Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
e-ISSN: 2788-7421 p-ISSN: 2958-0544

A Machine Learning Algorithms for Detecting Phishing Websites: A Comparative Study

Dr. Mohammed A. Taha[1]*, Haider D. A. Jabar[2], Widad K. Mohammed[3]

[1] Ministry of Education, Babylon Education Directorates, Babylon, Postcode, Iraq
[2] Ministry of Education, Baghdad, Postcode, Iraq
[3] Ministry of Education, Baghdad, Postcode, Iraq

*Corresponding Author: Dr. Mohammed A. Taha

DOI: https://doi.org/10.52866/ijcsm.2024.05.03.015
Received February 2024; Accepted April 2024; Available online July2024

ABSTRACT: Phishing website attacks are a type of cyber-attack in which perpetrators create fraudulent websites that mimic legitimate platforms, such as online banking or social media, with the intent of tricking unsuspecting users into divulging sensitive information. This includes passwords, credit card details, usernames, and other personal data. These phishing websites are designed to look authentic and often employ various techniques, such as URL spoofing, social engineering, and email or text message phishing, to lure victims into revealing their confidential information. Web apps are growing increasingly complex and difficult to identify at first glance, especially when they use encryption and obfuscation techniques. In order to effectively detect and stop phishing web applications from being uploaded to the server in real-time, machine learning must be developed. In addition to including analyses for the machine learning algorithms for identifying web application-based assaults, the study calibrates fresh analyses by executing machine learning algorithms and confirming the findings. The study uses unique and categorized results from a machine learning dataset. As per the outcomes obtained from experimental and comparative analyses of the applied classification algorithms, the random forest model demonstrated the highest accuracy, achieving an impressive rate of 96.89%, followed by the decision tree model at 94.57%, and Extreme Gradient Boosting (XG).

Keywords: Decision Tree, Logistic Regression, Phishing, Random Forest, XGBoost

1. INTRODUCTION

Phishing attacks are fraudulent attempts in which cybercriminals create deceptive communications, such as emails, messages, or websites, that appear to come from reputable sources. These attacks have become a major problem for companies, with losses totaling around $100 billion annually. Furthermore, they are on the rise, with a 200% increase from previous years. The current solutions available to combat these attacks are not effective, and there is a pressing need for new and innovative methods to protect both companies and individuals [1]. With the increasing reliance on computerized financial activities and the decrease in cash transactions, cybercriminals are exploiting this trend by using phishing techniques to fraudulently obtain sensitive financial information from unsuspecting victims [2]. Criminal organizations have transitioned their tactics from exploiting technical system vulnerabilities to exploiting human vulnerabilities, such as the lack of ability to discern between genuine and fraudulent online resources, such as emails and websites. Therefore, it is crucial to develop effective solutions to mitigate these issues [3]. Numerous elements of daily life, including social media, online banking, e-commerce, and other activities, have moved to the internet due to the fast development of worldwide networking and communication technologies. The open, private, and uncontrolled character of the Internet, however, also creates a favorable environment for cyberattacks, posing serious security dangers to networks as well as to common computer users, even seasoned ones. It is hard to completely prevent individuals from suffering from phishing scams, even though user care and skill are essential [4]. A phishing website is a deceptive and fraudulent website that aims to dupe and manipulate users into divulging confidential information. These websites are usually disguised as legitimate websites or emails and often contain fake login pages or other forms designed to steal information from unsuspecting users. Phishing websites typically use social engineering tactics to lure users into providing their sensitive information, such as posing as a trustworthy institution like a bank, social media platform, or e-commerce site. Once a user enters their information into the fake website, attackers can then use this information to steal money, identities, or commit other forms of fraud [5]. To prevent becoming prey to phishing websites, exercising caution when entering personal information online is critical. Verifying the website's URL, searching for security indicators like HTTPS and a lock icon, and abstaining from clicking on links in suspicious emails are all necessary measures.

*Corresponding author: cs.19.54@grad.uotechnology.edu.iq

Furthermore, employing anti-phishing software and maintaining the most recent security updates for your computer and browser is highly recommended [6].

The following recent works represent a variety of approaches that utilize machine learning (ML), behavioral analysis, and novel techniques to enhance the identification and prevention of phishing attempts.

MARIA S. & K. HAN in [7] define PhishHaven, which is an ensemble machine learning-based system designed to identify both AI-generated and human-crafted phishing URLs. This marked a notable advancement in phishing attack detection. PhishHaven employs a multi-threading approach to execute the classification parallelly, thus leading to real-time detection.

Detecting phishing attacks typically demands substantial processing power due to the use of a multitude of features, rendering it impractical for resource-constrained devices. To tackle this challenge, [8] developed a phishing detection method that relies on just nine lexical features, ensuring effective identification of phishing attacks. They used the ISCXURL-2016 dataset, which includes 11,964 instances comprising legitimate and phishing URLs.

In another research, [9] introduces PhiUSIIL, a framework for detecting phishing URLs that relies on a Similarity Index and Incremental Learning. The framework effectively identifies various visual similarity-based attacks, such as zero-width characters, homograph, punycode, homophone, bit squatting, and combosquatting attacks using the s imilarity index. By adopting incremental learning, PhiUSIIL continuously updates its knowledge base with new data, while diverse security profiles cater to different security needs. The framework extracts URL and HTML features, generating the PhiUSIIL phishing URL dataset, which includes 134,850 legitimate and 100,945 phishing URLs. Through extensive experiments using this dataset, PhiUSIIL significantly improves detection accuracy.

Additionally, [10] introduces a machine learning approach to identify phishing websites within FFSNs, utilizing an innovative set of 56 features. In contrast to previous methodologies, this approach achieves heightened accuracy, faster detection times, and integrates a wide range of features to fortify evasion-resilient detection capabilities. Evaluating feature effectiveness in binary and multi-class classification tasks, employing both traditional and deep learning machine learning algorithms, the proposed approach achieves 98.42% accuracy for binary classification and 97.81% for multi-class classification. Findings highlight temporal and DNS-based features as robust predictors, while network and host-related features exhibit comparatively weaker predictive power. This method marks a significant stride towards monitoring fundamental elements within FFSNs, with the ultimate goal of dismantling the entire phishing ecosystem.

To counter the threat to financial security within the blockchain, [11] introduced the LBPS model (LSTM-FCN and BP neural network-based Phishing Scam accounts detection model). This study aims to identify phishing scam accounts, specifically on Ethereum. The LBPS model employs a hybrid deep neural network approach, utilizing a BP neural network to reveal implicit relationships within transaction record features and an LSTM-FCN neural network to capture temporal features from the transaction records of a targeted account.

The main objective of this work is to enhance the performance of a phishing website classifier by analyzing its characteristics and selecting the optimal combination of features for training. The goal is to create more accurate and efficient classifiers capable of better identifying and preventing phishing attacks.

The subsequent sections of this study are organized as follows: Section 2 outlines the general architecture of the proposed system. Section 3 presents a performance evaluation of the system. Lastly, Section 4 provides the concluding remarks of the paper.

2. RESEARCH METHODOLOGY

The goal of this study is to analyze various machine learning techniques to explore the potential applications of five different classification models in identifying phishing websites. The primary objective is to develop an intelligent model capable of assessing the authenticity of a website and determining the degree of deception. The suggested method is illustrated in Figure 1.

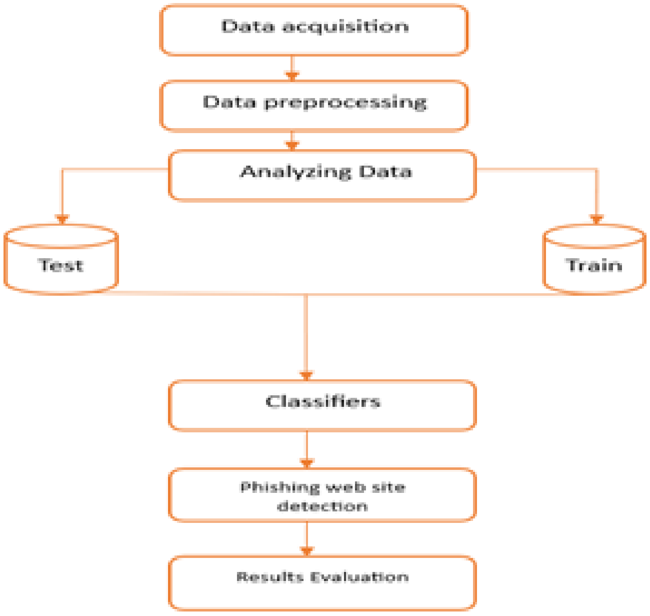

FIGURE 1. - Proposed system

2.1 Data Set

The "Phishing Websites Dataset" from Kaggle [12] is a collection of URLs classified as either legitimate or phishing websites. The dataset comprises 11,055 URLs, with 5,643 labeled as legitimate and 5,412 as phishing. It combines two sources: the first, legitimate URLs from the Alexa Top 1 million websites, and the second, known phishing URLs from PhishTank, a community-driven anti-phishing service. Users can report suspected phishing websites to PhishTank for examination and validation. A site is added to the PhishTank database upon confirmation. OpenPhish, a community-driven phishing website database, offers an open-source alternative to PhishTank, enabling submissions without requiring a user account. Google Safe Browsing, a vital cybersecurity service, safeguards users from online threats by utilizing a comprehensive network of data sources and advanced algorithms to protects users from online threats by using a comprehensive network of data sources and advanced algorithms to identify and block harmful content. This proactive approach empowers users to navigate the digital world confidently. The Phishing Websites Dataset is assembled from various sources, including user submissions, web scraping, and publicly accessible listings of reputable and phishing websites. The inclusion of diverse data sources enhances the dataset's comprehensiveness and representation of the actual threat posed by phishing websites.

This dataset may be utilized for a number of machines learning applications, including binary classification, anomaly detection, and feature engineering for identifying phishing websites.

2.2 Data Preprocessing

There are several preprocessing methods that can be applied to the "Phishing Websites Dataset" to prepare it for use in machine learning models. For instance, data cleaning involves removing duplicate or irrelevant entries. For example, eliminate URLs with an invalid format. Extract valuable features from URLs, such as length, presence of specific keywords, or count of special characters [2]. Normalize the data for consistency, like converting all URLs to lowercase and removing unnecessary white space. Convert categorical data, such as the label (phishing or legitimate), into a numerical format usable by machine learning algorithms. If the dataset is unbalanced (one class contains disproportionately more samples than the other), resample the data to balance the classes [13]. These preprocessing methods enhance the dataset's quality for machine learning tasks like binary classification. In the machine learning workflow, separating data into training (80%) and testing sets (20%) is a crucial stage since it enables the evaluation of model performance and generalization capacity.

2.3 Classification

Data classification involves the systematic organization and categorization of data into distinct groups or classes, leveraging similarities or dissimilarities in their features or characteristics. The purpose of data classification is to enable efficient and effective data management, analysis, and decision-making [14]. In machine learning, data classification is a common task that involves training a model to learn the relationships between the features of the data and the class labels or categories [15]. The model that has been trained can subsequently be applied to make predictions on the class labels of new and unseen data. Various algorithms, such as logistic regression, decision trees, XGBoost, among others, can be utilized for data classification, with the selection of the appropriate algorithm contingent upon the data characteristics and the specific classification objective. Upon successful training and validation of the model, it can be applied to classify new data in real-time [16] [17].

Overall, data classification is a crucial aspect of data analysis and machine learning, enabling the effective management and utilization of large and complex datasets. The objective of this work is to employ diverse machine learning algorithms, such as logistic regression, Adaptive Boosting, decision tree, Random Forest, and XGBoost, to classify a dataset of phishing websites. Each of these algorithms possesses its unique strengths and limitations, potentially making them more suitable for specific types of data or classification tasks.

2.3.1.Logistic Regression

Logistic regression is a popular technique used for binary classification tasks, where it models the probability of an outcome belonging to one of the two classes. It is frequently employed in machine learning problems involving binary classification, where the goal is to divide data into one of two groups based on a collection of characteristics [18]. Based on the provided independent variables, the logistic regression model uses a sigmoid or logistic function to predict the likelihood that the dependent variable will be 1. The sigmoid function maps real-valued inputs to a range of 0 to 1, representing the probability. The logistic regression algorithm utilizes maximum likelihood estimation to estimate the coefficients of the independent variables. These coefficients are then used to calculate the probability of the dependent variable being 1 based on the input data. In practice, logistic regression can be applied to a variety of applications such as credit scoring, disease diagnosis, and fraud detection. Due to its simplicity, interpretability, and resilience, it makes it a well-liked technique [19].

2.3.2.Decision Trees

A tree-based model is commonly used in machine learning for tasks involving regression and classification. It visually depicts decision-making options based on circumstances and their results. The structure comprises nodes, branches, and leaves. Nodes represent tests on input features, branches depict possible outcomes, and leaves represent final decisions or classifications. The algorithm constructs the tree by recursively partitioning data into subsets based on input feature values [20]. Both multi-class and binary categorization issues can benefit from the versatility of decision trees. The objective is to develop a model that predicts the target variable by sequentially making decisions based on input features. The algorithm learns optimal decision rules from the training data to minimize classification error and enhance predictive accuracy. Notably, decision trees are interpretable, as they offer a clear and intuitive means to visualize the decision-making process. This makes it easier to comprehend the factors that influence the final decision or outcome [21]. Furthermore, decision trees have the capability to handle both continuous and categorical input features, and they exhibit resilience to missing values and outliers. However, decision trees can be susceptible to overfitting, resulting in potential challenges with the generalization to new, unseen data.

2.3.3.Random Forest

The Random Forest is an ensemble learning technique that comprises decision trees trained independently on randomized subsets of the training data and input features. Results are obtained by aggregating the trees' outputs, usually through averaging or majority vote. This approach mitigates overfitting and can lead to improved predictive accuracy and robustness in the model's performance [22]. The fundamental concept underlying random forests is to address overfitting and enhance the model's accuracy by amalgamating multiple decision trees. A unique subset of the input characteristics and training data is used to train each tree in the forest, thereby reducing the variance of the model and improving its generalization performance. By combining the outputs of these individual trees, typically through averaging or majority vote, the random forest ensemble approach mitigates overfitting and yields a more robust and accurate prediction or classification model. The random forest algorithm works by selecting a random subset of the training data and input features at each node of each tree. It then constructs a decision tree based on the selected data and features [23]. This procedure is iteratively repeated several times, resulting in a collection or "forest" of decision trees. During the prediction phase, the random forest consolidates the outputs of all the trees by averaging or taking a majority vote, arriving at the final prediction or classification. The accuracy and resilience of the model are improved by this ensemble technique, leveraging the collective decision-making power of multiple trees.

2.3.4 Adaptive Boosting

Adaptive Boosting, also known as AdaBoost, is a popular ensemble learning technique that combines weak classifiers to create a more robust and accurate overall classifier. AdaBoost iteratively combines the predictions of multiple weak classifiers to create a stronger classifier, adaptively adjusting the weights of training samples to give more importance to misclassified samples. This boosting approach enhances the performance of the classifier, making it capable of handling complex data patterns and achieving higher accuracy compared to individual weak classifiers. AdaBoost is particularly effective when working with complex datasets containing many input features and classes [24]. The training data is divided into different subsets for each weak classifier, and the weights of the training samples are dynamically adjusted during each iteration to give higher importance to the samples that were misclassified by the previous weak classifiers [14]. This process creates a new training set biased towards the samples that were previously misclassified, forcing the weak classifiers to focus on these samples and improve their performance. The final output of the algorithm is a weighted sum of the predictions of all the weak classifiers, with the weights determined by the accuracy of each weak classifier. AdaBoost has several advantages over other ensemble methods, such as random forests and bagging. It is less prone to overfitting, works well with high-dimensional data, and can handle noisy and incomplete data [25]. One of the main limitations of AdaBoost is its sensitivity to outliers and noise in the training data. If the data contains many outliers or noisy samples, the algorithm may overfit to these samples and perform poorly on new data. Additionally, AdaBoost can be computationally expensive, requiring training many weak classifiers on multiple subsets of the training data.

2.3.5.Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting is a highly effective machine learning algorithm used for classification tasks. As an ensemble learning method, it combines predictions from multiple weak models, often in the form of decision trees, to generate a more precise and resilient final prediction. Notable for its superior efficiency, speed, and capacity to handle sizable datasets, XGBoost has gained popularity in the field of machine learning [26]. This process continues until the required degree of accuracy is reached for a certain number of iterations. XGBoost also incorporates several regularization techniques, such as L1 (Lasso) and L2 (Ridge) regularization, to prevent overfitting and improve the model's generalization performance. These regularization techniques penalize large weights or complex models, thereby promoting simpler and more stable models [26].

Once the ensemble of trees is built, predictions are made by aggregating the predictions of all the trees. Typically, XGBoost uses a combination of weighted voting or averaging to obtain the final predicted class probabilities [26]. XGBoost has emerged as a favored option for numerous classification tasks owing to its proficiency in managing imbalanced datasets, effectively handling missing values, and conducting feature importance analysis. This analysis aids in identifying the most pertinent features that contribute to precise predictions. As a result, XGBoost has gained popularity in the field of machine learning for its unique capabilities in addressing these common challenges. To classify the phishing websites dataset, each of these algorithms may be trained and evaluated using the training and testing sets created earlier.

Metrics like recall, accuracy, and precision may be used to measure each algorithm's performance. Based on the results, the most accurate and effective algorithm can be chosen for deployment in real-world scenarios.

3. RESULTS

Classification performance evaluation is the process of measuring the accuracy and effectiveness of a classification model in predicting the correct class label for a given set of input data [27]. It is an essential step in the model development process, providing insights into the strengths and weaknesses of the model and aiding in identifying areas for improvement.

The most common measures of classification performance evaluation include [28]:

1. Accuracy: Based on the proportion of properly identified samples to all of the dataset's samples, it provides a holistic assessment of the model's performance by quantifying its ability to make accurate predictions.

2. Precision: A classification model's capacity to isolate just the pertinent data elements. In mathematics, precision is calculated by dividing the total number of true positives by the sum of true positives and false positives [13][29].

3. Recall: The capacity of a model to locate all pertinent instances in a data source. Recall is calculated mathematically as the product of the number of true positives divided by the sum of true positives and false negatives. It indicates the proportion of positive predictions made by the model that are actually true positives [30].

4. F1-Score: The F1-score illustrates the compromise between recall and accuracy, calculating the harmonic mean between each pair. Consequently, it considers observations that are both falsely positive and falsely negative [30].

By using these measures, we can evaluate the performance of a classification model and select the best-performing model for our specific problem. Figure (2) shows the model’s accuracy.

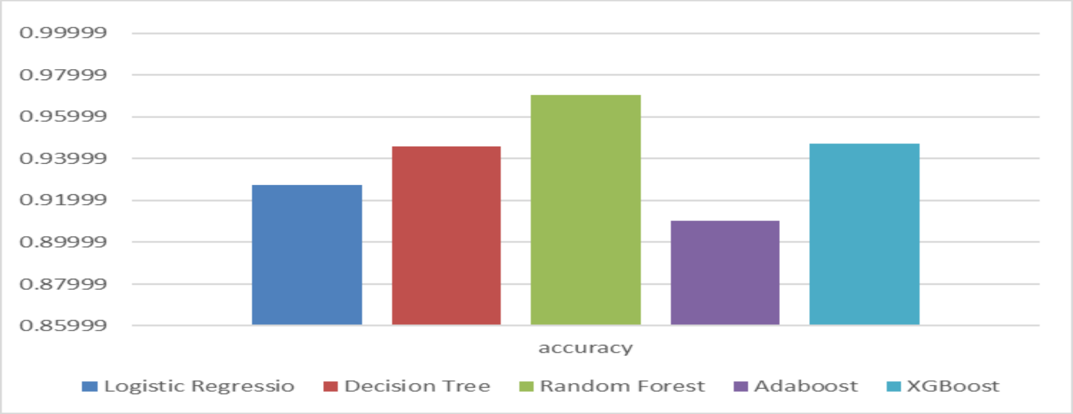

FIGURE 2. - **Model's accuracy**

The Random Forest model, with an accuracy of 0.97, exhibits the highest accuracy, followed by the Decision Tree and XGBoost models, both with accuracy scores of 0.94. The Logistic Regression model has an accuracy of 0.92, lower than the other three models. The Adaboost model has the lowest accuracy at 0.91.

Random Forest's superior performance stems from several key factors. Firstly, as an ensemble technique, it aggregates multiple decision trees, reducing overfitting and enhancing stability through the combination of individual predictions. The method's feature randomness—employing subsets of features for each tree—ensures diverse perspectives on the data, preventing the dominance of a single influential feature. This technique adeptly captures complex, non-linear relationships within the dataset, making it highly adaptable to various scenarios. Random Forest also provides feature importance metrics, aiding in data understanding and guiding feature selection. Its resilience against overfitting, robustness against noise, and simplicity in hyperparameter tuning contribute to its reliability and accuracy in classification tasks.

It is important to note that accuracy alone is not always the best metric for evaluating a classifier's performance. Other metrics, such as precision and recall, should also be considered to ensure the model performs well on all aspects of the data. Additionally, consider the context and specific requirements of the problem when selecting a model. As for the precision and recall of each method, they are shown in the following figures. Figure (3) shows the Recall and Precision of the Adaboost classifier.

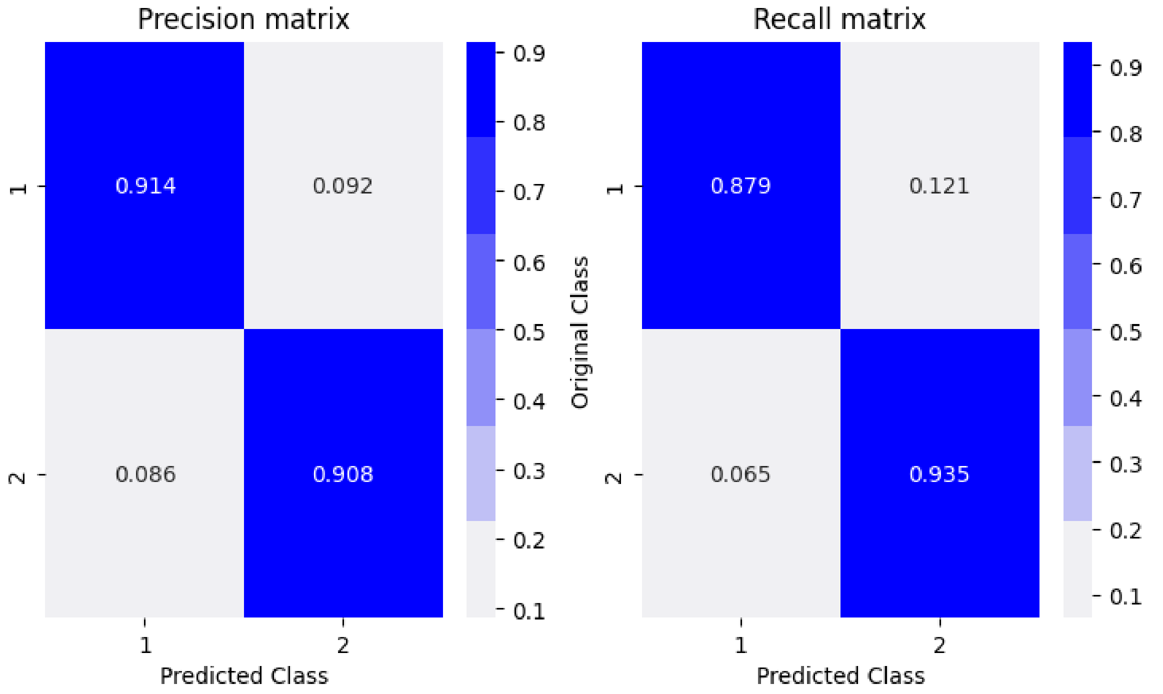

FIGURE 3. - **AdaBoost Precision and Recall Matrix**

The precision of the model is high for both classes: 88% for the negative class (-1) and 93% for the positive class (2), as seen in Figure (3). This means that when the model predicts a sample to be in a certain class, it is correct 88% of the time for the negative class and 93% of the time for the positive class. The recall of the model is also high for both classes: 91% for the negative class and 91% for the positive class. This means that the model correctly identified 91% of all negative samples and 91% of all positive samples in the test set.

Figure (4) shows the Recall and Precision for Decision Tree classifier.

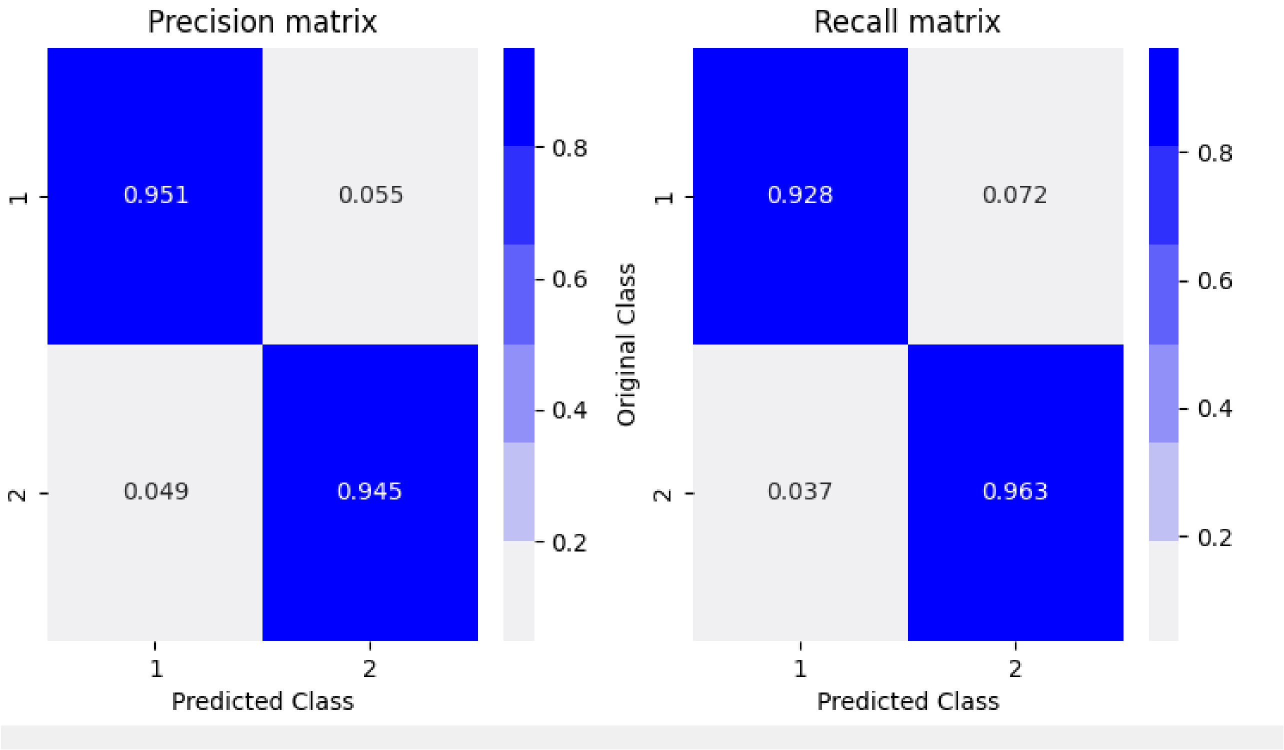

FIGURE 4. - **Decision Tree Precision and Recall Matrix**

The precision of the model for decision Tree as seen in Figure (4) is high for both classes: 93% for the negative class (-1) and 96% for the positive class (1). This means that when the model predicts a sample to be in a certain class, it is correct 93% of the time for the negative class and 96% of the time for the positive class.

The recall of the model is also high for both classes: 95% for the negative class and 94% for the positive class. This means that the model correctly identified 95% of all negative samples and 94% of all positive samples in the test set.

Figure (5) show the obtained for the Logistic Regression classifier.

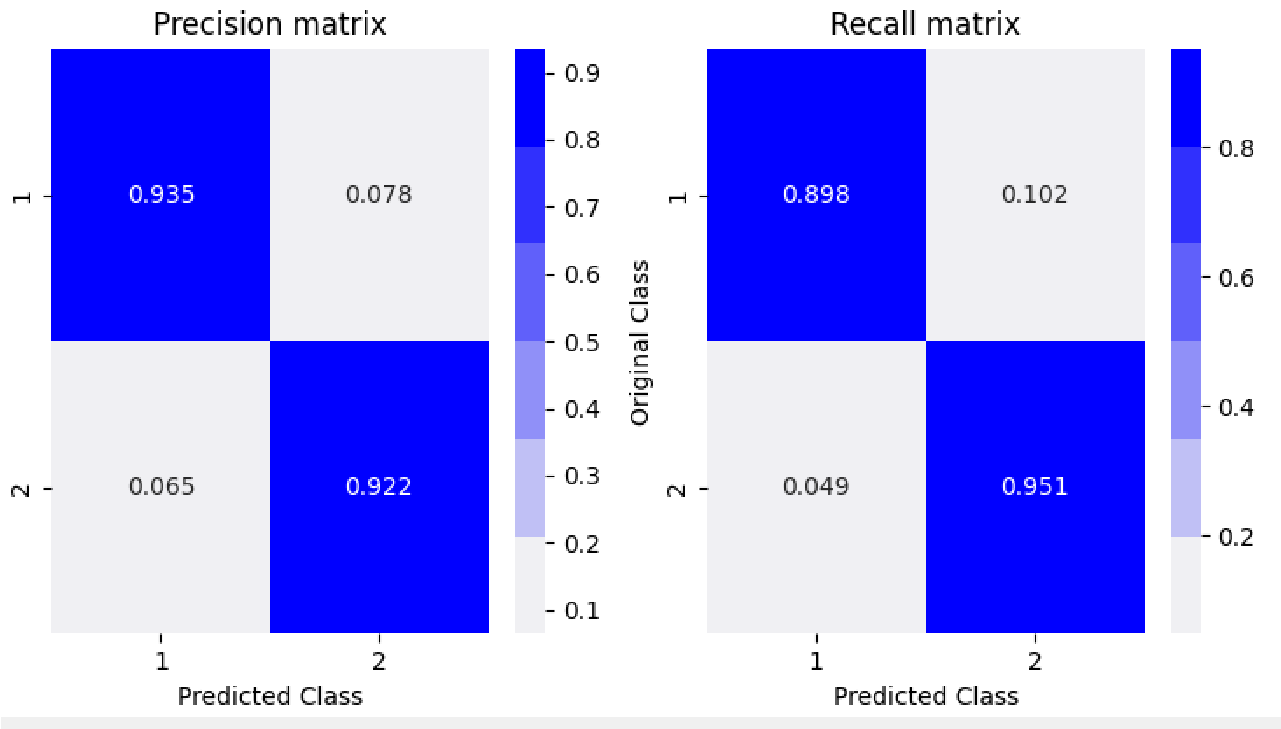

FIGURE 5. - Logistic Regression Precision and Recall Matrix

The precision of the model is high for both classes: 90% for the negative class (-1) and 95% for the positive class (1), as seen in Figure (5). This means that when the model predicts a sample to be in a certain class, it is correct 90% of the time for the negative class and 95% of the time for the positive class.

The recall of the model is also high for both classes: 94% for the negative class and 92% for the positive class. This means that the model correctly identified 94% of all negative samples and 92% of all positive samples in the test set.

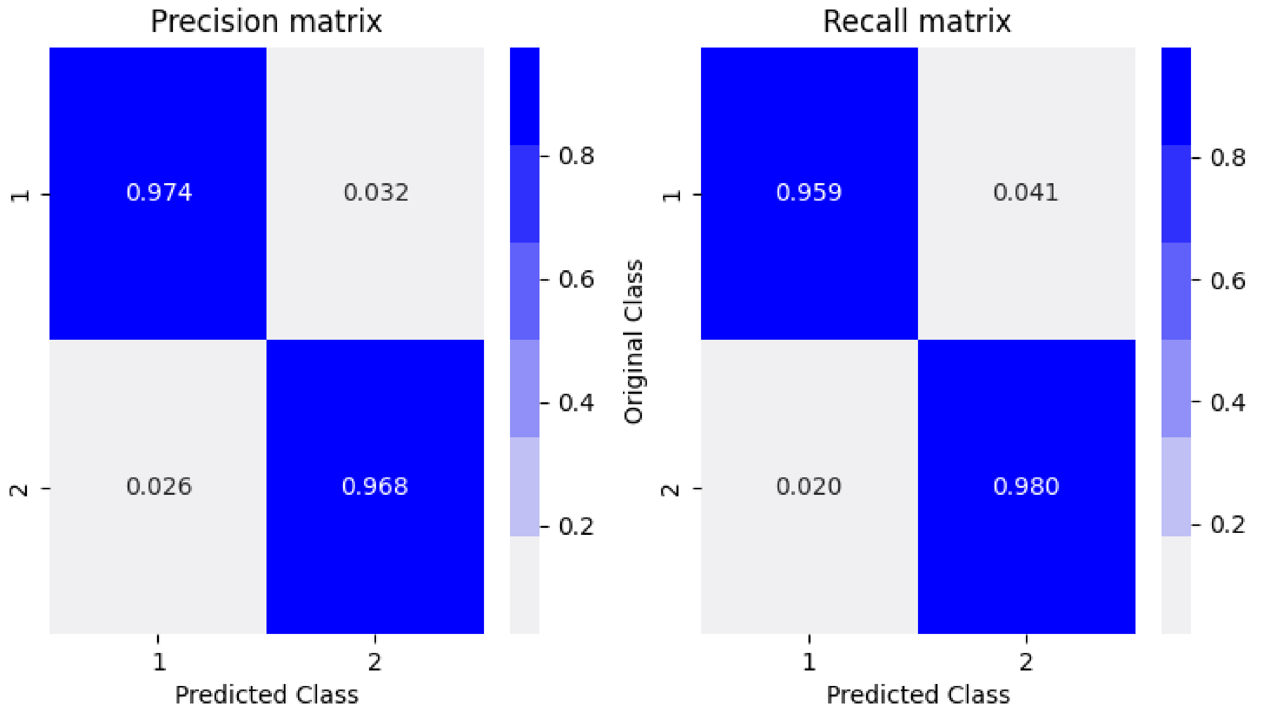

FIGURE 6. - Random Forest Precision and Recall Matrix

The above Figure (6) illustrates that the precision of the random forest model is high for both classes: 96% for the negative class (-1) and 98% for the positive class (1). This implies that when the model predicts a sample to be in a certain class, it is correct 96% of the time for the negative class and 98% of the time for the positive class. The recall of the model is also high for both classes: 97% for the negative class and 97% for the positive class. This means that the model correctly identified 97% of all negative samples and 97% of all positive samples in the test set.

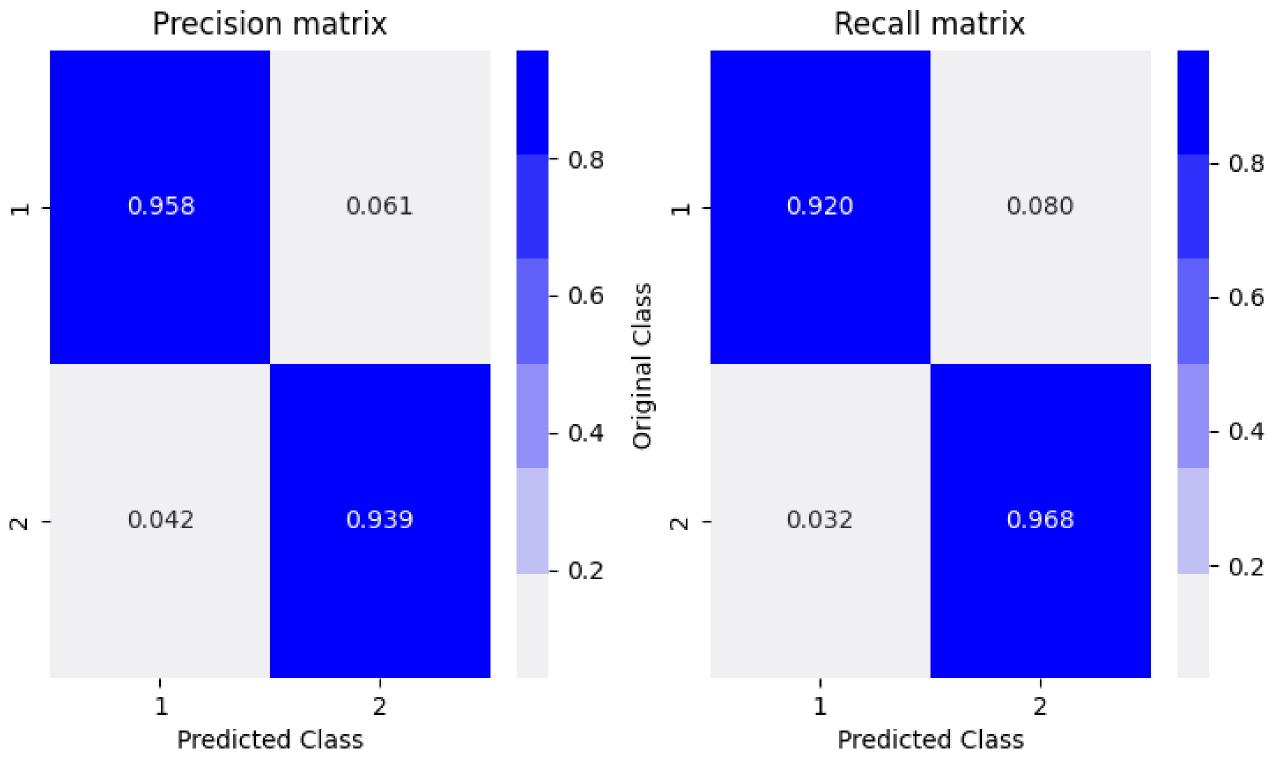

FIGURE 7. - **XGboost Precision and Recall Matrix**

It can be seen from Figure (7) that the precision of the XGBoost model is high for both classes: 92% for the negative class (-1) and 97% for the positive class (1). This implies that when the model predicts a sample to be in a certain class, it is correct 92% of the time for the negative class and 97% of the time for the positive class. The model's recall is also high for both classes: 96% for the negative class and 94% for the positive class. This indicates that the model correctly identified 96% of all negative samples and 94% of all positive samples in the test set.

The F1 score, a commonly used metric for evaluating classification models, is the harmonic mean of precision and recall. It balances both the proportion of positive predictions that are correct (precision) and the proportion of actual positives that are correctly identified (recall). The F1 score is valuable as a balanced measure of precision and recall, allowing for unbiased comparison of different models on diverse datasets. Table 1 displays the obtained F1 score metrics.

The results of F1 score metrics are tabulated in Table 1.

Method	Class	F1 score
Logistic Regression	Phishing website	0.92
	Normal	0.94
Decision Tree	Phishing website	0.94
	Normal	0.95
Random Forest	Phishing website	0.97
	Normal	0.97
ADaboost	Phishing website	0.90
	Normal	0.92
XGboost	Phishing website	0.94
	Normal	0.95

In summary, based on the F1 scores provided, it is observed that: Random Forest seems to perform exceptionally well, achieving high accuracy in classifying both "Phishing" and "Normal" websites. Decision Tree and XGBoost show strong performance, especially in classifying "Normal" websites. Logistic Regression and AdaBoost demonstrate decent performance but lag behind slightly in accuracy compared to the other models.

Table 1 compares the proposed system and some new methods to detect phishing websites.

Table 1. - **Methods' comparison**

Method	Advantages	Disadvantages
VGG [31]	1-Efficient at capturing complex characteristics and patterns in visual data. 2-Possesses robust feature extraction.	1-Expensive in terms of computing; requires strong hardware and extended training periods. 2-Huge volumes of labeled data are necessary for training.
ResNet (Residual Network) [32]	1-Capable of successfully training very deep networks. 2-Can effectively addresses the challenge of the vanishing gradient problem. 3-Will achieve state-of-the-art performance in various image tasks.	1-Computationally expensive; requires powerful hardware and longer training times. 2-Requires large amounts of labeled data for training.
CNN, RNN [33][34]	1-Can capture complex patterns and relationships in high-dimensional data. 2-Is effective in handling unstructured data, such as textual and visual information.	2-Requires large amounts of labeled data for training. 2-Computationally expensive; requires powerful hardware and longer training times.
Proposed system	1-Easy to interpret and understand. 2-Can work well with small- to medium-sized datasets.	1-May struggle with capturing complex patterns in high-dimensional data. 2-Limited in handling unstructured data.

Implementing machine learning-based phishing detection systems in real-world environments offers promising advantages but also poses significant challenges. Feasibility hinges on the accuracy, adaptability, and automated threat detection capabilities exhibited by models like Decision Trees, Random Forest, and boosting methods. These systems promise improved accuracy in identifying phishing attempts, adaptability to evolving threats, and the ability to automatically detect and respond to potential attacks. However, challenges persist in obtaining high-quality labeled datasets, ensuring model interpretability, integrating these models into existing systems, defending against adversarial attacks, and managing resource-intensive computations. Data quality, interpretability, deployment complexity, adversarial threats, and resource requirements need addressing to effectively implement these machine learning models into practical applications for robust phishing detection systems. Collaborative efforts among cybersecurity experts, data scientists, and industry professionals are essential to overcome these challenges and harness the potential of machine learning in real-world cybersecurity applications.

4. CONCLUSION

Phishing website detection systems have evolved significantly in recent years due to the increasing sophistication of phishing attacks. Phishing poses a significant threat in the corporate environment, resulting in substantial financial losses. Despite various solutions proposed and implemented by reputable cybersecurity companies, the number of successful phishing attacks is increasing rapidly, indicating that current methods are inadequate to combat this problem. In this paper, various phishing detection and mitigation methods are being developed to improve on previous approaches by providing higher accuracy and better results.

Based on the provided information, the random forest model emerged as the most effective classification algorithm among those evaluated. Its outstanding accuracy rate of 96.89% surpassed the performance of both the decision tree model (94.57%) and Extreme Gradient Boosting (XG). This superior performance suggests that the random forest model is well-suited for various classification tasks and should be considered the primary choice for such applications. New trends in phishing website detection systems include behavioral analysis. This involves analyzing user behavior on a website to identify suspicious activities such as multiple login attempts, rapid clicking, unusual navigation patterns, or domain-based authentication. Behavioral analysis gathers data on user interactions, such as mouse movements, keystrokes, time spent on pages, click patterns, and navigation behavior. These features can be extracted and processed to create a dataset for training machine learning models. Integrating behavioral analysis with machine learning models offers a proactive and user-focused approach to phishing detection, leveraging the strengths of both fields. It allows for a more comprehensive and dynamic system that adapts to the changing tactics of attackers, thereby enhancing the overall security posture against phishing attempts.

Researchers can delve into developing advanced machine learning techniques, such as deep learning models, to improve accuracy and robustness in identifying new phishing tactics. Exploring the integration of blockchain technology or secure authentication mechanisms to enhance website verification and user protection against phishing is another avenue. Lastly, addressing the challenge of detecting phishing attacks in non-English languages and developing cross-lingual solutions is an open question that requires further exploration to ensure global cybersecurity resilience.

Funding

None

ACKNOWLEDGEMENT

None

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] M. A. Adebowale, K. T. Lwin, M. A. Hossain, “Intelligent phishing detection scheme using deep learning algorithms,” Journal of Enterprise Information Management, 2020, doi: 10.1108/JEIM-01-2020-0036.

[2] S. Ozgur, B. Ebubekir, D. Onder, D. Banu, "Machine learning based phishing detection from URLs," Expert Systems with Applications, vol. 117, pp. 345–357, 2019.

[3] J. Tanimu, S. Shiaeles, “Phishing Detection Using Machine Learning Algorithm,” Proceedings of the 2022 IEEE International Conference on Cyber Security and Resilience, CSR 2022, pp. 317–322, 2022, doi: 10.1109/CSR54599.2022.9850316.

[4] T. N. Junaid Rashid, T. Mahmood, M. W. Nisar, “Phishing Detection Using Machine Learning Technique,” 2020, doi: DOI 10.1109/SMART-TECH49988.2020.00026.

[5] A. Hannousse, S. Yahiouche, “Towards benchmark datasets for machine learning based website phishing detection: An experimental study,” Engineering Applications of Artificial Intelligence, vol. 104, 2021, doi: 10.1016/j.engappai.2021.104347.

[6] N. Kumar, S. Sonowal, Nishant, “Email Spam Detection Using Machine Learning Algorithms,” Proceedings of the 2nd International Conference on Inventive Research in Computing Applications, ICIRCA 2020. pp. 108–113, 2020, doi: 10.1109/ICIRCA48905.2020.9183098.

[7] M. Sameen, K. Han, S. O. Hwang, "PhishHaven_An Efficient Real-Time AI Phishing URLs Detection System," IEEE ACCESS, vol. 8, DOI: 2020.2991403

[8] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," Computer Communications, vol. 175, pp. 47–57, 2021, ISSN 0140-3664, https://doi.org/10.1016/j.comcom.2021.04.023.

[9] A. Prasad, S. Chandra, "PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning," Computers & Security, vol. 136, 2024, 103545, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2023.103545.

[10] T. Nagunwa, P. Kearney, P. Fouad, "A machine learning approach for detecting fast flux phishing hostnames," Journal of Information Security and Applications, vol. 65, 2022, 103125, ISSN 2214-2126, https://doi.org/10.1016/j.jisa.2022.103125.

[11] T. Wen, Y. Xiao, A. Wang, H. Wang, "A novel hybrid feature fusion model for detecting phishing scam on Ethereum using deep neural network," Expert Systems with Applications, vol. 211, 2023, 118463, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2022.118463.

[12] Gfk Healthcare, “Kaggle. (n.d.). Phishing Websites Dataset," [Online] Available: https://www.kaggle.com/akashkr/phishing-website-dataset,” Accessed: Apr. 16, 2023.

[13] M. Dewis, T. Viana, “Phish Responder: A Hybrid Machine Learning Approach to Detect Phishing and Spam Emails,” Applied System Innovation, vol. 5, no. 4, 2022, doi: 10.3390/asi5040073.

[14] S. Alnemari, M. Alshammari, “Detecting Phishing Domains Using Machine Learning,” Appl. Sci, vol. 13, 2023, doi: https://doi.org/ 10.3390/.

[15] M. Taha, H. Ahmed, “Second-Order Statistical Methods GLCM for Authentication Systems,” Iraqi J. Electr. Electron. Eng., vol. 17, no. 1, pp. 1–6, 2021, doi: 10.37917/ijeee.17.1.10.

[16] D. M. A. D. S. A. A. A. A. Taha, “A Review of Classifications Techniques and computer aided used for Breast Cancer.pdf,” Wasit J. Pure Sci., vol. 1, no. 2, 2022.

[17] A. Sharma, “Guided Stochastic Gradient Descent Algorithm for inconsistent datasets,” Applied Soft Computing Journal, vol. 73. pp. 1068–1080, 2018, doi: 10.1016/j.asoc.2018.09.038.

[18] J. P. Li, A. U. Haq, S. U. Din, J. Khan, A. Khan, A. Saboor, “Heart Disease Identification Method Using Machine Learning Classification in E-Healthcare,” IEEE Access, vol. 8. pp. 107562–107582, 2020, doi:

10.1109/ACCESS.2020.3001149.

[19] J. Daniel, J. H. Martin, “Chapter 5 - Speech and Language Processing.” 2023.

[20] A. I. Adler, A. Painsky, “Feature Importance in Gradient Boosting Trees with Cross-Validation Feature Selection,” Entropy, vol. 24, no. 5, 2022, doi: 10.3390/e24050687.

[21] M. S. K. Swaroop, K. R. Chowdary, “Phishing websites detection using machine learning,” Int. J. Recent Technol. Eng., vol. 8, no. 4, pp. 1470–1474, 2021, doi: 10.35940/ijrte.B1018.0982S1119.

[22] A. Mandadi, S. Boppana, V. Ravella, R. Kavitha, “Phishing Website Detection Using Machine Learning,” 2022, doi: 10.1109/I2CT54291.2022.9824801.

[23] E. Y. Boateng, D. A. Abaye, “A Review of the Logistic Regression Model with Emphasis on Medical Research,” J. Data Anal. Inf. Process., vol. 07, no. 4, pp. 190–207, 2019, doi: 10.4236/jdaip.2019.74012.

[24] H. T. Elshoush, E. A. Dinar, “Using adaboost and stochastic gradient descent (SGD) algorithms with r and orange software for filtering e-mail spam,” 2019 11th Computer Science and Electronic Engineering Conference, CEEC 2019 - Proceedings. pp. 41–46, 2019, doi: 10.1109/CEEC47804.2019.8974319.

[25] C. Tu, H. Liu, B. Xu, "AdaBoost typical Algorithm and its application research," MATEC Web of Conferences, vol. 139, 2017, 00222. 10.1051/matecconf/201713900222.

[26] P. Bhavani, A. Chalamala, M. Likhitha, P. S. Sai, C. P. Sai (September 2, 2022). "Phishing Websites Detection Using Machine Learning." Available at SSRN: https://ssrn.com/abstract=4208185 or http://dx.doi.org/10.2139/ssrn.4208185

[27] H. M. A. Mohammed A. Taha, “A FUZZY VAULT DEVELOPMENT BASED ON IRIS IMAGES,” EUREKA Phys. Eng., vol. 5, no. 1, 2021.

[28] G. Chugh, S. Kumar, N. Singh, “Survey on Machine Learning and Deep Learning Applications in Breast Cancer Diagnosis,” Cognitive Computation, vol. 13, no. 6. pp. 1451–1470, 2021, doi: 10.1007/s12559-020-09813-6.

[29] S. R. A. S. 1, J. L. W. 5, A. B. 6, S. Balasubaramanian 2, A. S. Al-Kaabi 1, B. Sharma 3, S. Chowdhury 4, A. Mehbodniya 5, “Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection,” Electron., vol. 12, p. 1642, 2023, https//doi.org/ 10.3390/electronics12071642 Acad.

[30] B. Al-Ahmad, A. M. Al-Zoubi, A. K. Ruba, A. Ibrahim, “An Evolutionary Fake News Detection Method for COVID-19,” Symmetry (Basel)., pp. 1–16, 2021.

[31] B. Al-Ahmad, A. M. Al-Zoubi, A. K. Ruba, and A. Ibrahim, “An Evolutionary Fake News Detection Method for COVID-19,” Symmetry (Basel)., pp. 1–16, 2021.

[32] N. B. Trinh, T. D. Phan, V. H. Pham. “Leveraging Deep Learning Image Classifiers for Visual Similarity-based Phishing Website Detection,” in Proceedings of the 11th International Symposium on Information and Communication Technology (SoICT '22). Association for Computing Machinery, NY, USA, 2022, pp. 134–141. https://doi.org/10.1145/3568562.3568629

[33] U. Saeed, "Visual similarity-based phishing detection using deep learning," J. Electron. Imag., vol. 31, no. 5, p. 051607, https://doi.org/10.1117/1.JEI.31.5.051607

[34] R. Yang, K. Zheng, B. Wu, C. Wu, X. Wang, “PhishingWebsite Detection Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning,” Sensors, vol. 21, p. 8281, 2021, https://doi.org/10.3390/s21248281.