Bulletin of Electrical Engineering and Informatics
Vol. 14, No. 6, December 2025, pp. 4626~4640
ISSN: 2302-9285, DOI: 10.11591/eei.v14i6.9579

A powerful machine learning method for detecting phishing threats

Mahmoud Baklizi[1], Jamal Zraqou[1], Mohammad Alkhazaleh[2], Issa Atoum[3], Faisal Alzyoud[2], Musab B. Alzghoul[2]
[1]Department of Computer Science, Faculty of Information Technology, University of Petra, Amman, Jordan
[2]Department of Computer Science, Faculty of Information Technology, Isra University, Amman, Jordan
[3]Department of Software Engineering, Faculty of Information Technology, Philadelphia University, Amman, Jordan

Article Info

Article history:

Received Nov 18, 2024
Revised Sep 13, 2025
Accepted Sep 27, 2025

Keywords:

Cybersecurity
Intrusion detection
Machine learning
Phishing detection
Threat detection

ABSTRACT

Phishing threats exploit social engineering and deceptive web infrastructure to steal sensitive personal information, often by mimicking legitimate websites. With the proliferation of online services and the increasing prevalence of cybercrime, detecting phishing websites has become a critical challenge. This study presents a comprehensive machine learning (ML)-based approach for detecting phishing websites. A total of 48 discriminative features were extracted from 10,000 web pages—comprising 5,000 phishing and 5,000 legitimate sites. Nine ML classifiers were initially evaluated, including random forest (RF), support vector machine (SVM), and XGBoost. Ensemble models based on soft voting and stacking were then constructed to improve detection performance. Among the models, the soft voting classifier (VC) achieved the best performance with an accuracy and F1-score of 98.82%. The results indicate that ensemble learning offers a robust solution for the automated detection of phishing websites.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Jamal Zraqou
Department of Computer Science, Faculty of Information Technology, University of Petra
Queen Alia Airport Street, Amman, Jordan
Email: Jamal.Zraqou@uop.edu.jo

1. INTRODUCTION

The swift evolution of the internet, coupled with its expanding usage, complicates and furthers the security issue [1], [2]. Recently, numerous threats have emerged that aim to obtain sensitive personal information for financial gain or identity theft. One of these common threats is phishing.

Phishing is a criminal activity that uses technology and social engineering to steal sensitive personal information from victims [3]. This information can include financial account details, login credentials, usernames, passwords, personal contacts, and social relationships [4]. In phishing, attackers contact users by phone, text, or email to solicit personal information while posing as well-known or respected businesses [4], [5]. Phishing websites often pretend to have urgent issues, such as unpaid invoices, suspicious account activity, or requests to log in to "verify" your password or account details. These phony websites can also request confidential information, including bank account numbers or credit card details. If you enter this information, cybercriminals can gain access to your accounts, steal your data, commit identity theft, and even infect your device with malware [6]-[8].

According to Awasthi and Goel [9], about half of cybersecurity experts noticed these. Due to the COVID-19 pandemic, they are subjected to attacks. In addition, there are around 1185 phishing attacks directed at enterprises every month, too. In turn, security professionals might have to use 1–4 days to fend off a cyber

attack. Furthermore, about 30% of people in cybersecurity reported that phishing has a high rate of success now. Given the increase in phishing attacks, various methods to handle and mitigate them have been proposed.

To address this persistent challenge, the present study initiates the development and testing of an effective machine learning (ML)-based system for identifying phishing sites through systematic page features. Whereas earlier works were mainly centered on email content or user actions, our approach focuses on 48 manual features taken from both real and trick webpages. The main goals of this study are: i) to compare classic and ensemble ML classifiers for phishing detection, ii) to pick the strongest and most accurate model for real-time application, and iii) to prove the model's usability through analysis of inference time and model size. With understandable features and flexible algorithms, this work offers a simple yet very accurate detection model that can work in resource-limited settings. Furthermore, we demonstrate how ensemble techniques, such as soft voting, enable high performance levels comparable to more complex architectures, while maintaining efficiency and clarity.

Our research efforts have led to the examination of ML methods for detecting phishing websites, as artificial intelligence plays a significant role in smart city contexts [10]. When we reviewed the study challenge's goals and looked at the training data, we determined that support vector machine (SVM), random forest (RF), artificial neural network (ANN), k-nearest neighbors (KNN), logistic regression (LR), gradient boosting, XGBoost, CatBoost, LightGBM, and two ensemble meta-learners: voting and stacking classifiers were the best models to use. After that, we use each of the ML classifiers with the dataset to discover which one has the best outcome.

You'll find the sections are grouped as shown below. In section 2, the article describes the literature review. Section 3 explains the way the proposed approach is implemented. In section 4, the researchers show and evaluate the different results. Finally, the author sums up the report in the last section.

2. BACKGROUNDS AND LITERATURE REVIEW

As phishing websites pose a significant threat, numerous studies and reviews have been published. These recent works on phishing website detection provide crucial knowledge for researchers to understand various methods to detect phishing. Generally, phishing can be detected based on ML detection, blacklist detection, heuristic detection, and visual similarity detection. In the next section, the study will explain how phishing can be detected using ML.

2.1. Machine learning algorithms

Phishing websites are much easier to find with the help of ML [11]. ML a subset of AI, helps build models by learning from given data and using them for predictions [12]. Examples of ML classifiers include SVM, RF, and ANN.

2.1.1. Random forest

RF is an interesting ML model since it classifies data accurately and efficiently. RF is made up of supervised learning algorithms that are used for performing both classification and regression [13]. To find the best conclusion, it gathers and assesses the results from some decision trees (DT), as shown in Figure 1.

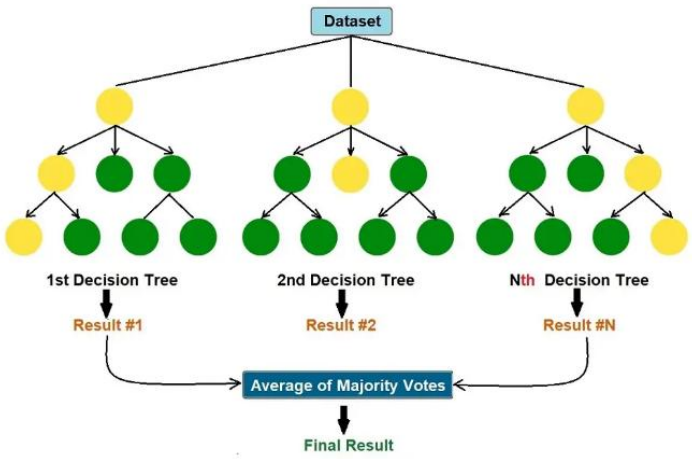

Figure 1. RF architecture

2.1.2. Support vector machines

The reason SVM is commonly used in text classification is that it operates at a high speed and is highly effective [13]. It creates a hyperplane or collection of hyperplanes that can be used to divide data into several classifications [14], as shown in Figure 2. SVM has been shown to perform better than several ML techniques [15].

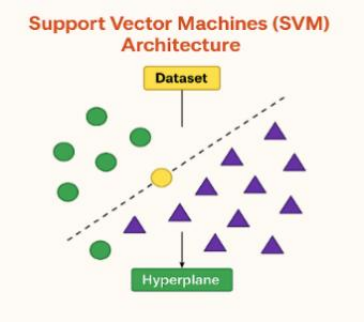

Figure 2. SVM architecture

2.1.3. Artificial neural networks

A supervised ML approach called ANN is utilized for regression and classification prediction. As shown in Figure 3, the ANN model consists of layers, and each layer of the model is specified by several nodes (neurons) [12]. Nonlinear statistical models reveal a complex relationship between inputs and outputs, enabling the discovery of novel patterns [13]. The ANN can learn from the data and produce replies in the form of predictions or classifications, just like the neurons in the human nervous system can do so [12].

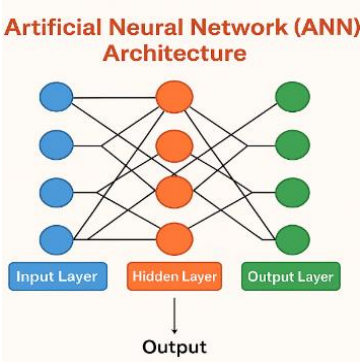

Figure 3. ANN architecture

2.2. Related works

Omari conducted a comparative study to identify phishing websites using ML algorithms, testing various models LR, SVM, KNN, Naïve Bayes, DT, RF, and gradient boosting) on the UCI phishing dataset [16]. The study highlights gradient boosting and RF as top performers, achieving accuracies of 97.2% and 97.1%, respectively. The paper underscores the importance of robust ML approaches to combat evolving phishing threats. Jain and Gupta [17] proposed a ML approach to detect phishing attacks based on URLs. The researchers used 14 characteristics of URLs to tell whether a website could be trusted or not. Two kinds of datasets were used in my research, one with 32,951 phishing URLs and another with valid URLs that came from various sources. These ML techniques were used as classifiers. They stated that SVM performed better than Naïve Bayes in terms of accuracy.

The work in [18] presented the role of feature selection in ML to detect spam and phishing attacks. Feature selection optimization was used to test the best classifiers among RF, KNN, ANN, SVM, LR, and Naïve Bayes. The authors utilized spam emails and UCI datasets (including 11,056 websites with 31 features) in the proposed work and employed the Weka application for feature selection optimization.

BestFirst+CfsSubsEval was used as the first feature selection, which reduces the features to 10. Ranker+Principal was the second feature selection optimization, reducing the features to 30. The results reveal that RF performed the best, achieving an accuracy of 94.77% with BestFirst+CfsSubsEvaluation and 97.33% with Ranker+Principal component. Furthermore, the performance of Ranker+Principal was superior to that of BestFirst+CfsSubsEval.

Lokesh and Gowda [19] proposed phishing website detection based on a practical ML approach. The best ML techniques for phishing detection were applied to show the pros and cons of each method. An automatic ranking system for feature detection was introduced to detect phishing with a certain level of accuracy. PhishTank and MillerSmiles are the resources of the used dataset, comprising 30 features. RF, KNN, DT, SVM, and support vector classifier (SVC) were utilized for testing. The result revealed that the RF technique has outperformed the former in terms of accuracy, scoring 96.87%. In [15], the former work was criticized for the data sources and feature extraction being insufficient.

Ramaiah *et al*. [20] proposed an ensemble stacking model for phishing website detection using DT-recursive feature elimination with cross-validation (DT-RFECV) to select optimal features, including URL syntax, SSL attributes, and DOM-based characteristics. Their framework achieved 97.7% accuracy on a balanced dataset of 10,000 samples (50% phishing/legitimate) from Mendeley repositories, leveraging only 10 features. Similarly, Nova *et al*. [21] proposed an ensemble learning approach combining RF, XGBoost, and CatBoost with RFECV-based feature selection to detect phishing websites. It extracts 35 features, including NLP-based, address-based, and HTML-based attributes, achieving 88.90% accuracy.

Chawla [12] suggested analyzing specific characteristics of phishing sites to identify and capture them. Moreover, the features of the UCI dataset and the ML methods were explained and illustrated in detail. Following this, the UCI dataset is used to test the following ML classifiers (LR, KNN, SVM, DT, RF, ANN, and max voting classifier (MVC)). The results show that MVC gains the best accuracy among all classifiers (97.73%). As mentioned, the UCI dataset comprises 11,056 websites, featuring 30 attributes. Of these, 6,157 are classified as non-phishing, and 4,898 are identified as phishing. Samad *et al*. [22] criticized this and said there is no balance between phishing and non-phishing websites. They also worked on testing them in a balanced way (6,157 phishing and 6,157 non-phishing websites) and explained that there are differences between them.

Alnemari and Alshammari [13] developed and evaluated four models (ANN, RF, DT, and SVM) to increase the effectiveness of ML for phishing domain detection. They also used the MinMax normalization technique to develop the four models. The MinMax normalization technique improved the four models by compressing the data to a domain of (1, 0). Moreover, the authors used the UCI dataset, which was criticized by Samad *et al*. [22]. The test results revealed that the RF technique performed best, achieving an accuracy of 97.3%.

After explaining and clarifying the previous research, the following observations were noted: most of the proposed works used the UCI data set, which we mentioned earlier is an unbalanced dataset. It contains a small number of features (30).

Lately, scientists in the phishing detection sector have looked into how various types of ML models, including advanced and ensemble methods, work to detect phishing threats. DT, Naïve Bayes, and LR are often picked because they are easy to work with and their results can be explained. Still, their efficiency drops when the data is complex or has many dimensions, according to what [23], [24] found in their review. To achieve better results and make the model more reliable, RF, AdaBoost, and gradient-based techniques have been mainly used as ensemble methods. In many cases, these strategies outperform single learners in detecting phishing scams, as demonstrated by [25]-[27].

Simultaneously, deep learning has also been experiencing a wave in its acceptance, especially via convolutional neural networks (CNNs), long short-term memory (LSTMs) networks, and CNN-based, LSTM networks. These models have shown superior performance in capturing sequential patterns and extracting contextual features from URLs or webpage contents [28]. Nevertheless, the computational cost, complexity, and data requirements of deep learning limit their applicability in real-time or lightweight environments. Hybrid frameworks that combine ML with feature selection, natural language processing (NLP), or URL parsing have also proven effective. These systems integrate multiple feature domains, improving phishing resilience while maintaining scalability [29]-[31].

In contrast to these complex approaches, ANN, SVM, and RF were chosen in this research to ensure there would be no serious issues when these models are put into practical use. These models are well-suited for real-time detection systems where interpretability, speed, and low-resource requirements are critical. Table 1 shows the summary of the proposed methods.

Table 1. The summary of proposed methods

Authors	Dataset	Technique	Objective
Alnemari and Alshammari [13]	UCI dataset 11,055 with 30 features	ANN, SVM, DT, and RF	To develop and evaluate four models to increase the effectiveness of ML for phishing domain detection
Chawla [12]	UCI dataset 11,056 websites with 30 features	LR, KNN, SVM, DT, RF, ANN, and MVC	Analyzing some characteristics of phishing sites to capture these sites using these characteristics.
Ramaiah *et al*. [20]	Two datasets (DS-1, DS2) from Mendeley, each with distinct characteristics.	DT, RF, bagging classifier (BC), and proposed ensemble stacking model	Develop an accurate phishing detection model using ensemble learning, DT-RFECV feature selection, and SMOTE to combat evolving cyber threats.
Lokesh and Gowda [19]	Datasets from the Phish Tank and Miller Smiles archives, with 30 features.	RF, KNN, DT, SVM, and SVC	To identify the best ML techniques for phishing detection and the pros and cons of each technique.
Salihovic *et al*. [18]	UCI Dataset, 11,056 websites with 31 features, using feature selection optimization, BestFirst + CfsSubsEvaluation feature 10, Ranker + Principal Component feature 30.	RF, KNN, ANN, SVM, LR, and Naïve Bayes	Test the most successful algorithms using feature selection optimization.
Jain and Gupta [17]	PhishTank has 14 features. The first dataset consists of 15000 URLs (1000 non-phishing URLs and 14000 phishing URLs). The second dataset contains 25000 URLs (2000 non-phishing URLs and 23000 phishing URLs).	SVM and Naïve Bayes	Anti-phishing URLs using ML method
Omari [16]	UCI Dataset 11,055 with 30 features.	SVM, Naïve Bayes, RF, LR, KNN, DT, and Gradient Boosting	To assess the efficiency of seven ML models in detecting phishing domains.
Almujahid *et al*. [27]	The data consisted of 48 features extracted from a collection of 10,000 web pages. 5,000 out of the 10,000 suspected sites were labeled as phishing, while the remainder were confirmed as trustworthy websites.	LR, KNN, DT, RF, SVM, XGBoost, and CNNmodel	Conduct a study to evaluate the efficiency of the eight ML and DL algorithms in detecting phishing.
Alshingiti *et al*. [28]	The dataset, comprising 20,000 records with 80 features, was very detailed; therefore, the 30 "best" features were identified and selected using the SelectKBest method.	LSTM, CNN, and n LSTM–CNN-based approach	develop something that can determine if a website is phishing.
This study – stacking	The names of four datasets are like this: D1, D2, D3, and D4. D1 is taken up in the UCI repository, and D2 consists of 48 features. Also, the two datasets D3 and D4 are taken; D3 consists of 111 features with a total of 58,645 observations, and D4 consists of 111 features with 88,647 records. There are two classes in every dataset: phishing and legitimate.	MLSELM	An ensemble model made up of multiple layers to find phishing sites.
This study – voting soft, 2023	Three intermediate-sized websites' phishing data (1000-31000 variables in each dataset).	XGBoost, CatBoost, and LightGBM	Evaluates the promise of ML on phishing domain identification

3. PROPOSED APPROACH AND METHODOLOGY

Phishing website detection can significantly benefit from the application of ML. Figure 4 outlines a comprehensive methodology for leveraging ML to identify phishing websites. It is composed of the following phases:

- Data collection: compile a labelled dataset comprising legitimate and phishing websites. Labels indicate the authenticity of each website, distinguishing between genuine, and phishing attempts.
- Feature extraction: isolate relevant attributes of the data maintained on the websites, which may include information on send addresses, web headers, email subject, textual data, embedded links, attachments, and metadata. The attributes are the inputs of the ML model.
- Preprocessing: clean and preprocess the website data, involving tasks like removing stop words, text tokenization, case normalization, and addressing special characters or formatting issues.
- Feature engineering: convert the extracted features into a suitable format for ML algorithms. Techniques may include one-hot encoding, term frequency-inverse document frequency (TF-IDF) representation, or word embeddings to capture semantic meaning.
- Model training: the experiments were carried out with the use of RF, SVM, and ANN ML algorithms. The data was split into training and testing parts.

- Model evaluation: the testing dataset was used to determine how well the model was performing. To find out the accuracy of binary classification models, most researchers employ accuracy, precision, recall, F1-score, and receiver operating characteristic (ROC).
- Model optimization: several hyperparameters were adjusted to control how a learning algorithm learns from data. Grid search was employed to find optimal parameter settings that yield the best results.
- Performance evaluation: the model was tested on real-time phishing detections. The performance was periodically monitored using the learned datasets.

Training data is essential to ML-based phishing detection algorithms. In addition, the choice of features, the selection of appropriate algorithms, and ongoing monitoring and tuning to be up to date are crucial to improving accuracy against emerging phishing threats.

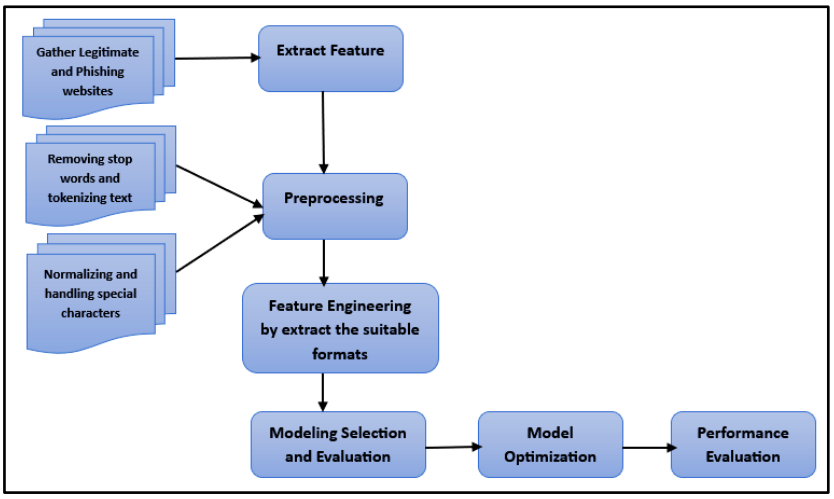

Figure 4. Methodology of the proposed approach

3.1. Model configuration and validation

Various types of ML techniques were tested in this study for detecting phishing, among them were SVM, RF, ANN, KNN, LR, gradient boosting, XGBoost, CatBoost, LightGBM, and two ensemble meta-learners: voting and stacking classifiers. We used scikit-learn version 1.6.1, in conjunction with Python 3.10, to carry out all the models.

Special hyperparameters for each model were found using grid search, and the results were checked with 10-fold stratified cross-validation so that the data was equally divided among all classes. Metrics considered in this experiment were accuracy, precision, recall, F1-score, and ROC-AUC. Table 1 summarise how each model is configured.

Ensemble models, such as stacking and voting classifiers (VC), integrate multiple base learners, with a LR meta-model used in the final estimator for stacking. To ensure comparability, all models were evaluated under the same folds and training pipeline. The use of standard scaling was applied where appropriate (e.g., SVM, ANN, KNN, and LR) using pipelines. Table 2 shows the ML model configuration.

Table 2. ML model configuration

ML model	Hyperparameter configuration
SVM	C = [1, 10]; gamma = 'scale'; kernel = 'rbf'; probability = True; random_state = 42
RF	n_estimators = 100; max_depth = [10, 20]; criterion = 'gini'; bootstrap = True; random_state = 42
ANN	hidden_layer_sizes = [(5, 2), (100,), (100, 50)]; activation = 'relu'; solver = 'adam'; alpha = 0.0001; learning_rate = 'adaptive'; max_iter = 500
KNN	n_neighbors = 5; weights = 'uniform'
LR	C = 1; solver = 'liblinear'; max_iter = 500; random_state = 42
Gradient boosting	n_estimators = 100; learning_rate = 0.1; max_depth = 3; random_state = 42
XGBoost	n_estimators = 100; learning_rate = 0.1; max_depth = [3, 5]; eval_metric = 'logloss'; random_state = 42
CatBoost	iterations = 100; depth = [4, 6]; learning_rate = 0.1; verbose = 0; random_state = 42
LightGBM	n_estimators = 100; learning_rate = 0.1; max_depth = [3, 5]; objective = 'binary'; random_state = 42; verbose = -1
VC (Soft)	Base estimators: XGBoost, CatBoost, LightGBM; voting = 'soft'; random_state = 42
Stacking cassifier	Base estimators: SVM, RF, XGBoost; Final estimator: LogisticRegression(); n_jobs = -1; random_state = 42

4. RESULTS AND DISCUSSION

4.1. dataset description and preprocessing

The data utilized in this research is from a freely available Kaggle repository named "Phishing Dataset for ML" Chiew *et al*. [29]. The data includes 10,000 records that have been labelled, balanced between 5,000 phishing and 5,000 legitimate web entries. Each instance consists of 48 features derived from URL characteristics, HTML content patterns, and client-side behavioural signals.

The features include syntactic indicators such as the number of dots, dashes, and special characters (e.g., '@', '%', '&'), as well as structural and contextual markers like URL length, path complexity, and domain irregularities. It also captures web behaviour signals such as the use of insecure forms, JavaScript anomalies, pop-up triggers, and iframe usage. These features have been widely adopted in phishing research due to their effectiveness in distinguishing malicious behaviour, as shown in Figure 5.

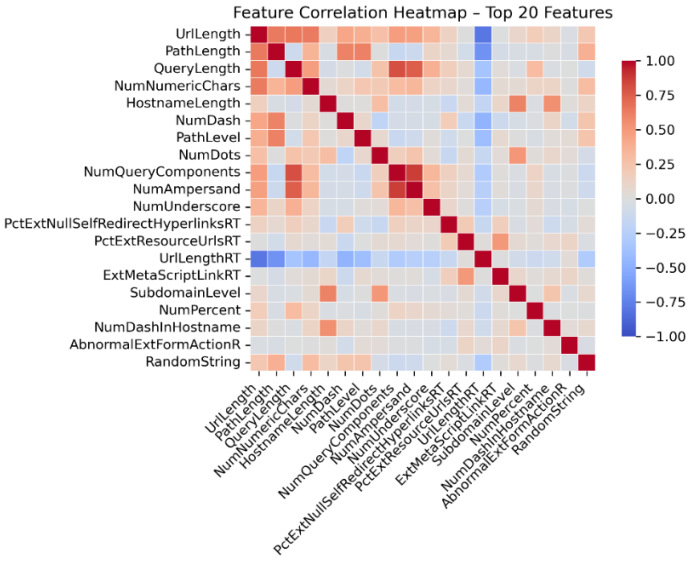

Figure 5. Heatmap correlation between features

Before training, the dataset was preprocessed by removing duplicates and verifying label consistency. Binary labels were encoded as 1 (phishing) and 0 (legitimate). All numerical features were scaled using Min-Max normalisation to the [0, 1] range to ensure consistency in learning. No categorical variables required encoding, and no rows contained null values. All 48 features were retained without dimensionality reduction to preserve the interpretive value of each attribute, given their domain relevance and collective contribution to phishing detection. Table 3 shows the dataset specifications.

Table 3. The dataset specifications

Number of features	Number of phishing web pages	Number of phishing web pages
48	5000	5000

4.2. Evaluation criteria

The evaluation criteria to evaluate the ML models on the tested dataset are introduced. This framework gauges the significance of the ML models in terms of threat detection. Accuracy, precision, recall, and F1-score are exposed as crucial benchmarks for assessing the effectiveness of the ML models [32]. By meticulously analyzing these metrics, the study ensures a comprehensive understanding of the models' proficiency in accurately discerning and categorizing threats.

– Accuracy is used to correctly measure the classified instances in a dataset, providing an overall view of the model's correctness as shown in (1):

$$Accuracy = (True\ Positives\ +\ True\ Negatives)/Total\ Predictions \quad (1)$$

where: true positives is the successful identification of phishing websites and true negatives is the successful identification of legitimate websites.

– Precision is the extent to which true positive (TP) predictions are measured against all predictions.

$$Precision = (True\ Positive)/(True\ Positives\ +\ False\ Positives) \quad (2)$$

– Recall: hitting on all positive cases.

$$Recall = (True\ Positive)/(True\ Positives\ +\ False\ Negatives) \quad (3)$$

– F1-score: relates precision and recall, delivering a stable performance measure.

$$F1 - score = (2\ \times\ Precision\ \times\ Recall)/(\ Precision\ +\ Recall) \quad (4)$$

As indicated in [33], the robust evaluation of the tested ML models can be assessed using these metrics in threat detection.

4.3. Results of selected models

All the classifiers showed excellent performance, and the accuracy percentages ranged from 0.9443 to 0.9882 while using ML algorithms. Between the VCs, the soft VC did the best on every measurement, improving accuracy (0.9882), precision (0.987425), recall (0.989), F1-score (0.988207), and ROC-AUC (0.9882). Performances were mainly top-notch because of ensemble approaches such as XGBoost (0.9878), stacking classifier (0.9875), CatBoost (0.9874), and LightGBM (0.987). While they did not match the high scores of the deep learning methods, the traditional ML algorithms still worked steadily. Specifically, the low score was achieved by LR (0.9443), and KNN had moderate results (0.9519). Since the difference in accuracy between the top and bottom classifiers is small, it reveals that the features of the dataset fit with various algorithms. Nevertheless, ensemble methods usually result in small but functional gains.

Different algorithm families display appropriate hyperparameter configurations that match the established guidelines of ML. Most of the time, tree-based ensemble methods choose simple, moderate-depth options and varied learning rates, opting for a balanced level of complexity and ability to generalize. The used neural network was the multilayer perceptron, with ReLU for activation, adaptive learning rate, and Adam optimizer—a regular configuration for working with tables. Traditional algorithms required minimal tuning, as SVM maintained C=10 and RBF scale gamma, while KNN used k=5 neighbors with uniform weights. Both the VC and stacking classifier performed optimally without further tuning, as no additional hyperparameter adjustments were required for them or their constituent models, thereby simplifying model selection while maintaining the best results. Tables 4 and 5 show the results.

Table 4. Results for various classifiers

	Accuracy	Precision	Recall	f1	AUC
Stacking classifier	0.9875	0.9862	0.9888	0.9875	0.9875
Voting cassifier (soft)	0.9882	0.9874	0.9890	0.9882	0.9882
ANN (MLP)	0.9767	0.9727	0.9810	0.9768	0.9767
RF	0.9841	0.9834	0.9848	0.9841	0.9841
SVM (RBF)	0.9659	0.9642	0.9678	0.9660	0.9659
KNN	0.9519	0.9479	0.9564	0.9521	0.9519
LR	0.9443	0.9375	0.9522	0.9447	0.9443
Gradient boosting	0.9776	0.9755	0.9798	0.9776	0.9776
XGBoost	0.9878	0.9866	0.9890	0.9878	0.9878
CatBoost	0.9874	0.9855	0.9894	0.9874	0.9874
LightGBM	0.9870	0.9858	0.9882	0.9870	0.9870

Table 5. The classifier results

Stacking classifier	best_params
VC (soft)	{}
ANN (MLP)	{}
RF	{'clf__activation': 'relu', 'clf__alpha': 0.0001, 'clf__hidden_layer_sizes': (100,), 'clf__learning_rate': 'adaptive', 'clf__solver': 'adam'}
SVM (RBF)	{'clf__criterion': 'gini', 'clf__max_depth': 20, 'clf__n_estimators': 100}
KNN	{'clf__C': 10, 'clf__gamma': 'scale'}
LR	{'clf__n_neighbors': 5, 'clf__weights': 'uniform'}
Gradient boosting	{'clf__C': 1, 'clf__solver': 'liblinear'}
XGBoost	{'clf__learning_rate': 0.1, 'clf__max_depth': 3, 'clf__n_estimators': 100}
CatBoost	{'clf__learning_rate': 0.1, 'clf__max_depth': 5, 'clf__n_estimators': 100}
LightGBM	{'clf__depth': 6, 'clf__iterations': 100, 'clf__learning_rate': 0.1}
Stacking classifier	{'clf__learning_rate': 0.1, 'clf__max_depth': 5, 'clf__n_estimators': 100}

The VC operates well across all classes, but especially ensures that websites, whether phishing (class 1) or legitimate (class 0), get the same level of accuracy during detection. The precision of class 0 was 0.9900, the recall was 0.9853, and the F1-score was 0.9876; meanwhile, class 1 had a precision of 0.9854, a recall of 0.9900, and an F1-score of 0.9877. The importance of avoiding class bias is underscored in cybersecurity, where both types of errors can be hazardous. Table 6 illustrates the voting ensemble classifier.

Table 6. Phishing detection per class for the voting ensemble classifier

	Precision	Recall	F1-score	Support
0	0.9900	0.9853	0.9876	1500
1	0.9854	0.9900	0.9877	1500

Statistically, these results confirm that while all models performed competitively, the VC achieved the most consistent and statistically reliable accuracy. Using the friedman test ($\chi^2(4)=14.2041$, $p=0.0067$) followed by the Nemenyi post hoc test, we found that voting significantly outperformed RF ($p=0.0227$) and XGBoost ($p=0.0409$). No significant difference was observed between voting and stacking ($p=0.9751$) or LightGBM ($p=0.4973$). These statistical findings, combined with the model's ensemble architecture, reinforce its suitability for high-stakes phishing detection applications where both reliability and interpretability are crucial.

4.4. Comparative analysis of existing work

Our method not only achieves state-of-the-art accuracy, but does so using interpretable, lightweight models. Furthermore, the inclusion of 48 well-engineered features and the use of 10-fold cross-validation ensure both generalizability and deployment readiness. The proposed phishing detection method using a voting ensemble outperformed all the compared models in Table 7. However, the deep learning model of Alshingiti *et al*. [28] outperformed the voting ensemble, possibly due to its consideration of different datasets that may contain features not extracted by the current model. For instance, Karim *et al*. [30] achieved 98.12% accuracy using a hybrid ensemble model, while Almujahid *et al*. [27] reported 98.00% accuracy for both RF and XGBoost. Kalabarige *et al*. [31] presented an ensemble stacked model with 98.43% accuracy but lacked SHAP-based interpretability. Alshingiti *et al*. [28] reported 95.3% accuracy using a deep learning CNN-LSTM ensemble, though at a significantly higher computational cost.

Table 7. The accuracy and F1-scores for the existing module

Study/model	Model type	Accuracy (%)	F1-score
This study–RF	Classical ensemble	98.40	0.9840
This study–XGBoost	Gradient boosting	98.78	0.9878
Kalabarige *et al.* 2022 [31]	Stacked ensemble	98.43	0.9844
Karim *et al.* 2023 [30]	Hybrid ML with URL features	98.12	0.9589
Almujahid *et al*. 2024 [27]	RF and XGBoost	98.00	0.9800
Alshingiti *et al*. 2023 [28]	Deep learning (CNN+LSTM)	99.20	99.200
This study–stacking	Ensemble (SVM+RF+XGB)	98.73	0.9874
This study–voting (soft)	Ensemble (XGB+CatBoost+LGBM)	98.82	0.9882

4.4.1. Feature importance analysis

We checked why the RF model performed better by testing the importance of each feature using both the permutation method and the shapley additive explanations (SHAP) tool. They enable the assessment of the impact of each feature on the prediction. Permutation importance increases the prediction error if you randomly mix the values of a specific feature. Still, SHAP values show how each variable influences the outcome of each prediction with the help of game theory.

Figure 6 shows the top 15 most significant characteristics based on mean SHAP values. Notably, features related to URL structure and suspicious content indicators—such as RandomString, NumSensitiveWords, PctExtHyperlinks, EmbeddedBrandName, and IpAddress—emerged as dominant predictors of phishing behavior. These results are consistent with findings in recent literature [29], [30], which confirm that lexical, contextual, and structural attributes of web pages are strong phishing signals.

The RF model's strength lies in its ensemble of DT that can naturally model non-linear relationships and interactions among features. In contrast, SVM and ANN showed comparatively lower sensitivity to nuanced patterns, potentially due to parameter limitations and sensitivity to feature scaling.

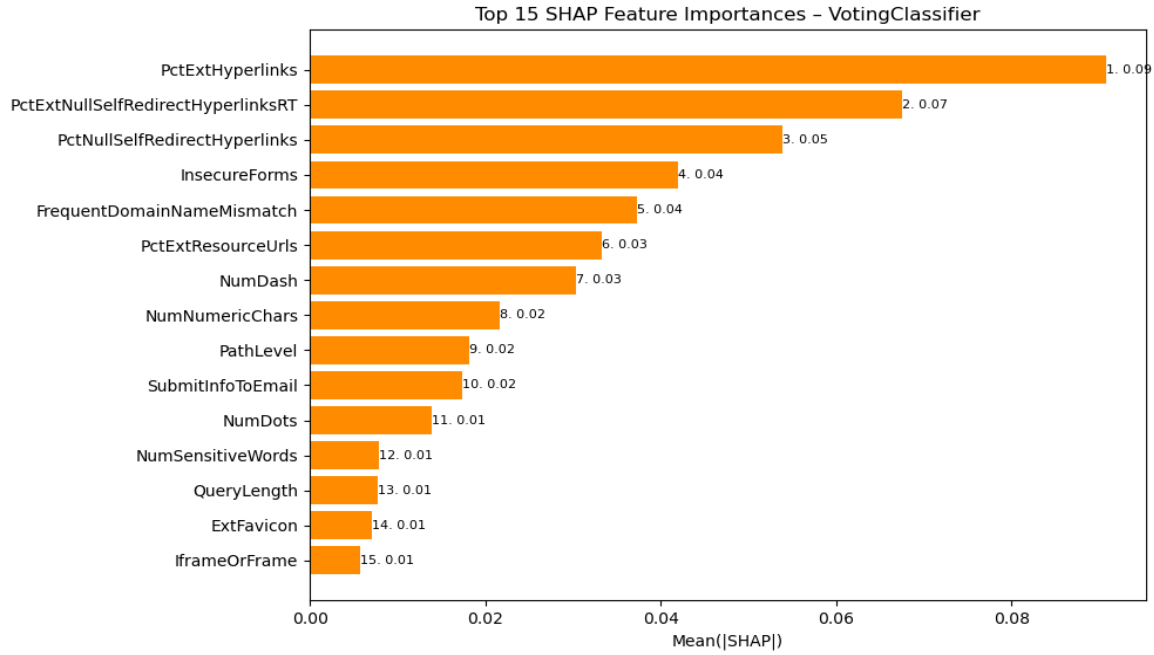

Figure 6. Top 15 features using SHAP importance for voting

4.4.2. Feature attribution and shapley additive explanations-based analysis

To explain the predictive behaviour of the models, SHAP was used to rank the top 15 most influential features for both the RF and VC models. Figure 7 presents the SHAP summary plots, and Table 8 lists the top-ranked features with interpretive descriptions.

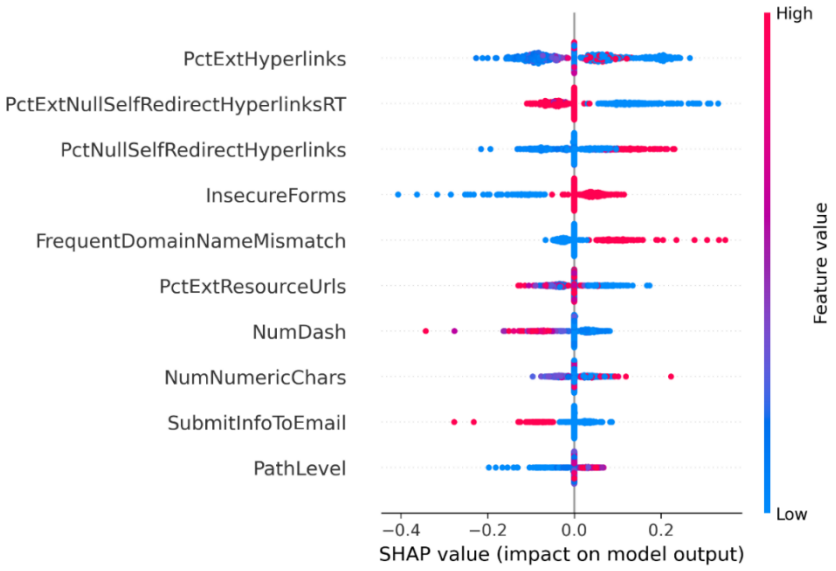

Figure 7. Voting SHAP summary

For the RF model, the top three contributors were:

- PctExtHyperlinks (*percentage of external hyperlinks in the webpage HTML source code*),
- PctExtNullSelfRedirectHyperlinksRT (*rate of hyperlinks that use "#", self-redirect, or abnormal values*),
- FrequentDomainNameMismatch (*whether the most frequent domain in the HTML does not match the page URL*).

Table 8. Features based on mean SHAP values

Feature name	Description
PctExtHyperlinks	Percentage of external hyperlinks in the webpage HTML source code.
PctExtNullSelfRedirectHyperlinksRT	Rate of links with empty, "#", self-redirect, or abnormal values in thresholded format.
FrequentDomainNameMismatch	If the domain that often appears in the HTML is not the same as the website's domain.
PctExtResourceUrls	Percentage of external resource URLs in HTML (e.g., scripts, CSS).
NumDash	Number of "-" symbols in the URL.
PctNullSelfRedirectHyperlinks	Share of hyperlinks using self-reference, "file://", or "#".
InsecureForms	Whether the form action attribute uses an insecure protocol (e.g., HTTP).
RelativeFormAction	Whether the form action uses a relative URL instead of an absolute one.
DomainInSubdomains	Whether the domain includes top-level domain patterns in its subdomain section.
RandomString	Presence of randomly generated-looking strings in the URL.
IframeOrFrame	Whether the HTML source includes iframe or frame tags.
FakeLinkInStatusBar	Use of JavaScript to modify the status bar link preview.
PopUpWindow	Use of JavaScript to open pop-up windows.
RightClickDisabled	JavaScript disabling the right-click menu.
EmbeddedBrandName	Presence of a brand name embedded in the URL structure.

Similarly, the VC, which combines XGBoost, CatBoost, and LightGBM, highlighted overlapping signals but also placed higher importance on:

- InsecureForms (*forms submitted over HTTP or without a secure protocol*),
- PctNullSelfRedirectHyperlinks (*proportion of empty/self-pointing hyperlinks*),
- RelativeFormAction (*presence of relative rather than absolute URLs in form actions*).

These findings validate the model's reliance on known phishing indicators—particularly structural deception, insecure interaction points, and abnormal link behaviour. The interpretability gains from SHAP reinforce the RF strength in handling high-dimensional, rule-based feature interactions in phishing detection tasks.

We conducted an additional evaluation by applying principal component analysis (PCA) and re-training the best-performing VC on datasets reduced to 5, 10, 15, and 20 principal components. The results demonstrated a consistent decline in classification accuracy compared to the full-feature model (original accuracy: 0.9882), with PCA-based accuracies ranging from 0.8429 (5 components) to 0.9510 (20 components). While PCA successfully preserved variance (with six components capturing over 95%), it failed to retain the discriminative structure necessary for optimal classification. Consequently, we conclude that PCA-based dimensionality reduction is not appropriate in this context, and we keep all 48 original features to preserve both performance and interpretability.

4.5. Confusion matrix and misclassification analysis

Figures 8 and 9 show the confusion matrices for the RF and VC (soft) models, respectively. Both models achieved high classification performance; however, differences in their error distribution provide insight into their behaviour.

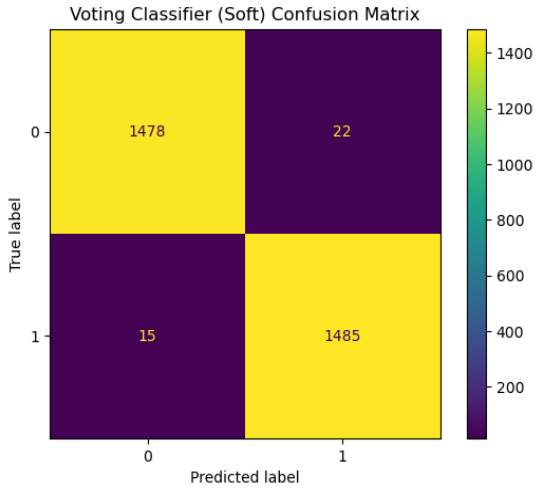

Figure 8. Confusion matrix for VC

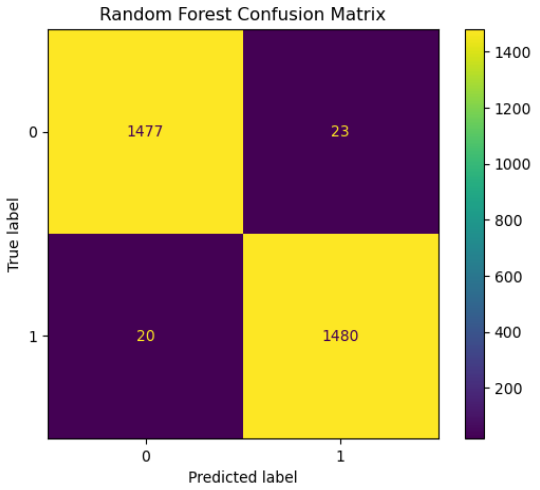

Figure 9. Confusion matrix for RF classifier

The RF model misclassified 23 legitimate websites as phishing (false positives) and failed to detect 20 phishing websites (false negatives). In contrast, the VC slightly improved these results, reducing false positives to 22 and false negatives to 15. Table 9 shows the RF and VC positive and negative results.

Table 9. RF and VC results

Model	True positives	True negatives	False positives	False negatives
RF	1480	1477	23	20
VC	1485	1478	22	15

Upon examining SHAP-based feature rankings, false positives were often associated with high values of PctExtHyperlinks, PctExtResourceUrls, and InsecureForms—features that are typically strong phishing indicators but occasionally present in legitimate sites with external content or embedded resources (e.g., advertising links, and analytics scripts).

Conversely, false negatives (missed phishing instances) were typically characterised by relatively benign patterns—URLs lacking suspicious characters (NumDash and RandomString) or showing fewer indicators in fields like RightClickDisabled or IframeOrFrame. These phishing sites may use obfuscation or mimic legitimate designs to evade detection, underscoring the evolving sophistication of phishing tactics.

Overall, the confusion matrix confirms that ensemble models like the VC reduce both types of errors. When paired with interpretable SHAP-based explanations, this provides a practical and reliable basis for real-world phishing mitigation systems.

This study demonstrates the efficacy of classical and ensemble ML models for phishing detection, achieving high accuracy and interpretability. The proposed models, particularly the VC, are suitable for deployment in resource-constrained environments where transparency and efficiency are essential. To further substantiate the practicality of deploying the VC in real-time phishing detection systems, we evaluated its inference performance. Benchmark tests conducted on standard hardware (Intel Core i7 CPU and 16GB RAM) showed an average prediction latency of approximately 2 milliseconds per sample, with a minimal serialized model size of less than 5 MB. These results underscore the classifier's efficiency, making it highly suitable for integration into real-time threat detection frameworks, such as browser security extensions or mobile applications, where computational resources and response times are critical factors.

However, challenges remain—particularly in handling adversarially crafted phishing attempts and maintaining detection quality across evolving attack vectors. Future work will explore adversarial defense mechanisms, domain adaptation, and real-time classification using streaming web data. Incorporating contextual and behavioural features from multilingual phishing websites may also enhance the results.

5. CONCLUSION

This study presents a comprehensive evaluation of classical and ensemble ML models for detecting phishing websites, utilizing a dataset of 10,000 labeled web pages and 48 extracted features. The experimental results demonstrate that ensemble methods, particularly the soft VC, deliver state-of-the-art performance with an accuracy and F1-score of 98.82%, surpassing the performance of most individual classifiers. Our findings affirm the effectiveness of ensemble learning in improving both accuracy and robustness for phishing detection tasks. Moreover, the proposed model exhibits strong suitability for real-time deployment, with an average inference time of approximately 2 milliseconds per sample and a compact model size under 5 MB, ensuring compatibility with resource-constrained environments. This work contributes to the field by demonstrating that lightweight, interpretable models can match or exceed the performance of more complex alternatives, while maintaining efficiency and scalability. Future research will explore the integration of behavioral and temporal features, as well as NLP techniques, to further enhance model generalizability and resilience against evolving phishing strategies.

ACKNOWLEDGMENTS

We want to express our sincere gratitude to Al-Isra University and the University of Petra for granting us access to their Cyber Lab facilities. The resources and support provided were invaluable to the success of this research. Thank you to the faculty and staff for their assistance and for fostering an environment that promotes academic and practical growth in cybersecurity.

FUNDING INFORMATION

This section should describe sources of funding agency that have supported the work. Authors should state how the research described in their article was funded, including grant numbers if applicable. Example: Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Mahmoud Baklizi	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Jamal Zraqou	✓	✓	✓			✓			✓	✓			✓	✓
Mohammad Alkhazaleh				✓	✓	✓	✓		✓	✓	✓			✓
Issa Atoum			✓	✓	✓	✓	✓	✓	✓	✓				
Faisal Alzyoud			✓	✓	✓	✓	✓	✓	✓	✓				
Musab B. Alzghoul				✓	✓				✓	✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] M. Baklizi, "Stabilizing average queue length in active queue management method," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 3, pp. 77-83, 2019, doi: 10.14569/IJACSA.2019.0100310.
[2] M. Baklizi, J. Ababneh, M. M. Abualhaj, N. Abdullah, and R. Abdullah, "Markov-modulated bernoulli dynamic gentle random early detection," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 20, pp. 6688-6698, 2018.
[3] Y. Fang, C. Zhang, C. Huang, L. Liu, and Y. Yang, "Phishing Email Detection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism," *IEEE Access*, vol. 7, pp. 56329-56340, 2019, doi: 10.1109/ACCESS.2019.2913705.
[4] A. K. Dutta, "Detecting phishing websites using machine learning technique," *PLoS ONE*, vol. 16, no. 10, Oct. 2021, doi: 10.1371/journal.pone.0258361.
[5] M. Baklizi, I. Atoum, N. Abdullah, O. A. Al-Wesabi, A. A. Otoom, and M. A. S. Hasan, "A Technical Review of SQL Injection Tools and Methods: A Case Study of SQL Map," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 10, no. 3, pp. 75-85, 2022.
[6] M. Baklizi, I. Atoum, M. A. S. Hasan, N. Abdullah, O. A. Al-Wesabi, and A. A. Otoom, "Prevention of Website SQL Injection Using a New Query Comparison and Encryption Algorithm," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 1, pp. 228-238, 2023.
[7] M. K. Baklizi *et al.*, "Web Attack Intrusion Detection System Using Machine Learning Techniques," *International Journal of Online and Biomedical Engineering*, vol. 20, no. 3, pp. 24-38, 2024, doi: 10.3991/ijoe.v20i03.45249.
[8] M. N. Trisolvena and N. H. Saputra, "Phishing Cyber Security Threats," *Jurnal Improsci*, vol. 2, no. 1, pp. 38-48, 2024, doi: 10.62885/improsci.v2i1.440.
[9] A. Awasthi and N. Goel, "Phishing website prediction using base and ensemble classifier techniques with cross-validation," *Cybersecurity*, vol. 5, no. 1, pp. 1-23, 2022, doi: 10.1186/s42400-022-00126-9.
[10] N. Abdullah *et al.*, "IoT-Based Waste Management System in Formal and Informal Public Areas in Mecca," *International Journal of Environmental Research and Public Health*, vol. 19, no. 20, pp. 1-31, 2022, doi: 10.3390/ijerph192013066.
[11] M. A. Khan, "Phishing Website Detection System Using Machine Learning," *Journal of Networking and Communication Systems (JNACS)*, vol. 7, no. 2, pp. 1-44, 2024, doi: 10.46253/jnacs.v7i2.a1.
[12] A. Chawla, "Phishing website analysis and detection using Machine Learning," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 10, no. 1, pp. 10-16, 2022, doi: 10.18201/ijisae.2022.262.
[13] S. Alnemari and M. Alshammari, "Detecting Phishing Domains Using Machine Learning," *Applied Sciences*, vol. 13, no. 8, pp. 1-16, 2023, doi: 10.3390/app13084649.
[14] A. Testas, "Support Vector Machine Classification with Pandas, Scikit-Learn, and PySpark," in *Distributed Machine Learning with PySpark*. Berkeley, CA: Apress, 2023, pp. 259-280.

[15] L. Tang and Q. H. Mahmoud, "A Survey of Machine Learning-Based Solutions for Phishing Website Detection," *Machine Learning and Knowledge Extraction*, vol. 3, no. 3, pp. 672-694, 2021, doi: 10.3390/make3030034.
[16] K. Omari, "Comparative Study of Machine Learning Algorithms for Phishing Website Detection," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 9, pp. 417-425, 2023, doi: 10.14569/IJACSA.2023.0140945.
[17] A. K. Jain and B. B. Gupta, "PHISH-SAFE: URL features-based phishing detection system using machine learning," in *Advances in Intelligent Systems and Computing*, vol. 729, 2018, pp. 467-474.
[18] I. Salihovic, H. Serdarevic, and J. Kevric, "The Role of Feature Selection in Machine Learning for Detection of Spam and Phishing Attacks," in *Lecture Notes in Networks and Systems*, vol. 60, 2019, pp. 476-483.
[19] G. H. Lokesh and G. B. Gowda, "Phishing website detection based on effective machine learning approach," *Journal of Cyber Security Technology*, vol. 5, no. 1, pp. 1-14, 2021, doi: 10.1080/23742917.2020.1813396.
[20] M. Ramaiah *et al.*, "Enhanced Phishing Detection: An Ensemble Stacking Model with DT-RFECV and SMOTE," *Applied Mathematics and Information Sciences*, vol. 18, no. 6, pp. 1481-1493, 2024, doi: 10.18576/amis/180624.
[21] S. I. Nova, A. Das, and A. Dey, "Multi-Feature Extraction-Based Phishing Website Detection Using Ensemble Learning," in *2024 27th International Conference on Computer and Information Technology (ICCIT)*, Cox's Bazar, Bangladesh, 2024, pp. 861-866, doi: 10.1109/ICCIT64611.2024.11021929.
[22] S. R. A. Samad *et al.*, "Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection," *Electronics*, vol. 12, no. 7, pp. 1-26, 2023, doi: 10.3390/electronics12071642.
[23] D. Sarma, T. Mittra, R. M. Bawm, T. Sarwar, F. F. Lima, and S. Hossain, "Comparative Analysis of Machine Learning Algorithms for Phishing Website Detection," in *Lecture Notes in Networks and Systems*, vol. 173 LNNS, 2021, pp. 883-896.
[24] M. Almseidin, A. M. A. Zuraiq, M. Al-Kasassbeh, and N. Alnidami, "Phishing detection based on machine learning and feature selection methods," *International Journal of Interactive Mobile Technologies*, vol. 13, no. 12, pp. 171-183, 2019, doi: 10.3991/ijim.v13i12.11411.
[25] O. Ajayi, A. Adetunmbi, T. Olowookere, and S. Sodiya, "Performance evaluation of ensemble learning algorithms and classical machine learning algorithms for phishing detection," in *Proceedings of the 2002 Smart, Secure and Sustainable Nation*, Abuja, Nigeria, 2022, pp. 1-13.
[26] A. Basit, M. Zafar, A. R. Javed, and Z. Jalil, "A Novel Ensemble Machine Learning Method to Detect Phishing Attack," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, Bahawalpur, Pakistan, Nov. 2020, pp. 1-15, doi: 10.1109/INMIC50486.2020.9318210.
[27] N. F. Almujahid, M. A. Haq, and M. Alshehri, "Comparative evaluation of machine learning algorithms for phishing site detection," *PeerJ Computer Science*, vol. 10, 2024, doi: 10.7717/peerj-cs.2131.
[28] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q. E. U. Haq, K. Saleem, and M. H. Faheem, "A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN," *Electronics*, vol. 12, no. 1, pp. 1-8, 2023, doi: 10.3390/electronics12010232.
[29] K. L. Chiew, C. L. Tan, K. Wong, K. S. C. Yong, and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Information Sciences*, vol. 484, pp. 153-166, 2019, doi: 10.1016/j.ins.2019.01.064.
[30] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing Detection System Through Hybrid Machine Learning Based on URL," *IEEE Access*, vol. 11, pp. 36805-36822, 2023, doi: 10.1109/ACCESS.2023.3252362.
[31] L. R. Kalabarige, R. S. Rao, A. Abraham, and L. A. Gabralla, "Multilayer Stacked Ensemble Learning Model to Detect Phishing Websites," *IEEE Access*, vol. 10, pp. 79543-79552, 2022, doi: 10.1109/ACCESS.2022.3194672.
[32] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *arXiv*, 2020, doi: 10.48550/arXiv.2010.16061.
[33] V. M. Rios, P. R. M. Inácio, D. Magoni, and M. M. Freire, "Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms," *Computer Networks*, vol. 186, 2021, doi: 10.1016/j.comnet.2020.107792.

BIOGRAPHIES OF AUTHORS

Mahmoud Baklizi 17 years of experience in academia in Jordan. Extensive theoretical and practical expertise in network security, network design, and implementation, with additional proficiency in security, vulnerability assessment, and programming. Several years of managerial experience in the academic sector in Jordan, with strong capabilities in strategic and action plan development. Interdisciplinary and intercultural experience in scientific research, applied research, and research and development within both academic and industrial fields in Jordan and Malaysia. Advanced proficiency in English. He can be contacted at email: mbaklizi@uop.edu.jo.

Jamal Zraqou is an Associate Professor at the Department of Computer Science/Virtual and Augmented Reality, University of Petra, Jordan, where he has been a faculty member since 2022. From 2018-2029, he was also the Dean faculty of IT at IU. His research interests include computer vision, virtual and augmented reality, IoT, cyber security, and image processing. He can be contacted at email: Jamal.Zraqou@uop.edu.jo.

Mohammad Alkhazaleh received the degree in information technology and computing from Arab Open University (AOU), Jordan in 2009. He received the master's degree in computer science from The Middle East University for the Graduate Studies (MEU), Jordan in 2011. He received the Ph.D. degree in computer engineering from University Malaysia Perlis (UniMAP), Malaysia in 2021. He was a lecturer at Faculty of Applied Studies and Continuing Education at Al-Baha University (2011-2016), and is currently an Assistant Professor in Department of Computer Sciences at the College of Information Technology at Isra University (2022-present). He can be contacted at email: m.alkhazaleh@iu.edu.jo.

Issa Atoum is an Associate Professor in the Department of Software Engineering, Faculty of Information Technology, Philadelphia University, Amman, Jordan. He can be contacted at email: iatoum@philadelphia.edu.jo.

Faisal Alzyoud is an Associate Professor in the Department of Cyber Security, Faculty of Information Technology, Isra University (IU), Amman, Jordan. He can be contacted at email: faisal.alzyoud@iu.edu.jo.

Musab B. Alzghoul received his Bachelor of Science in computer science in Don state technical university (DSTU) Russia in 2005. In 2006, he received his master's degree in computer science at DSTU, and then in 2009, he received a Ph.D. in Computer Science. He was a member of the Faculty of Information Technology at Zarqa University where he was a lecturer between 2009 and 2012. He was a lecturer in 2012-2023 at the Faculty of Information Technology of Um-AlQura University. He is now an Assistant Professor in the Department of Computer Sciences at the Isra University where he has been teaching since 2023. He can be contacted at email: musab.alzgool@iu.edu.jo.