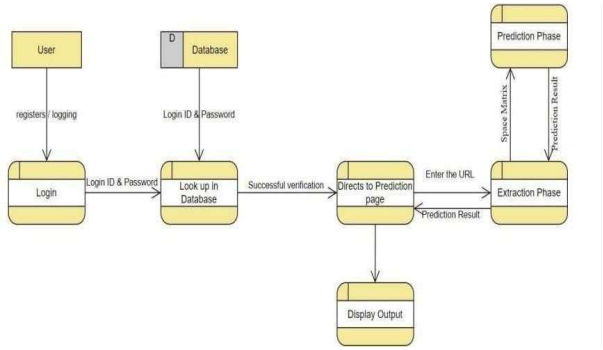

Fig. 3. Implimentation of antiphishing model

To develop a robust classification model, we first fine-tune the model's settings, known as classifier parameters, to achieve the best possible performance.
Implementing a phishing attack detection system using the XGBoost (Extreme Gradient Boosting) algorithm involves several steps, from data collection and preprocessing to model evaluation and deployment. Data Collection First, you need to collect a dataset that contains information about phishing and legitimate websites. You can use publicly available datasets or create your own dataset.

The data needs to be cleaned and transformed into a format suitable for training the model. Identify and handle missing values by either removing or imputing them with appropriate strategies. Convert raw website features (URLs, HTML content, etc.) into meaningful numerical features. Divide your dataset into two sets: training and testing (commonly 80% for training and 20% for testing). Now , train the XGBoost model. Install the xgboost library if you haven't already and then implement the xgboost classifier. XGBoost has several hyperparameters that can be tuned for better performance. You can use grid search or random search to find the optimal hyperparameters. After training the model, evaluate its performance on the test set using various metrics such as Accuracy: The ratio of correctly predicted instances to the total instances. Precision is important in finding imbalanced datasets to evaluate the model's effectiveness in detecting phishing URLs. Once the model is trained and evaluated, you can deploy it as part of a larger phishing detection system.

Confusion Matrix:

Confusion matrix(CM) is a graphical summary of the correct predictions and incorrect predictions that is made by a classifier that can be used to determine the performance. In abstract terms, the CM is as shown in the following figure.

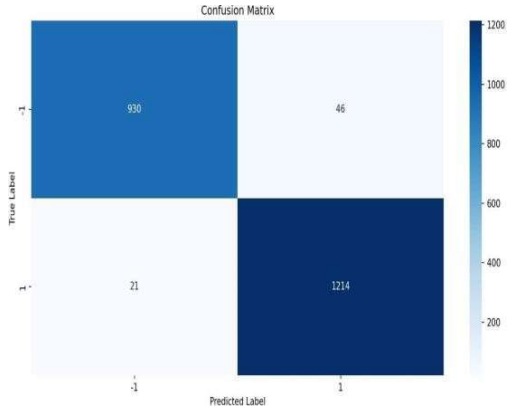

Fig 4. Shows confusion matrix

V. MODEL DEPLOYMENT

From the above classification report and the confusion matrix, it is clearly shown that XGBOOST is having a high accuracy than other ML algorithms. so, this algorithm is stored by using pickle for deployment. The model is deployed as a webpage with the help of Flask API. This webpage contains a textbox and a submit button which is developed using Hyper Text Markup Language (HTML).

Whenever we enter a URL and click on a submit button then this URL will be processed by the model and returns a value as 1 or -1. If the returned value is '1' then the output is displayed as "Legitimate" and if the returned value is '-1' the output is displayed as "Phishing".

Fig. 5. Shows Admin Login page

Fig. 6. Shows About Us page

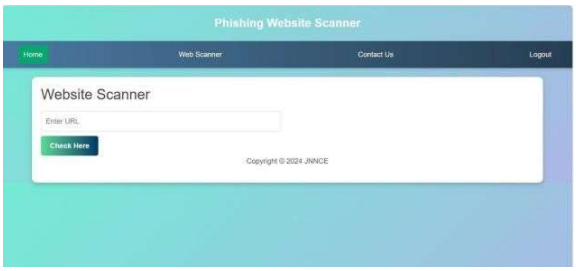

Fig.7. Shows Web Scanner Page

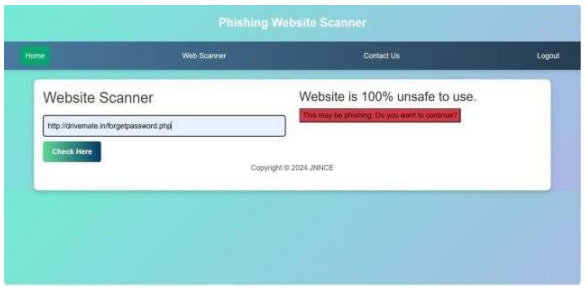

Fig. 8. Shows detection of phishing website

Fig. 9. Shows detection of legitimate website

Admin Login page where it allows the user to enter the username and username. The Home page where it consists of description about website scanner. The website predictor page which allows users to enter relevant data.The users input there data and press "check here" to determine whether the entered website is phishing or legitimate. website predictor page mentioned as "Web Scanner" which allows users to enter relevant data. The users input there data and press "check here" to determine whether the entered website is phishing or legitimate. The detection of legitimate website, after the user have entered the URL in predictor page it will predict legitimate website based on the features. The detection of phishing website, after the user have entered the URL in predictor page it will predict phishing website based on the features and alerts the user that the website is unsafe and asks the user whether they want to continue although it is unsafe. The contact us page, where the admin details is stored like admin name, email and contact number through which the user can contact the admin.

VI. CONCLUSION

This paper helps to develop a model by using Machine Learning which is used to detect the phishing URL's and warn the user in advance. The features of the URL is extracted which is entered by the user in the respective field and this acts as input data for the Machine learning model. The model process this and gives the output as to whether it is phishing or legitimate. The algorithms that are used to build this model is XGBOOST. After training the accuracy of the algorithm is 97.0%.

VII. REFERENCES

[1]. N. Ghatasheh, I. Altaharwa and K. Aldebei, "Modified Genetic Algorithm for Feature Selection and Hyper Parameter Optimization: Case of XGBoost in Spam Prediction," in IEEE Access, vol. 10, pp. 84365-84383, 2022.

[2]. M. Patil, N. Shivsharan, Y. Naik, H. Yeram and A. Gawade, "Enhancing Cybersecurity: A Comprehensive Analysis of Machine Learning Techniques in Detecting and Preventing Phishing Attacks with a Focus on Xgboost Algorithm," 2024 International Conference on Intelligent Systems for Cybersecurity (ISCS), Gurugram, India, 2024.

[3]. L. Allodi, T. Chotza,E.Panina and N. Zannone, "The Need for New Antiphishing Measures Against Spear-Phishing Attacks," in IEEE Security & Privacy, vol. 18, no. 2, pp. 23-34, March-April 2020

[4]. M. Sameen, K. Han and S. O. Hwang, "PhishHaven-An Efficient Real-Time AI Phishing URLs Detection System," in IEEE Access, vol. 8, pp. 83425-83443, 2020

[5]. J. Mao, W. Tian, P. Li, T. Wei and Z. Liang, "Phishing-Alarm: Robust and Efficient Phishing Detection via Page Component Similarity," in IEEE Access, vol. 5, pp. 17020-17030, 2017

[6]. A. Mandadi, S. Boppana, V. Ravella and R. Kavitha, "Phishing Website Detection Using Machine Learning," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India.

[7]. P. T. Duy, V. Q. Minh, B. T. H. Dang, N. D. H. Son, N. H Quyen and V. -H. Pham, "A Study on Adversarial Sample Resistance and Defense Mechanism for Multimodal Learning-Based Phishing Website Detection," in IEEE Access, vol. 12, pp. 137805-137824, 2024

[8]. F. Castaño, E. F. Fernañdez, R. Alaiz-Rodríguez and E. Alegre, "PhiKitA: Phishing Kit Attacks Dataset for Phishing Websites Identification," in IEEE Access, vol. 11, pp. 40779-40789, 2023

[9]. M. Almousa and M. Anwar, "A URL-Based Social Semantic Attacks Detection With Character- Aware Language Model," in IEEE Access, vol. 11, pp. 10654-10663, 2023,

[10]. H. Shirazi, S. R.Muramudalige, I. Ray, A. P. Jayasumana and H. Wang, "Adversarial Autoencoder Data Synthesis for Enhancing Machine Learning-Based Phishing Detection Algorithms," in IEEE Transactions on Services Computing, vol. 16, no. 4, pp. 2411-2422, 1 July-Aug. 2023

[11]. L. Jovanovic et al., "Improving PhishingWebsite Detection using a Hybrid Two-level Framework for Feature Selection and XGBoost Tuning," in Journal of Web Engineering, vol. 22, no. 3, pp. 543- 574, May 2023.

[12]. N. Megha, K. R. Remesh Babu and E. Sherly, "An Intelligent System for Phishing Attack Detection and Prevention," 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2019, pp. 1577-1582.

[13]. M. A. Ivanov, B. V. Kliuchnikova, I. V. Chugunkov and A.M. Plaksina, "Phishing Attacks and Protection Against Them," 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), St. Petersburg, Moscow, Russia.

[14]. H. Wen, J. Fang, J. Wu and Z. Zheng, "Hide and Seek: An Adversarial Hiding Approach Against Phishing Detection on Ethereum," in IEEE Transactions on Computational Social Systems, vol. 10, no. 6, pp. 3512-3523, Dec. 2023.

[15]. A. El Aassal, S. Baki, A. Das and R. M. Verma, "An In-Depth Benchmarking and Evaluation of Phishing Detection Research for Security Needs," in IEEE Access, vol. 8, pp. 22170-22192, 2020.

[16]. M. J. Pillai, S. Remya, V. Devika, S. Ramasubbareddy and Y. Cho, "Evasion Attacks and Defense Mechanisms for Machine Learning-Based Web Phishing Classifiers," in IEEE Access, vol. 12, pp. 19375-19387, 2024, doi: 10.1109/ACCESS.2023.

[17]. C.N.Gutierrez et al.,"Learning from the Ones that Got Away: Detecting New Forms of Phishing Attacks," in IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 6, pp. 988-1001, 1 Nov.-Dec. 2018.

[18]. E.Nowroozi, Abhishek, M. Mohammadi and M. Conti, "An Adversarial Attack Analysis on Malicious Advertisement URL Detection Framework," in IEEE Transactions on Network and Service Management, vol. 20, no. 2, pp. 1332-1344, June 2023.