Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.051778

ARTICLE

Phishing Attacks Detection Using Ensemble Machine Learning Algorithms

Nisreen Innab[1], Ahmed Abdelgader Fadol Osman[2], Mohammed Awad Mohammed Ataelfadiel[2], Marwan Abu-Zanona[3,*], Bassam Mohammad Elzaghmouri[4], Farah H. Zawaideh[5] and Mouiad Fadeil Alawneh[6]

[1]Department of Computer Science and Information Systems, College of Applied Sciences, Almaarefa University, Diriyah, Riyadh, 13713, Saudi Arabia

[2]Applied College, King Faisal University, Al-Ahsa, 31982, Saudi Arabia

[3]Department of Management Information Systems, College of Business Administration, King Faisal University, Al-Ahsa, 31982, Saudi Arabia

[4]Department of Computer Science, Faculty of Computer Science and Information Technology, Jerash University, Jerash, 26110, Jordan

[5]Department of Business Intelligence and Data Analysis, Faculty of Financial Sciences and Business, Irbid National University, Irbid, 21110, Jordan

[6]Faculty of Information Technology, Ajloun National University, Ajloun, 26767, Jordan

*Corresponding Author: Marwan Abu-Zanona. Email: mabozanoneh@kfu.edu.sa

Received: 15 March 2024 Accepted: 03 June 2024 Published: 18 July 2024

ABSTRACT

Phishing, an Internet fraud where individuals are deceived into revealing critical personal and account information, poses a significant risk to both consumers and web-based institutions. Data indicates a persistent rise in phishing attacks. Moreover, these fraudulent schemes are progressively becoming more intricate, thereby rendering them more challenging to identify. Hence, it is imperative to utilize sophisticated algorithms to address this issue. Machine learning is a highly effective approach for identifying and uncovering these harmful behaviors. Machine learning (ML) approaches can identify common characteristics in most phishing assaults. In this paper, we propose an ensemble approach and compare it with six machine learning techniques to determine the type of website and whether it is normal or not based on two phishing datasets. After that, we used the normalization technique on the dataset to transform the range of all the features into the same range. The findings of this paper for all algorithms are as follows in the first dataset based on accuracy, precision, recall, and F1-score, respectively: Decision Tree (DT) (0.964, 0.961, 0.976, 0.968), Random Forest (RF) (0.970, 0.964, 0.984, 0.974), Gradient Boosting (GB) (0.960, 0.959, 0.971, 0.965), XGBoost (XGB) (0.973, 0.976, 0.976, 0.976), AdaBoost (0.934, 0.934, 0.950, 0.942), Multi Layer Perceptron (MLP) (0.970, 0.971, 0.976, 0.974) and Voting (0.978, 0.975, 0.987, 0.981). So, the Voting classifier gave the best results. While in the second dataset, all the algorithms gave the same results in four evaluation metrics, which indicates that each of them can effectively accomplish the prediction process. Also, this approach outperformed the previous work in detecting phishing websites with high accuracy, a lower false negative rate, a shorter prediction time, and a lower false positive rate.

KEYWORDS

Social engineering; attacks; phishing attacks; machine learning; security; artificial intelligence

1 Introduction

The Internet has become a vital tool for individuals. In 2014, over 40% of the global population utilized the Internet, with industrialized countries experiencing a higher adoption rate of 78%. The North Atlantic Treaty Organization or NATO recognizes the Internet as a crucial asset for governments, an essential component of national infrastructures, and a significant catalyst for socio-economic progress and advancement [1]. The proliferation of Internet usage has led to the emergence of malicious code and malware that aim to infiltrate computer systems by attacking and destroying the information stored within them. These attacks are specifically crafted to collect users' information, including credit card numbers and passwords, as well as to distribute information without the user's consent. Malware is software that possesses the ability to do harm to data and systems [2,3]. The threat extends beyond individuals to encompass organizations, enterprises, and even governments, encompassing both civil and military infrastructures. These entities face the risk of losing vital information and damaging their reputation. Several instances have occurred in recent years when credit and debit cards have been unlawfully obtained from online payment systems, Google's intellectual property has been unlawfully taken, and users' personal information has been exposed, among other examples [4,5].

Various definitions exist for cybersecurity, one of which is provided by Kaspersky Lab: Cybersecurity refers to the act of safeguarding computers, servers, mobile devices, electronic systems, networks, and data from harmful attacks [6,7]. It is alternatively referred to as information technology security or electronic information security. The word encompasses a wide range of topics, including computer security, disaster recovery, and end-user education. Cybersecurity aims to safeguard personal, governmental, and business data from unauthorized access or alteration. It primarily involves three key tasks: (a) implementing measures to protect hardware, software, and the information they store, (b) ensuring the state or quality of protection against various threats, and (c) implementing and enhancing these activities [8,9].

In the dynamic realm of cybersecurity, characterized by more complex and varied threats, the use of machine learning (ML) has emerged as a crucial factor in strengthening digital security measures. ML enables cybersecurity experts to examine large information, identify irregularities, and forecast potential risks in real time. This innovative technology not only improves the efficiency and precision of identifying potential threats but also allows for proactive measures to be taken in response to developing cyber hazards [10,11]. Conventional security solutions frequently face difficulties in keeping up with the ever-changing nature of cyber threats. Given the vast amount and intricate nature of data produced by networks, systems, and users, conducting manual analysis becomes unfeasible. Machine learning algorithms are particularly adept at handling large datasets, detecting trends, and adjusting to changing attack methods. The flexibility of ML to adapt makes it a significant resource for addressing the always-evolving threat scenario [12].

Machine learning is utilized in multiple areas of cybersecurity, encompassing tasks such as recognizing harmful malware, identifying abnormal user actions, forecasting potential weaknesses, and automating incident response. Through the utilization of sophisticated algorithms, cybersecurity experts may maintain an advantageous position over cyber adversaries, thereby acquiring a proactive advantage in protecting vital digital resources. Nevertheless, this integration is not devoid of its difficulties. ML models necessitate meticulous training and validation to guarantee precision and mitigate the risk of erroneous positive or negative outcomes. Furthermore, the ethical considerations surrounding the use of automated systems in security settings and the possibility of adversarial assaults on machine learning models contribute to the intricacy of their implementation [13,14].

Phishing attacks continue to be a substantial menace to cybersecurity, presenting dangers to both individuals and companies. Phishers utilize diverse strategies to trick users into revealing sensitive information, such as login passwords, financial data, or personal information. Conventional rule-based techniques used to identify phishing assaults may have difficulties in keeping up with the ever-changing tactics used by malevolent individuals. Also, the nature of these attacks makes it difficult for humans to distinguish between legitimate and phishing attacks.

The objective of this research is to create a proficient system for detecting phishing attacks by utilizing a combination of machine learning techniques under an ensemble classifier. Ensemble method utilizes a combination of numerous base learners to enhance the accuracy of predictions and improve overall performance in generalization. Fig. 1 shows the problem statement formation of this research.

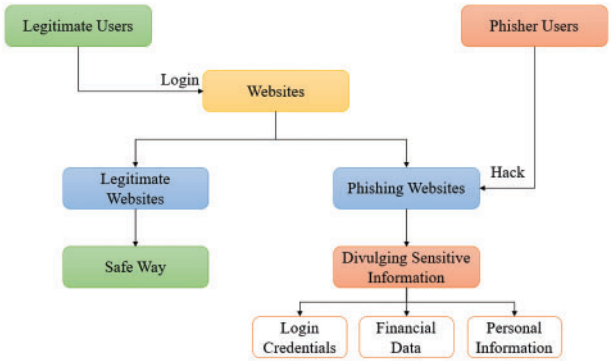

Figure 1: Problem statement formation

The reminder for this paper is organized as following sections: Section 2 presents the literature review. Section 3 describes the methodology used in terms of dataset, machine learning algorithm, and performance metrics. Section 4 describes the proposed ensemble learning approach. Section 5 explains and illustrates the experimental results for the two datasets. Section 6 discusses the findings. Finally, the conclusion of this paper and future work.

2 Literature Review

Table 1 shows the summary of the previous articles that are related to this study in terms of the machine learning algorithms used, the phishing dataset, preprocessing steps, evaluation metrics, and the performance results. Abutaha et al. [12] employed four machine learning methods: Gradient Boosting, Random Forest, Neural Network, and Support Vector Machine to detect a URL phishing. The dataset consisted of 1,056,937 URLs that were labelled as either phishing or legal. The dataset was processed to create 22 distinct features, which were subsequently reduced to a smaller set using several feature reduction approaches. They applied data preprocessing to the dataset, such as remove 14,786 duplicate records and handle missing values. To assess the algorithms performance, they used five evaluation metrics: precision, recall, F1-score, false positive rate, and accuracy. The findings demonstrated that Support Vector Machine attained the highest level of accuracy in identifying the examined URLs, with an accuracy value of 97.3%. The method can be integrated with add-on/middleware functionalities in Internet browsers to notify online users whenever they attempt to enter a phishing website only based on its URL. Abu-Nimeh et al. [13] utilized ML methods: support vector machines (SVM), classification and regression trees (CART), logistic regression (LR), Bayesian additive regression trees (BART), random forests (RF), and neural networks (NNet), to

forecast phishing websites. The dataset utilized consisted of 2889 websites, comprising both phishing and legitimate websites. This dataset was employed in the training and testing processes, utilizing a total of 43 characteristics. Four assessment criteria were employed to analyze the performance of the algorithms: area under the ROC curve (AUC), precision, F1-score, and recall, they have shown that the RF algorithm achieved the highest performance in predicting phishing websites, with an AUC of 0.9442.

Table 1: Previous papers summarization

Ref.	Year	Algorithms	Dataset	Evaluation metrics	Results
Abutaha et al. [12]	2021	Random Forest, Gradient Boosting, Neural Network, and Support Vector Machine	1,056,937 URLs	Precision, Recall, F1-score, False Positive Rate, and Accuracy	Accuracy of support vector machine is 97.3%
Abu-Nimeh et al. [13]	2007	LR, CART, BART, SVM, RF, and NNet	2889 websites	AUC, Precision, Recall, and F1-score	AUC of RF = 0.9442
Samad et al. [14]	2020	Multi-layer approach, SVM, and Random Forest	3000 websites	Precision, Recall, F1-score, and Accuracy	RF accuracy of 97%
Yadav et al. [15]	2021	J48 Random Forest Logistic regression	1500 websites	Accuracy, and Precision	Accuracy of RF = 97.4%
Medelbekov et al. [16]	2023	Logistic Regression, Random Forest, Support Vector machine, KNN, and KNN k-Fold Cross Validation	11,055 websites	Precision, Recall, F1-score, and Accuracy	Accuracy of random forest is 0.967
Shahrivari et al. [17]	2020	Logistic Regression, KNN, SVM, Decision Tree, RF, AdaBoost, Gradient Boosting, and Artificial Neural Networks	11,055 websites	Precision, Recall, F1-score, and Accuracy	0.972682 of RF accuracy
Alazaidah et al. [18]	2024	Bayes, Functions, Lazy, Meta, Rules, and Tress	11,055 and 1353 websites	Accuracy, True Positive, False Positive, Precision, Recall, F1-score, MCC, and ROC Area	Accuracy of RF is 97.25%

Samad et al. [14] introduced a multi-layer strategy to reduce the consequences of spear-phishing attacks, which are highly successful phishing attacks because of the social and psychological obstacles they provide. The proposed methodology utilized both the textual content and accompanying files of an email in order to combat phishing campaigns. They utilized sentiment analysis techniques, specifically SVM and RF classifiers, to categorize websites as either spam or non-spam. This approach yielded impressive accuracy rates. They utilized a dataset sourced from the Kaggle platform, comprising 3000 websites that were categorized as either spam or non-spam. In addition, they utilized Latent Dirichlet Allocation (LDA) for topic modeling in order to identify the prevailing topics within the dataset. They have demonstrated that the RF algorithm achieved the highest accuracy of 97% in

comparison to the other algorithms during the detecting procedure. In their study, Yadav et al. [15] examined the application of machine learning algorithms for the identification of phishing websites. Their concentration was on feature selection, a process that entails analyzing and reducing a complicated data set to a smaller dimension by considering several attributes. They utilized a dataset consisting of 1500 data tuples obtained from the SPAMASSASIAN corpus, along with a separate validation dataset comprised of websites sourced from Gmail users. The data was preprocessed by techniques such as HTML parsing, data cleansing, stemming, stop word deletion, and tokenization. The study utilized three machine learning classification techniques, including J48, random forest, and logistic regression, to forecast the occurrence of phishing and non-phishing websites. The random forest method demonstrated superior performance in the prediction process, achieving a precision rate of 99% and an accuracy rate of 97.4%.

Medelbekov et al. [16] conducted an independent analysis and created a model to identify phishing sites. The researchers utilized a phishing dataset comprising 30 distinct characteristics and a total of 11,055 instances. These instances were categorized into three classes: 0 denoting suspicious, −1 denoting legitimate, and 1 denoting phishing. The model was trained using five algorithms: LR, SVM, RF, K-nearest neighbors, and KNN k-Fold Cross Validation. Four assessment measures were employed to analyze the performance of the algorithms: accuracy, recall, F1-score, and precision. The Random Forest algorithm demonstrated superior performance in the detection process when compared to other methods. Specifically, it achieved an accuracy of 0.967, precision of 0.90, recall of 0.946, and a F1-score of 0.963. Shahrivari et al. [17] used machine learning algorithms (LR, KNN, SVM, Decision Tree, RF, AdaBoost, Gradient Boosting, and Artificial Neural Networks) to predict the phishing websites based on a phishing dataset. They used the phishing website dataset that contains 11,055 websites with 32 attributes divided into legitimate, and phishing. Then, they used four classification metrics to evaluate the performance of these algorithms in prediction process: accuracy, recall, F1-score, and precision. They have shown that the RF gave the best performance results in the prediction phishing websites as a follow: 0.972682 of accuracy, 0.981484 of precision, 0.969852 of recall, and 0.975622 of F1-score.

Alazaidah et al. [18] determined the most effective classifier for detecting phishing out of twenty-four different classifiers representing six learning methodologies in machine learning. They are utilizing two datasets pertaining to Phishing with distinct properties. The initial dataset is a binary classification dataset. The dataset has 30 integer features, with the majority of them being binary. There are 11,055 instances in this collection. The second dataset is a multiclass dataset with three class labels. It has 9 integer-type features and 1353 instances. The classifiers are divided into six groups: Bayes (Bayes Net, Naïve Bayes, Naïve Bayes Updateable), Functions (Logistic Regression, Multilayer Perceptron, Simple Logistic, SMO-C), Lazy (IBk, K-Star, LWL), Meta (AdaBoostM1, Filtered Classifier, LogitBoost, MultiClass Classifier, Random Committee), Rules (Decision Table, JRip, PART, Zero), and Trees (Decision Stump, J48, LMT, Random Forest, and Random Tree). They assessed the performance of these classifiers using eight evaluation metrics: Accuracy, True Positive, False Positive, Precision, Recall, F1-score, MCC, and ROC Area. They showed that Random Forest, Filtered Classifier, and J48 classifiers were the most effective in identifying phishing websites.

3 Methodology

This section presents the methodology used in this paper, which contains three steps: the datasets used, the machine learning algorithms used to build the models, and the performance metrics applied to assess the algorithms performance. In the first step, we describe the datasets in terms of the number

of features with brief information, the number of instances, and the frequency of websites divided into two groups: legitimate and phishing. In the second step, the building models are explained, and the performance metrics that are used are presented in the third step: accuracy, F1-score, recall, and precision.

3.1 Phishing Datasets

We used two phishing websites datasets that contain many characteristics related to the websites. Each dataset has different features and number of instances.

1) First Dataset[1]: This dataset contains 11,055 instances with 32 features that is related to the website's information divided into two groups: legitimate, and phishing, as shown in Table 2.

Table 2: Features with description–first dataset

No.	Feature	No.	Feature
1	Index	17	SFH
2	having_IPhaving_IP_Address	18	Submitting_to_email
3	URLURL_Length	19	Abnormal_URL
4	Shortining_Service	20	Redirect
5	having_At_Symbol	21	on_mouseover
6	double_slash_redirecting	22	RightClick
7	Prefix_Suffix	23	popUpWidnow
8	having_Sub_Domain	24	Iframe
9	SSLfinal_State	25	age_of_domain
10	Domain_registeration_length	26	DNSRecord
11	Favicon	27	web_traffic
12	Port	28	Page_Rank
13	HTTPS_token	29	Google_Index
14	Request_URL	30	Links_pointing_to_page
15	URL_of_Anchor	31	Statistical_report
16	Links_in_tags	32	Result

Fig. 2 shows the frequency of the websites divided into two groups: legitimate, and phishing. Fig. 3 shows the snapshots of the first dataset.

2) Second Dataset[2]: This dataset contains 10,000 instances with 50 features hat is related to the website's information divided into two groups: legitimate, and phishing, as shown in Table 3.

[1] https://www.kaggle.com/code/akashkr/phishing-url-eda-and-modelling/input (accessed on 01/05/2024).
[2] https://www.kaggle.com/datasets/amj464/phishing (accessed on 01/05/2024).

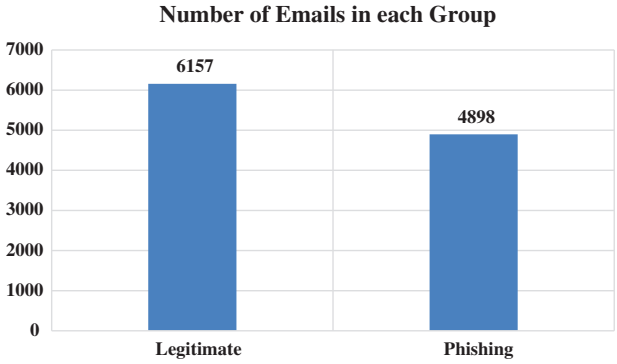

Figure 2: Dataset frequency–first dataset

index	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain
1	-1	1	1	1	-1	-1	-1
2	1	1	1	1	1	-1	0
3	1	0	1	1	1	-1	-1
4	1	0	1	1	1	-1	-1
5	1	0	-1	1	1	-1	1
6	-1	0	-1	1	-1	-1	1
7	1	0	-1	1	1	-1	-1
8	1	0	1	1	1	-1	-1
9	1	0	-1	1	1	-1	1
10	1	1	-1	1	1	-1	-1
11	1	1	1	1	1	-1	0
12	1	1	-1	1	1	-1	1
13	-1	1	-1	1	-1	-1	0
14	1	1	-1	1	1	-1	0
15	1	1	-1	1	1	1	-1
16	1	-1	-1	-1	1	-1	0
17	1	-1	-1	1	1	-1	1
18	1	-1	1	1	1	-1	-1
19	1	1	1	1	1	-1	-1
20	1	1	1	1	1	-1	-1
21	1	0	-1	1	1	-1	0
22	1	0	1	1	1	-1	0
23	1	1	1	1	1	-1	-1
24	1	1	1	1	1	-1	1
25	1	-1	-1	-1	1	-1	1

Figure 3: First dataset snapshot

Table 3: Features with description–second dataset

No.	Feature	No.	Feature	No.	Feature	No.	Feature	No.	Feature
1	Id	11	NumPercent	21	HttpsIn Hostname	31	InsecureForms	41	IframeOr Frame
2	NumDots	12	NumQuery Components	22	Hostname Length	32	RelativeForm Action	42	MissingTitle
3	Subdomain Level	13	NumAmpersand	23	PathLength	33	ExtForm Action	43	ImagesOnly InForm
4	PathLevel	14	NumHash	24	QueryLength	34	Abnormal FormAction	44	Subdomain LevelRT

(Continued)

Table 3 (continued)

No.	Feature	No.	Feature	No.	Feature	No.	Feature	No.	Feature
5	UrlLength	15	NumNumeric Chars	25	DoubleSlash InPath	35	PctNullSelf RedirectHyper-links	45	UrlLengthRT
6	NumDash	16	NoHttps	26	NumSensitive Words	36	FrequentDomain NameMismatch	46	PctExtResource UrlsRT
7	NumDashIn Hostname	17	RandomString	27	EmbeddedBrand Name	37	FakeLinkIn Sta-tusBar	47	Abnormal ExtFormAc-tionR
8	AtSymbol	18	IpAddress	28	PctExtHy perlinks	38	RightClick Disabled	48	ExtMetaScript LinkRT
9	TildeSymbol	19	DomainIn Sub-domains	29	PctExtResource Urls	39	PopUpWindow	49	PctExtNullSelf RedirectHyper-linksRT
10	NumUnderscore	20	DomainInPaths	30	ExtFavicon	40	SubmitInfoTo Email	50	CLASS_LABEL

Fig. 4 shows the frequency of the websites of two class labels for the second dataset and Fig. 5 shows the snapshot of this dataset.

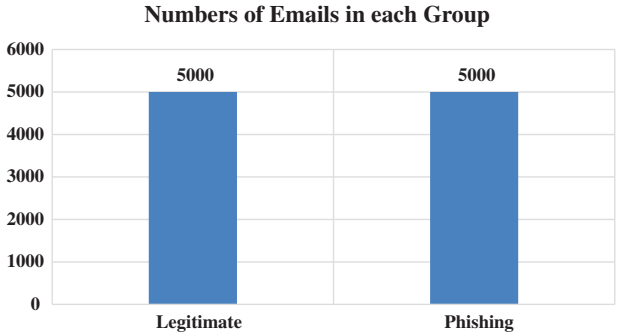

Figure 4: Dataset frequency–second dataset

3.2 Building Models

We applied seven classification machine learning algorithms on each dataset to predict the type of website whether legitimate or phishing. Machine learning algorithms are computational models that enable computers to discern patterns and make predictions or decisions based on data, without requiring explicit programming. These algorithms are fundamental to contemporary artificial intelligence and are applied in several domains such as image and audio recognition, natural language processing, recommendation systems, fraud detection, phishing websites and autonomous vehicles.

3.2.1 Random Forest (RF) Algorithm

Random forests utilize ensemble learning, a technique that mixes many decision trees to generate predictions for both classification and regression tasks. Ensemble learning has various advantages in machine learning, such as enhanced performance, resilience, and the capability to tackle intricate

issues. Random forests employ ensemble learning methodologies to augment their prediction capability. In the classification task, the RF calculate the prediction average for all trees, while it is computing the mean for the regression task [19]. Decision trees are the essential building elements that form the core of random forests. Decision trees are hierarchical models that use binary splits on features to produce predictions. Every division partitions the data into smaller subsets according to specific criteria, ultimately resulting in the prediction of a target variable [20].

id	NumDots	SubdomainLevel	PathLevel	UrlLength	NumDash	NumDashInHostname	AtSymbol	TildeSymbol	NumUnderscore
1	3	1	5	72	0	0	0	0	0
2	3	1	3	144	0	0	0	0	2
3	3	1	2	58	0	0	0	0	0
4	3	1	6	79	1	0	0	0	0
5	3	0	4	46	0	0	0	0	0
6	3	1	1	42	1	0	0	0	0
7	2	0	5	60	0	0	0	0	0
8	1	0	3	30	0	0	0	0	0
9	8	7	2	76	1	1	0	0	0
10	2	0	2	46	0	0	0	0	0
11	5	4	2	64	1	1	0	0	0
12	2	0	2	47	0	0	0	0	0
13	2	1	2	61	1	1	0	0	0
14	2	1	3	35	0	0	0	0	0
15	2	1	2	60	1	1	0	0	0
16	3	0	4	73	0	0	0	0	0
17	3	0	5	50	0	0	0	1	0
18	3	1	2	59	1	1	0	0	0
19	2	0	3	28	0	0	0	0	0
20	1	0	4	59	0	0	0	0	0
21	1	0	4	32	0	0	0	0	0
22	5	1	2	52	0	0	0	0	0
23	2	1	6	62	1	0	0	0	0
24	1	0	10	105	2	0	0	0	0
25	4	1	2	55	0	0	0	0	0

Figure 5: Second dataset snapshot

Random forests can enhance accuracy, mitigate overfitting, and effectively address intricate problems by amalgamating the predictions of many decision trees. By utilizing an aggregation of decision trees, random forests are able to effectively capture many facets of the data, resulting in more resilient predictions. In the Eq. (1), the RF compute the final prediction denoted by y, $h_i(x)$ is the prediction for each decisions tree, and the N refers to the number of trees.

$$y(x) = \frac{1}{N} \sum_{i=1}^{N} h_i(x) \tag{1}$$

3.2.2 Decision Tree Algorithm (DT)

Decision trees are utilized for the purpose of classifying and regressing jobs, offering models that are straightforward and comprehensible. Decision tree is a hierarchical model utilized in decision support systems to illustrate actions and their possible outcomes, taking into account chance events, resource expenditures, and utility. The tree structure consists of a central root node, which is connected to other nodes through branches [21]. These nodes might be internal nodes or leaf nodes, creating a hierarchical and tree-like arrangement. The idea of this algorithm in the prediction process is it splitting

the data into groups and sub groups. These groups have a root node that it selected based on many methods like entropy, information gain, gain ratio, and gini index, in this paper, we select the entropy as a main method in splitting process. The formula of this method is shown in Eq. (2) [22]:

$$H(S) = -p + log_2(p+) - p - log_2(p-) \tag{2}$$

where, $H(S)$ is the entropy of the dataset S, $p+$ is the proportion of positive instances (samples belonging to the positive class) in the dataset S, and $p-$ is the proportion of negative instances (samples belonging to the negative class) in the dataset S.

3.2.3 Multilayer Perceptron Algorithm (MLP)

A nodes, commonly referred to as neurons or perceptrons. The neural network architecture is a feedforward design, indicating that information is transmitted in a unidirectional manner from the input layer, to the hidden layers, and finally to the output layer. MLPs are extensively employed for diverse tasks such as pattern recognition, classification, regression, and other applications. The MLP contains three layers: input, hidden and output [23,24]. 1) Input Layer: The first layer of the network, where the input features are fed into the network. Each node in this layer represents a feature of the input data. 2) Hidden Layers: Intermediate layers between the input and output layers. These layers are responsible for learning complex patterns and representations from the input data. 3) Output Layer: The final layer of the network, which produces the output or prediction. The number of nodes in this layer depends on the type of task (e.g., binary classification, multiclass classification, regression).

The formula of this method is shown in Eq. (3):

$$a_j = f\left(\sum_{i=1}^{n} w_{ij} \,.\, x_i + b_j\right) \tag{3}$$

where, a_j is the output of neuron j in a particular layer, $f()$ is the activation function applied to the weighted sum of inputs, w_{ij} is the weight of the connection between the ith neuron in the previous layer and the jth neuron in the current layer, x_i is the output of the ith neuron in the previous layer, and bj is the bias associated with neuron j.

3.2.4 eXtreme Gradient Boosting Algorithm (XGB)

XGBoost, also known as eXtreme Gradient Boosting, is a machine learning technique that falls under the category of ensemble learning. Supervised learning tasks, such as regression and classification, are now fashionable. XGBoost constructs a prognostic model by amalgamating the prognostications of numerous independent models, frequently decision trees, in an iterative fashion. The method operates by progressively including weak learners into the ensemble, with each subsequent learner specifically targeting the rectification of faults produced by the preceding ones. The system employs a gradient descent optimization method to minimize a predetermined loss function while undergoing training. The XGBoost algorithm possesses several notable characteristics [25]. It excels at handling intricate relationships within data, use regularization approaches to mitigate overfitting, and incorporates parallel processing for enhanced computational efficiency. XGBoost is extensively utilized in several fields owing to its exceptional predicted accuracy and adaptability over a wide range of datasets. The formula of this method is shown in Eq. (4) [26]:

$$Objective = \sum_{i=1}^{n} \left(y_i + \hat{y}_i\right) \sum_{k=1}^{K} \Omega\left(f_k\right) \tag{4}$$

where, n is the number of training instances, K is the number of trees in the model, f_k represents the kth tree, and $\Omega(f_k)$ is the regularization term which penalizes complexity of the model to prevent overfitting.

3.2.5 AdaBoost Algorithm

Within the realm of machine learning models, there are a variety of possibilities to select from, and AdaBoost is among them. It belongs to the family of advanced ensemble learning models that trains a sequence of weak classifiers on distinct subsets of the training data in an iterative manner. In each iteration, the algorithm increases the weights of the samples that were categorized incorrectly in the previous iteration. This allows the system to priorities the more difficult examples. This approach enables the succeeding weak classifiers to allocate greater focus to the previously misclassified examples, hence enhancing their performance. Adaptive boosting is a technique used to minimize the error of a machine-learning algorithm. It achieves this by combining multiple weak machine-learning models into a single, more powerful model. The formula of this method is shown in Eq. (5) [27,28]:

$$\mathrm{H}\left(\mathrm{x}\right) = sign\left(\sum_{t=1}^{T} \alpha_t \,.\, h_t(x)\right) \tag{5}$$

where, H(x) is the strong learner, $h(x)$ is the weak learner, and $sign\,(\cdot)$ is the sign function which returns -1 for negative values and 1 for non-negative values.

3.2.6 Gradient Boosting (GB) Algorithm

Gradient Boosting is a robust machine learning method employed for solving regression and classification tasks. It is a member of the ensemble learning methods category, in which numerous weak learners (models that perform somewhat better than random guessing) are integrated to form a powerful learner. Gradient Boosting constructs models in a progressive manner, where each subsequent model is designed to specifically address the faults generated by the preceding models, The formula of this method is shown in Eq. (6) [29]:

$$F\left(X\right) = \sum_{m=1}^{M} o_m . h_m\left(x\right) \tag{6}$$

where, $F(X)$ is the strong learner, $h(x)$ is the weak learner, and om is the optimal step size (learning rate).

3.3 Performance Metrics

The experimental results for the aforementioned machine learning are based on four evaluation metrics: precision, recall, F1-score, and accuracy. The formula for each metric and shallow explanation are shown below, where FN refers to False Negative, TP refers to True Positive, TN refers to True Negative and FP refers to False Positive [30]:

- Accuracy is computed by divided the ratio of samples that are correctly predicted to the total number of samples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

- Precision is computed by divided the ratio of positive samples that are correctly predicted to the total number of expected positive samples.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

- The ratio of true positive samples in the dataset to the total number of predicted positive samples is referred to as the Recall.

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

- F1-score is the average of Recall and Precision.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{10}$$

4 Proposed Ensemble Learning Approach

4.1 Proposed Approach Overview

The proposed approach used in this paper is a Voting classifier that combined the best three machine leaning algorithms: RF, XGB and MLP. This classifier is applied on thew two phishing datasets to predict the type of the website if it normal or malicious.

4.2 Preprocessing

Data preprocessing is an essential stage in machine learning and data analysis. Data preprocessing include the tasks of cleansing, converting, and structuring unprocessed data into a usable format for model training or analysis. In this paper, we applied a normalization method as a preprocessing step. We relied on a Python library to apply this step, which is called MinMaxScaler. MinMaxScaler is a preprocessing method employed to rescale numerical characteristics to a predetermined range, typically ranging from 0 to 1, as shown in Eq. (1), where x is the required sample to be normalized, and i is the index in the dataset [31].

$$MinMaxScalar = \frac{x_i - \min(x)}{\max(i) - \min(i)} \tag{11}$$

This is accomplished by converting the data using the lowest and highest values of each characteristic. The MinMaxScaler class from the sklearn.preprocessing module in the scikit-learn library can be utilized in Python. Figs. 6 and 7 illustrate the examples from two dataset after applied this step.

index	having_IPhaving_IP_Address	URLURL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain
0.000000	0.0	1.0	1.0	1.0	0.0	0.0	0.0
0.000090	1.0	1.0	1.0	1.0	1.0	0.0	0.5
0.000181	1.0	0.5	1.0	1.0	1.0	0.0	0.0
0.000271	1.0	0.5	1.0	1.0	1.0	0.0	0.0
0.000362	1.0	0.5	0.0	1.0	1.0	0.0	1.0
...	...	...	...	...	...	...	...
0.999638	1.0	0.0	1.0	0.0	1.0	1.0	1.0
0.999729	0.0	1.0	1.0	0.0	0.0	0.0	1.0
0.999819	1.0	0.0	1.0	1.0	1.0	0.0	1.0
0.999910	0.0	0.0	1.0	1.0	1.0	0.0	0.0
1.000000	0.0	0.0	1.0	1.0	1.0	0.0	0.0

Figure 6: First dataset–applied MinMaxScalar

id	NumDots	SubdomainLevel	PathLevel	UrlLength	NumDash	NumDashInHostname	AtSymbol	TildeSymbol	NumUnderscore
0.0000	0.10	0.071429	0.277778	0.248963	0.000000	0.0	0.0	0.0	0.000000
0.0001	0.10	0.071429	0.166667	0.547718	0.000000	0.0	0.0	0.0	0.111111
0.0002	0.10	0.071429	0.111111	0.190871	0.000000	0.0	0.0	0.0	0.000000
0.0003	0.10	0.071429	0.333333	0.278008	0.018182	0.0	0.0	0.0	0.000000
0.0004	0.10	0.000000	0.222222	0.141079	0.000000	0.0	0.0	0.0	0.000000
...	...	...	...	...	...	...	...	...	...
0.9996	0.10	0.071429	0.055556	0.157676	0.000000	0.0	0.0	0.0	0.000000
0.9997	0.05	0.071429	0.222222	0.195021	0.018182	0.0	0.0	0.0	0.000000
0.9998	0.05	0.071429	0.222222	0.186722	0.000000	0.0	0.0	0.0	0.000000
0.9999	0.10	0.071429	0.055556	0.153527	0.000000	0.0	0.0	0.0	0.000000
1.0000	0.10	0.071429	0.111111	0.165975	0.054545	0.0	0.0	0.0	0.000000

Figure 7: Second dataset–applied MinMaxScalar

4.3 Training Machine Learning Classifiers

The training data contained labels indicating whether a specific output corresponded to an expected class. The primary objective is to train the learning model to accurately identify the location of unfamiliar data by comparing it to the reference data. Nevertheless, we discovered that in several instances, a solitary learning model may have yielded the optimal outcomes or the least number of errors. Consequently, we implemented an ensemble learning approach that entailed creating many hypotheses based on the training data and integrating them to accurately identify the position of the sample. By amalgamating the decisions from many models, this strategy significantly improved the overall efficiency of the model, leading to heightened accuracy in the outputs. Furthermore, this method resulted in a stable and more resilient model compared to separate models. In order to construct our ensemble model, we methodically carry out the training process for each machine learning classifier that comprises our ensemble. The classifiers mentioned in Section 3.2 encompass the Random Forest, XGBoost, Gradient Boosting, DT, MLP, and AdaBoost Classifiers. The varied structures, hyperparameters, and distinct capabilities of each classifier play a crucial role in facilitating a comprehensive learning process. These trained classifiers are subsequently used as the foundation

for the ensemble approach. This strategy enhances the ensemble's ability to detect malicious actions in loT situations by using Weighted Voting.

4.4 Ensemble Voting Classifier

A Voting classifier is a machine learning model that is trained on an ensemble of many models and makes predictions by selecting the class with the highest probability among the models. The Voting classifier combines the results of multiple classifiers and predicts the output class based on the majority vote, as shown in Fig. 8. The concept involves consolidating individual specialized models and determining their accuracy. Instead of constructing separate models and evaluating their correctness individually, we develop a unified model that trains on these models and predicts the output by considering the majority vote from each model for each output class. Eq. (12) shows the formula of Voting classifier that combined three machine learning algorithms: RF, XGB, and MLP, where: P_i is the prediction for each classifier, and the W_i denotes the wight assigned to the prediction for the classifier [32,33]. The reason behind the selection process of these classifiers was they gave the best performance results individually compared with the others. The strategy of combining these classifiers is each classifier gives a prediction value and the prediction class is determined by considering the majority vote.

$$FinalPrediction = \sum_{i=1}^{n} W_i * P_i \tag{12}$$

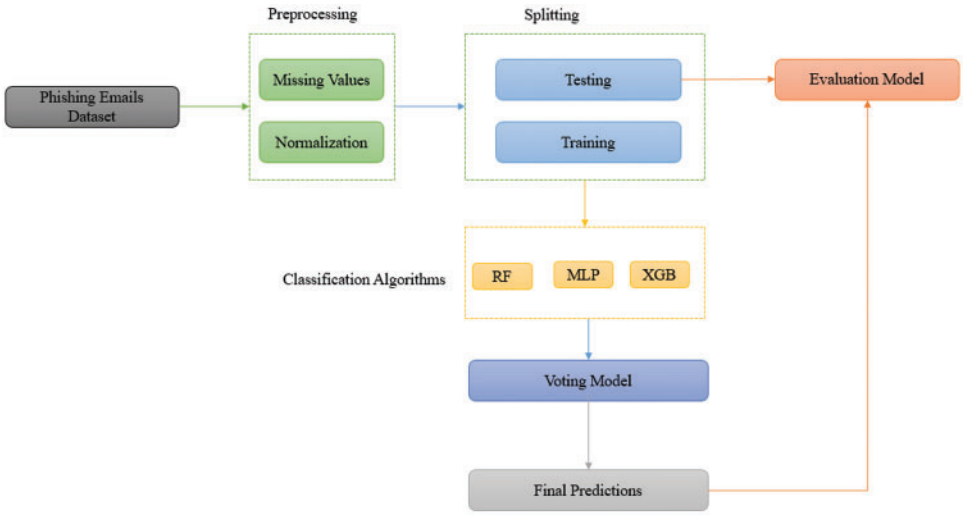

Figure 8: Flow chart of Voting algorithm

5 Evaluation and Results

This section explained our findings by applied seven machine learning techniques on two datasets based on the above evaluation metrics: recall, accuracy, F1-score, and precision. Then, we analysis the experiments setup for each algorithm that include the parameters used with their values.

5.1 Experimental Setup

This section presents experimental setup, as shown in Table 4, for the machine learning algorithms used in this paper in order to detect the phishing attacks in phishing websites datasets.

Table 4: Machine learning algorithms–experimental setup

Algorithm	Parameters	Value
RF	No. of estimators	100
	Criterion	Gini
	Max depth	None
	Random state	42
DT	Criterion	Entropy
	Max depth	None
	Random state	42
MLP	Solver	lbfgs
	Hidden layer sizes	100
	Activation function	relu
	Random state	42
XGB	Learning rate	0.1
	No. of estimators	100
	Random state	42
AdaBoost	Learning rate	1.0
	No. of estimators	50
	Algorithm	SAMMER.R
	Random state	42
GB	Learning rate	0.2
	No. of estimators	200
	Random state	42
Voting classifier	Estimators	RF, XGB, MLP
	Vote type	Hard

Before the dataset fed to machine learning algorithms, it must be divided into two groups: testing and training. The training dataset is used to build the models based on these algorithms, while the testing dataset is used to assess the models' performance that are built. In this paper, the ratio of the training and testing is as follows: 0.90 of whole dataset is used in training process and the remainder is used in testing process.

5.2 Experimental Results

This section presents the results that obtained in this paper after applied the six machine leaning algorithms: RF, DT, MLP, XGB, AdaBoost, and GB in two phishing datasets. We conducted these experiments on Anaconda environment for binary classification task. Then, we compared the results with Voting classifier based on four evaluation metrics: accuracy, recall, F1-score, and precision.

5.2.1 First Dataset–Results

Fig. 9 and Table 5 show the findings for each algorithm in each dataset. The Voting has the best performance results in prediction process compared with the rest of algorithms as follows: accuracy = 0.978, precision = 0.975, recall = 0.987, and F1-score = 0.981.

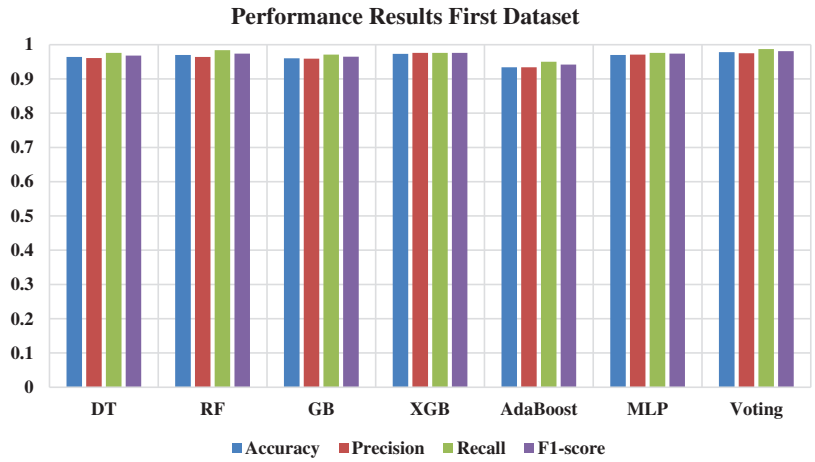

Figure 9: Performance results–first dataset

Table 5: Performance results–first dataset

Model	Accuracy	Precision	Recall	F1-score
DT	0.964	0.961	0.976	0.968
RF	0.970	0.964	0.984	0.974
GB	0.960	0.959	0.971	0.965
XGB	0.973	0.976	0.976	0.976
AdaBoost	0.934	0.934	0.950	0.942
MLP	0.970	0.971	0.976	0.974
Voting	0.978	0.975	0.987	0.981

5.2.2 Second Dataset–Results

Fig. 10 and Table 6 show the findings for each algorithm in each dataset. All the algorithms gave the same results in four evaluation metrics, which indicates the each of them can effectively accomplish the prediction process.

6 Discussion

This section discusses and explains the findings obtained from the two experiments to detect phishing attacks in two phishing datasets. These findings are obtained based on seven machine learning algorithms: RF, DT, MLP, XGB, AdaBoost, GB, and a Voting classifier. The Voting classifier outperformed the other machine learning algorithms in this study and the previous studies in terms of accuracy, recall, F1-score, and precision in the first dataset. As shown in Table 7, Sindhu et al. [19]

applied five machine algorithms to the first dataset, and the RF gave the best results with an accuracy of 0.967. Other papers, like Pandey et al. [20] used 8 algorithms, and Zhu et al. [21] used 24 classifiers, and the results were 97.26 and 97.25, respectively. In this study, we obtained a higher accuracy in the same dataset with 97.8. In the second dataset, to our knowledge, we did not find any study applying machine learning or deep learning to it. We gave higher results in the second dataset for all seven algorithms based on four evaluation metrics.

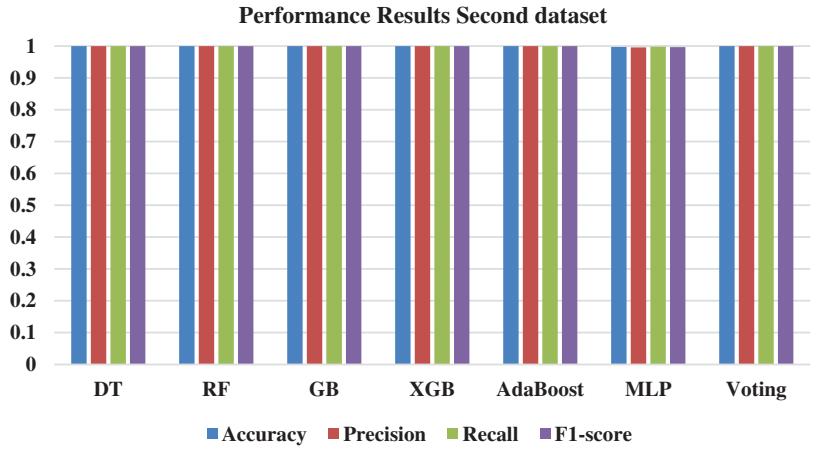

Figure 10: Performance results–second dataset

Table 6: Performance results–second dataset

Model	Accuracy	Precision	Recall	F1-score
DT	1	1	1	1
RF	1	1	1	1
GB	1	1	1	1
XGB	1	1	1	1
AdaBoost	1	1	1	1
MLP	0.997	0.995943	0.997967	0.996954
Voting	1	1	1	1

Table 7: Comparison between our work and previous works

Ref.	Year	Algorithms	Results
Medelbekov et al. [16]	2023	Logistic regression RF Support vector machine KNN KNN k-Fold cross validation	Accuracy of RF is 0.967

(Continued)

Table 7 (continued)

Ref.	Year	Algorithms	Results
Shahrivari et al. [17]	2020	Logistic regression KNN Support vector machine Decision tree RF AdaBoost Gradient Boosting Artificial neural networks	RF accuracy = 0.973
Alazaidah et al. [18]	2024	Bayes Functions Lazy Meta Rules Tress	Accuracy of RF is 97.25%
Our approach	2024	RF DT MLP GB AdaBoost XGB Ensemble learning approach	Ensemble approach in first dataset: 97.8, and 100 in the second

Based on our findings, we achieved the following goals:

1.1 We built a robust ensemble learning approach based on three algorithms (RF, XGB, and MLP) that gave the best detection accuracy in both datasets compared with the other algorithms in this study or in previous works.
1.2 In the detection process, we took less time compared with previous work, and we used the same computer settings.
1.3 The false negative and false positive rates are decreased by obtaining a higher accuracy value based on the robust approach that was built into both datasets.

The contributions of this paper are summarized in the below points:

1.1 Developing an ensemble learning methodology utilizing resilient machine learning algorithms to differentiate between authentic and phishing websites within a larger phishing dataset.
1.2 Achieving a high level of precision in distinguishing between genuine and phishing websites in order to minimize both incorrect identifications and missed detections. The goal is to ensure that the detection system effectively recognizes phishing attacks while minimizing the likelihood of erroneously labeling legitimate websites.
1.3 To minimize the occurrence of false positives in phishing detection, hence avoiding the incorrect identification of legitimate websites as phishing websites, which can lead to user

annoyance and undermine confidence in the detection system. The objective is to find a balance between sensitivity and specificity, maximizing the accuracy of detection while decreasing the occurrence of false positives.

1.4 Improved Detection Accuracy: Machine learning algorithms have the ability to analyze large amounts of data to identify subtle patterns and characteristics that indicate phishing attacks. This leads to enhanced detection accuracy in comparison to traditional rule-based or heuristic methods.

7 Conclusion and Future Work

Phishing, an online scam in which individuals are tricked into divulging important personal and financial details, presents a substantial threat to both consumers and Internet-based institutions. Evidence suggests a continuous increase in phishing attacks. Furthermore, these deceptive techniques are also growing more complex, making them more difficult to detect. Therefore, it is crucial to employ advanced algorithms to tackle this problem. Machine learning is an exceptionally efficient method for detecting and revealing these detrimental behaviors. Machine learning algorithms can detect shared attributes in the majority of phishing attacks. This paper utilizes seven machine learning methods to analyze two phishing datasets, aiming to classify the type of websites and establish its normality. Subsequently, we will employ the normalization procedure on the dataset to standardize the range of all the features to a uniform scale. The results indicated that the XGB algorithm demonstrates superior performance in the prediction process, achieving an accuracy of 0.978, precision of 0.975, recall of 0.987, and F1-score of 0.981 in the initial dataset. In the second dataset, all the algorithms yielded identical results across four evaluation measures, suggesting their same effectiveness in performing the prediction procedure. As a future work, we plan to: 1) Apply the deep learning algorithms on the aforementioned dataset and another machine learning. 2) Apply feature selection methods, 3) the ensemble model will be generalized, and 4) study the impact of another data preprocessing techniques.

Acknowledgement: Nisreen Innab would like to express sincere gratitude to AlMaarefa University, Riyadh, Saudi Arabia, for supporting this research. Ahmed, Mohamed and Marwan extend sincere thanks and appreciation to the administration of King Faisal University in the Kingdom of Saudi Arabia for providing all forms of support to the university's faculty members, especially in the field of scientific research.

Funding Statement: This article is funding from Deanship of Scientific Research in King Faisal University with Grant Number KFU 241085.

Author Contributions: Conceptualization: Ahmed Abdelgader Fadol Osman; methodology: Nisreen Innab; formal analysis: Mohammed Awad Mohammed Ataelfadiel and Marwan Abu-Zanona; original draft preparation: Farah H. Zawaideh, Mouiad Fadeil Alawneh; review and editing: Bassam Mohammad Elzaghmouri; visualization: Bassam Mohammad Elzaghmouri and Ahmed Abdelgader Fadol Osman; project administration: Nisreen Innab and Marwan Abu-Zanona . All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data is openly available in a public repository in section datasets.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Villamil, C. Hernández, and G. Tarazona, "An overview of Internet of Things," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 5, pp. 2320–2327, 2020. doi: 10.12928/telkomnika.v18i5.15911.

[2] B. B. Gupta and M. Quamara, "An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols," *Concurr. Comput.*, vol. 32, no. 21, pp. e4946, 2020. doi: 10.1002/cpe.4946.

[3] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments," *Energy Rep.*, vol. 7, no. 8, pp. 8176–8186, 2021. doi: 10.1016/j.egyr.2021.08.126.

[4] K. T. Smith, L. M. Smith, M. Burger, and E. S. Boyle, "Cyber terrorism cases and stock market valuation effects," *Inf. Comput. Secur.*, vol. 31, no. 4, pp. 385–403, 2023. doi: 10.1108/ICS-09-2022-0147.

[5] I. H. Sarker, "Deep cybersecurity: A comprehensive overview from neural network and deep learning perspective," *SN Comput. Sci.*, vol. 2, no. 3, pp. 154, 2021. doi: 10.1007/s42979-021-00535-6.

[6] D. Bhamare, M. Zolanvari, A. Erbad, R. Jain, K. Khan and N. Meskin, "Cybersecurity for industrial control systems: A survey," *Comput. Secur.*, vol. 89, pp. 101677, 2020. doi: 10.1016/j.cose.2019.101677.

[7] H. Kavak, J. J. Padilla, D. Vernon-Bido, S. Y. Diallo, R. Gore and S. Shetty, "Simulation for cybersecurity: State of the art and future directions," *J. Cybersecur.*, vol. 7, no. 1, pp. tyab005, 2021. doi: 10.1093/cybsec/tyab005.

[8] D. Dasgupta, Z. Akhtar, and S. Sen, "Machine learning in cybersecurity: A comprehensive survey," *The J. Def. Model. Simul.*, vol. 19, no. 1, pp. 57–106, 2022. doi: 10.1177/1548512920951275.

[9] K. Shaukat *et al.*, "Performance comparison and current challenges of using machine learning techniques in cybersecurity," *Energies*, vol. 13, no. 10, pp. 2509, 2020. doi: 10.3390/en13102509.

[10] T. Berghout, M. Benbouzid, and S. M. Muyeen, "Machine learning for cybersecurity in smart grids: A comprehensive review-based study on methods, solutions, and prospects," *Int. J. Crit. Infrastruct. Prot.*, vol. 38, no. 19, pp. 100547, 2022. doi: 10.1016/j.ijcip.2022.100547.

[11] I. D. Aiyanyo, H. Samuel, and H. Lim, "A systematic review of defensive and offensive cybersecurity with machine learning," *Appl. Sci.*, vol. 10, no. 17, pp. 5811, 2020. doi: 10.3390/app10175811.

[12] M. Abutaha, M. Ababneh, K. Mahmoud, and S. A. H. Baddar, "URL phishing detection using machine learning techniques based on URLs lexical analysis," in *2021 12th Int. Conf. Inf. Commun. Syst. (ICICS)*, Valencia, Spain, IEEE, May 2021, pp. 147–152.

[13] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proc. Anti-Phish. Work. Groups 2nd Annual eCrime Res. Summit*, Oct. 2007, pp. 60–69.

[14] D. Samad and G. A. Gani, "Analyzing and predicting spear-phishing using machine learning methods," *Multidiszciplináris Tudományok*, vol. 10, no. 4, pp. 262–273, 2020. doi: 10.35925/j.multi.2020.4.30.

[15] N. Yadav and S. P. Panda, "Feature selection for email phishing detection using machine learning," in *Int. Conf. Innov. Comput. Commun.: Proc. ICICC 2021*, Springer Singapore, 2022, vol. 2, pp. 365–378. doi: 10.1007/978-981-16-2597-8.

[16] M. Medelbekov, M. Nurtas, and A. Altaibek, "Machine learning methods for phishing attacks," *J. Problems Comput. Sci. Inf. Technol.*, vol. 1, no. 2, 2023. doi: 10.26577/JPCSIT.2023.v1.i2.02.

[17] V. Shahrivari, M. M. Darabi, and M. Izadi, "Phishing detection using machine learning techniques," arXiv preprint arXiv:2009.11116, 2020.

[18] R. Alazaidah *et al.*, "Website phishing detection using machine learning techniques," *J. Stat. Appl. Probab.*, vol. 13, no. 1, pp. 119–129, 2024. doi: 10.18576/jsap/130108.

[19] S. Sindhu, S. P. Patil, A. Sreevalsan, F. Rahman, and M. S. AN, "Phishing detection using random forest, SVM and neural network with backpropagation," in *Int. Conf. Smart Technol. Comput., Electr. Electron. (ICSTCEE)*, IEEE, 2020, pp. 391–394.

[20] A. Pandey, N. Gill, K. Sai Prasad Nadendla, and I. S. Thaseen, "Identification of phishing attack in websites using random forest-SVM hybrid model," in *Intell. Syst. Des. Appl.: 18th Int. Conf. Intell. Syst. Des. Appl. (ISDA 2018)*, Vellore, India, Springer International Publishing, 2020, vol. 941, pp. 120–128.

[21] E. Zhu, Y. Ju, Z. Chen, F. Liu, and X. Fang, "DTOF-ANN: An artificial neural network phishing detection model based on decision tree and optimal features," *Appl. Soft Comput.*, vol. 95, no. 13, pp. 106505, 2020. doi: 10.1016/j.asoc.2020.106505.
[22] O. Kayode-Ajala, "Applying machine learning algorithms for detecting phishing websites: Applications of SVM, KNN, decision trees, and random forests," *Int. J. Inf. Cybersecur.*, vol. 6, no. 1, pp. 43–61, 2022.
[23] S. Al-Ahmadi, "PDMLP: Phishing detection using multilayer perceptron," *Int. J. Netw. Secur. Appl. (IJNSA)*, vol. 12, pp. 59–72, 2020.
[24] A. Odeh, I. Keshta, and E. Abdelfattah, "Efficient prediction of phishing websites using multilayer perceptron (MLP)," *J. Theor. Appl. Inf. Technol.*, vol. 98, no. 16, pp. 3353–3363, 2020.
[25] N. N. Naik, "Modelling enhanced phishing detection using XGBoost," Doctoral dissertation, National College of Ireland, Dublin, 2021.
[26] K. Joshi *et al.*, "Machine-learning techniques for predicting phishing attacks in blockchain networks: A comparative study," *Algorithms*, vol. 16, no. 8, pp. 366, 2023. doi: 10.3390/a16080366.
[27] B. Sharma and P. Singh, "An improved anti-phishing model utilizing TF-IDF and AdaBoost," *Concurr. Comput.*, vol. 34, no. 26, pp. e7287, 2022.
[28] F. Nthurima, A. Mutua, and W. S. Titus, "Detecting phishing emails using random forest and AdaBoost classifier model," 2023. doi: 10.32591/coas.ojit.0602.03123n.
[29] K. Omari, "Phishing detection using gradient boosting classifier," *Proc. Comput. Sci.*, vol. 230, no. 5, pp. 120–127, 2023. doi: 10.1016/j.procs.2023.12.067.
[30] P. Flach, "Performance evaluation in machine learning: The good, the bad, the ugly, and the way forward," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 9808–9814, Jul. 2019. doi: 10.1609/aaai.v33i01.33019808.
[31] H. Shaheen, S. Agarwal, and P. Ranjan, "MinMaxScaler binary PSO for feature selection," in *First Int. Conf. Sustainable Technol. Comput. Intell.: Proc. ICTSCI 2019*, Springer Singapore, 2020, pp. 705–716.
[32] A. Dogan and D. Birant, "A weighted majority voting ensemble approach for classification," in *2019 4th Int. Conf. Comput. Sci. Eng. (UBMK)*, IEEE, Sep. 2019, pp. 1–6.
[33] T. N. Rincy and R. Gupta, "Ensemble learning techniques and its efficiency in machine learning: A survey," in *2nd Int. Conf. Data, Eng. Appli. (IDEA)*, IEEE, Feb. 2020, pp. 1–6.