applied sciences

MDPI

Article

Detecting Phishing Domains Using Machine Learning

Shouq Alnemari * and Majid Alshammari *

Collage of Computer and Information Technology, Taif University, Taif 26571, Saudi Arabia
* Correspondence: shougalnemari@gmail.com (S.A.); m.alshammari@tu.edu.sa (M.A.)

Abstract: Phishing is an online threat where an attacker impersonates an authentic and trustworthy organization to obtain sensitive information from a victim. One example of such is trolling, which has long been considered a problem. However, recent advances in phishing detection, such as machine learning-based methods, have assisted in combatting these attacks. Therefore, this paper develops and compares four models for investigating the efficiency of using machine learning to detect phishing domains. It also compares the most accurate model of the four with existing solutions in the literature. These models were developed using artificial neural networks (ANNs), support vector machines (SVMs), decision trees (DTs), and random forest (RF) techniques. Moreover, the uniform resource locator's (URL's) UCI phishing domains dataset is used as a benchmark to evaluate the models. Our findings show that the model based on the random forest technique is the most accurate of the other four techniques and outperforms other solutions in the literature.

Keywords: phishing detection; machine learning; phishing domains; artificial neural networks; support vector machine; decision tree; random forest

Citation: Alnemari, S.; Alshammari, M. Detecting Phishing Domains Using Machine Learning. *Appl. Sci.* **2023**, *13*, 4649. https://doi.org/10.3390/app13084649

Academic Editor: Luis Javier García Villalba

Received: 23 January 2023
Revised: 17 March 2023
Accepted: 4 April 2023
Published: 7 April 2023

1. Introduction

Phishing is an online crime that tries to trick unsuspecting users into exposing their sensitive (and valuable) personal information. This can include usernames, passwords, financial account details, login credentials, personal addresses, and social relationships, which the attacker then uses for malicious purposes, such as identity theft. Phishing is usually perpetrated by a hacker disguising themself as a trustworthy entity, an effect achieved by combining both social engineering and technical tricks.

Phishing domains are one type of attack. These domains obtain sensitive information without authorization, either through blackmail or by directing users to a fake website that looks similar to a real one. Both then request personal information. Security breaches occur when users enter their private data into these sites, as the assailant now has personal information that may be used to commit identity theft.

Most financial and government institutions have improved their direct internet offerings to combat potential security breaches such as phishing domains. However, as services on the Internet continue to grow, so has the public's reliance on online services. Despite the risks that phishing attacks pose, online shopping, banking, and bill payment have all become popular in the United States and developed European countries. Successful phishing attempts have had an impact on global finances, raising the risk for both clients and businesses and further increasing the need for protection online. The process of defending cyberspace against threats such as phishing is known as cybersecurity [1–3]. Protecting internet-connected resources from cyber-attacks is cybersecurity's main goal [4–6].

Cybersecurity is becoming increasingly complicated as cyber-attacks become more complex and more frequent, making it difficult to recognize, assess, and handle significant risk events. The Anti-Phishing Working Group (APWG) discovered more than 51,000 distinct phishing websites. According to the Rivest–Shamir–Adleman (RSA) analysis, phishing attacks cost global enterprises $9 billion in 2016 [7]. Over one million phishing

attacks were listed in 2016, a 65% increase from the previous year [8]. The frequency of these attacks erodes consumers' trust in social works, such as webpages.

There are various types of web fraud [9], and phishing websites are a common entry point for online social engineering attempts. To start, the hacker creates a webpage by impersonating a reputable website. They then send these fishy URLs to potential victims via spam chats, messages, or social media sites, hoping that unsuspecting users will believe it is a real URL [10]. If users enter their personal information (bank account numbers, government savings numbers, and so on) at the link sent by the hacker, that data will be compromised.

There are a lot of strategies to combat phishing [11]. Artificial intelligence (AI) has had a huge impact on almost every industry, including cybersecurity because AI can detect spam, phishing, spear phishing, and other assaults using past attacks in the form of datasets.

This paper develops and compares the effectiveness of machine learning (ML) classification models in detecting phishing domains. The goal is to improve detection by using the most accurate model of the four to predict if a webpage is a phish or legal. A phished domain is difficult to analyze and comprehend since it involves social and technical issues for which there is no one-size-fits-all analysis. As a result, all phishing domain causes and features were analyzed quantitatively and qualitatively to determine where to focus the model to better decrease the danger arising from a visit to a phishing website, particularly regarding consumer trust.

The rest of this paper is organized as follows: Section 2 reviews the latest research on phishing attacks. The data used, and the solutions based on ML classifiers are presented in Section 3. The proposed model architecture is presented in Section 4. Section 5 includes the results and evaluation of the proposed model. Finally, Section 6 presents the paper's conclusion and directions for future work.

2. Background

Common machine learning classification techniques have proven efficient in phishing domain detection, including the following:

2.1. Decision Tree

A decision tree helps individuals make better decisions via a tree-like graph or modeling of alternatives and their possible implications, such as likely outcomes, resource costs, and utility. It is one strategy of many to demonstrate an algorithm completely made up of conditional control statements [12]. Decision trees are frequently used to analyze the underlying relationships in big datasets. The decision tree's goal is to observe a process; by doing this, researchers can utilize its attributes, allowing it to be assigned to a certain class, as shown in Figure 1, which shows a training algorithm that creates the structure of such a decision tree. After its construction, a decision tree may be used to assess further samples with variable degrees of success, depending on how well it represents the dataset. The success rate is determined by several aspects, including the size of the dataset used to create the tree, the class-wise overlapping of variable observations, the algorithm used to build the tree, and the usage of extra methods to enhance tree development.

The root node is the starting point of the decision tree (also known as the parent node). It represents the full dataset, which is then split into two or more homogenous groups, also called child nodes. These eventually lead to the leaf nodes, which are the tree's final output, after which no further splits are possible. Splitting is the process of separating the root node into sub-nodes based on the conditions specified. A branch/sub-tree is a tree that has been created by splitting a larger tree. Pruning is the removal of unwanted branches.

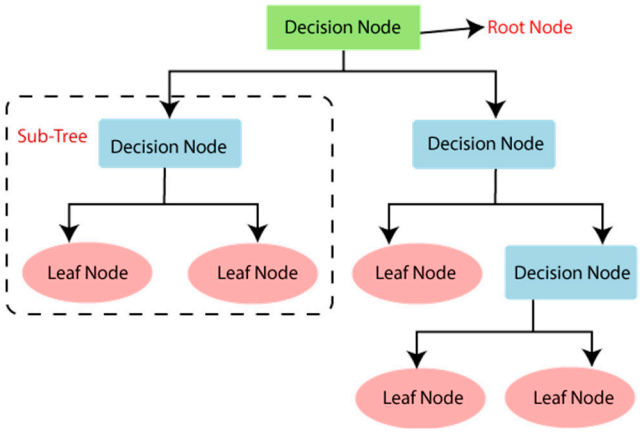

Figure 1. Decision tree algorithm [13].

2.2. Random Forest

Random forest is an ensemble of supervised learning algorithms for classification and regression used in predictive modeling and machine learning techniques [14]. The random forest has attracted the attention of academics because of its speed and accuracy in categorization. It gathers the results and predictions of several decision trees to choose the best output: the mode of the classes (the value that appears most often in the decision tree results) or mean prediction. Random forest splits the dataset into two sections: the training set and the test set. It then randomly selects multiple samples from the training set. Next, the researcher uses the decision tree for each sample, which divides each selection into two daughters using best division. Thereafter, users must repeat the last step to vote for each prediction result and select the most voted prediction as the final result. The main hyper-parameters in the random forest are used to either increase the predictive power of the model or to make the model faster [15]. In this context, a higher number of trees can increase the performance as well as make the predictions more stable, but it also increases the processing time. The employment of a maximum number of features, in addition to a minimum number of leaves, may improve algorithm performance.

Once the training step is completed, the model can be applied to a test dataset. This procedure allows for the estimation of predictions and then for the comparison of the results against the expected values [16]. Figure 2 shows how each tree is responsible for producing a distinct output after being fed an independent random sample vector. The random forest is used for its error generalization technique, and the random forest's accuracy improves as the forest grows in size. After randomly picking the features for the error rate, the accuracy is entirely dependent on the correlation between the trees. The random forest's characteristics might be created by tracking the error and correlation between nodes. As a consequence, the relevance of a variable can be measured [17].

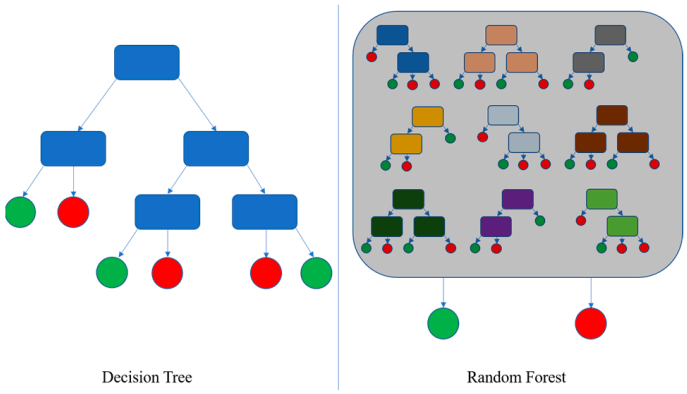

Figure 2. A comparison of DT and RF [18].

2.3. Support Vector Machine

SVM is a supervised learning method based on statistical learning theory utilized for pattern identification and regression. Statistical learning theory can pinpoint the factors needed to successfully learn specific, easy algorithms; real-world applications frequently require more complicated tools and algorithms (such as neural networks), which are much more difficult to analyze theoretically. SVMs are the meeting point of learning theory and practice. They create models that are both complicated (including a huge class of neural networks, for example) and simple enough to be mathematically examined. This is because an SVM is a linear algorithm in a high-dimensional space [19]. As shown in Figure 3, SVM predicts labels by generating a decision boundary, such as a hyperplane, between two specified classes with a minimum of one label. The data points and support vectors are handled by the hyperplane. It takes advantage of the distance between data points to categorize each class independently.

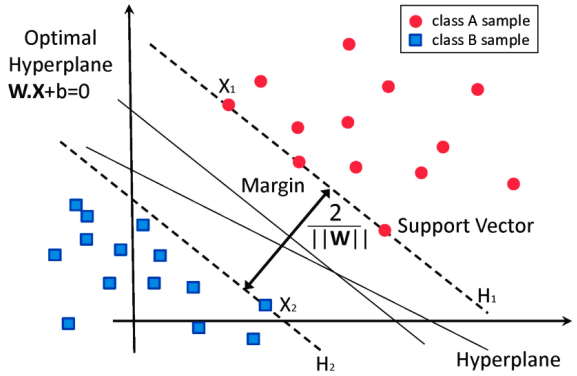

Figure 3. Support vector machine [20].

Previous research has demonstrated that the hyperplane with the greatest margin of separation between the two classes offers the highest generalization performance [21]. The best hyperplane is found by solving a convex optimization problem involving the minimization of a quadratic function under linear inequality constraints. The answer may be expressed in terms of support vectors, which are a subset of the training instances. Support vectors include all the information required to solve a classification issue since the result will remain the same even if all other vectors are removed.

2.4. Ensemble Classification Techniques

Building a fair model from a dataset is one of the main goals of machine learning algorithms. Learning, or training, is the process of developing models from data, and the learned model is referred to as a hypothesis or learner. Ensemble methods learn algorithms that create a set of classifiers and then use their predictions to put new data points into categories [22].

Ensembles are far more accurate than the individual classifiers that make them up. Ensemble methods, also known as committee-based learning or learning multiple classifier systems, are used to train numerous hypotheses to solve a problem. Random forest trees are a common form of ensemble modeling in which many decision trees are utilized to predict outcomes. Figure 4 shows a general ensemble architecture [23].

An ensemble is made up of numerous hypotheses or learners that are produced from training data using a basic learning method. Most ensemble methods produce homogeneous base learners or homogeneous ensembles using a single-based learning algorithm, but some approaches use multiple learning algorithms to build heterogeneous ensembles. The ability of ensemble approaches to enhance weak learners is well established. Below are three major types of meta-algorithms that are regularly used in ensemble approaches [23].

2.4.1. Bagging

Bagging, or bootstrap aggregation, is a powerful, effective, and simple ensemble method [24]. The method uses bootstrapping to sample several copies of a training set. It may be applied with any form of classification or regression model, as demonstrated in Figure 4. Bagging works well with nonlinear models that are unstable.

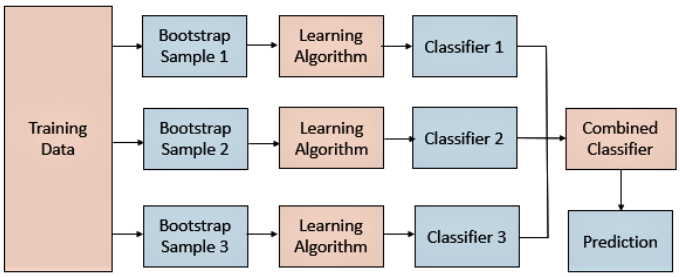

Figure 4. Ensemble learning (bagging) [25].

2.4.2. Boosting

Boosting is a meta-algorithm that can be thought of as a method of model averaging [26]. It is the most popular ensemble approach, as well as one of the most effective learning concepts. This method was created for classification, but it can also be applied to regression. The original boosting algorithm created a strong learner by combining three weak learners.

2.4.3. Stacking

Stacking is the process of integrating numerous classifiers created by various learning algorithms into a single dataset of feature vector pairs and their classifications [27]. A set of base-level classifiers is constructed in the first phase, and a meta-level classifier is trained in the second phase, as shown in Figure 5.

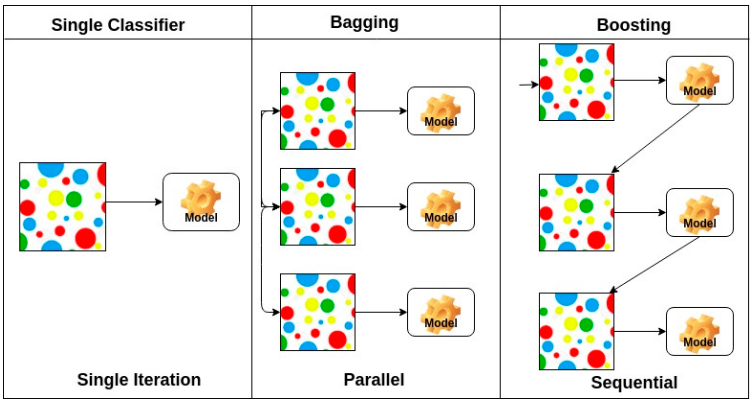

Figure 5. Ensemble learning, bagging, and boosting machine learning techniques [28].

2.5. Ensemble Classification Techniques

A neural network (NN) is a mathematical model that mimics the behavior of biological neurons and the nervous system. ANNs utilize technological solutions to imitate the architecture and functions of the neural system of human brains [29]. They use neural network topologies to represent physical systems in this way. McCulloch and Pitts introduced the ANN theory for the first time in [30]. ANNs are appropriate for addressing the mapping issue from one dataset to another when they have strong nonlinear mapping capabilities [31]. ANNs can be categorized into two types of signal transmission modes: feedforward and feedback neural networks, each of which has a distinct framework. Feedback neural networks play a significant role in AI; however, they have only been used in a few applications

due to solid waste concerns. In the application of biosorption capacity, several researchers compared the models of feedforward neural networks such as multilayer perception ANNs and feedback neural networks, which found that feedforward neural networks had lower prediction errors than feedback neural networks [32].

In a multilayer feedforward neural network, neurons in one layer communicate with those in the next layer through various weighted linkages. There are three kinds of neuron layers: input, hidden, and output. The neurons in the input layer receive external data, such as from sensory receivers; the neurons in the hidden layer imitate a biological neural network to transmit that data, and the neurons in the output layer offer a judgment output. Although several hidden layers are feasible, typically, only one hidden layer is employed, especially with small sample sizes. Neurons only link between layers, not inside them. In a feedforward neural network, signals can only go one direction, from input to output [33]. ANNs have been extensively employed in numerous activities, including environmental difficulties and even solid waste-related issues, due to these simplifications.

Complex systems and correlations in labeled data are recognized using these models. Deep neural networks (DNNs) are more complicated neural networks with hidden layers that conduct much more complex functions than basic sigmoid or ReLU activations [34]. The architecture of a deep learning model is shown in Figure 6.

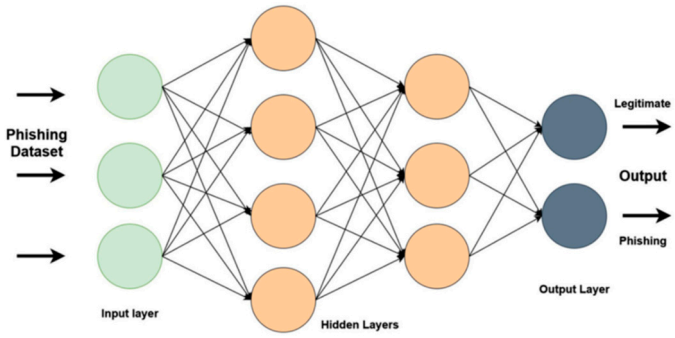

Figure 6. A deep neural network used for phishing detection [35].

3. Related Work

Users, overall, tend to overlook a website's URL. This makes them more likely to fall prey to a phishing domain, which might otherwise be avoided by determining whether a URL is authentic. Unfortunately, traditional methods for detecting phishing attacks have limited accuracy and can only detect roughly 20% of attempts. ML techniques for phishing detection produce better results, but they are time-consuming, even on small databases, and they are not scalable. Furthermore, heuristics-based phish detection has a significant false-positive rate. Previous research on anti-phishing models has concentrated on strategies to modify efficiency. Even so, feature reduction and the use of an ensemble model can improve these models' accuracy even further.

For phishing domain detection, machine learning algorithms are prevalent, and using them has become a straightforward categorization problem. The data at hand must have properties relevant to phishing and legitimate website classes to build an ML-based detection model. Previous works have shown that when robust machine-learning approaches are utilized, detection accuracy is high. To reduce features, a variety of feature selection strategies are applied.

To train a machine learning model to predict phishing attacks versus legal traffic, a batch of data is given as the input. Dataset visualization becomes more efficient and intelligible when characteristics are reduced. The DT, C4.5, k-NN, and SVM algorithms are the most important classifiers; they have been utilized in numerous research projects, and they have detected phishing attacks with the greatest accuracy and efficiency. According to the empirical experiment's findings, manual parameter adjustment, protracted training

periods, and poor detection accuracy are prevalent problems with modern deep learning systems [36].

Despite these benefits, researchers have noted the limits of their studies. Many pointed out that ensemble learning techniques have not been applied and that feature selection and reduction have not been performed. A range of strategies has been applied to combat phishing attacks. One paper [37] used different classifiers, such as naive Bayes and SVM. Similarly, the authors in [38] utilized random forest to differentiate phishing attacks from normal websites.

Subasi et al. [17] reported that their proposed classifiers were extremely effective at classifying phishing websites. They reported that random forest was the most accurate classifier, at 97.26%.

The authors of [39] proposed a paper concentrating on feature selection in phishing websites. They sorted the characteristics into six groups using the UCI dataset, which has more than 11,000 URLs and 30 characteristics. They chose three groups and decided that these were the best solutions for detecting phishing attacks accurately.

Patil et al. [40] suggested three strategies for detecting phishing websites. The first entailed assessing various URL attributes; the second determined the validity of the website by determining where it was hosted and who managed it; and the third method determined the authenticity of the website through visual, appearance-based analysis. They used ML methodologies and algorithms to assess the numerous aspects of the URLs and websites.

Joshi et al. [41] used a binary classifier based on an RF algorithm and a feature selection algorithm based on the reliefF algorithm. They utilized data from the Mendeley domain as the source for their feature selection algorithm. They then used the selected features to train an RF algorithm to predict phishing attacks.

The work of Ubing et al. [42] employed three ensemble learning strategies: bagging, boosting, and stacking. Their dataset had 30 characteristics and 5126 records in the result column. The data comes from UCI, which is open to the public. They integrated their classifiers to achieve the highest level of accuracy possible from a DT.

The authors of [43] suggested a new method based on both URLs as inputs and HTML-related data. After the features were extracted, a stacking strategy merged the learners. The researchers then ran tests on a variety of datasets, including 2000 webpages taken from Phishtank (1000 legitimate and 1000 phishing sites). The second dataset came from Alexa and contained nearly 50,000 websites. To improve their accuracy, they used SVMs, NNs, DT, and RF, which they combined through stacking. This study obtained a high level of accuracy using a variety of classifiers.

The authors of [44] looked at how stacking techniques could be used to identify phishing websites. The goal of these tests was to enhance precision metrics using PCA and stacking the most efficient classifiers. Other classifiers using proposed features N1 and N2 outperformed stacking (RF, NN, stowing). The tests were carried out using datasets from phishing websites. With 11,055 web pages, the dataset had 32 preprocessed characteristics.

Another strategy is the extra-tree base classifier utilized by the authors of [45], who used it to classify several meta models: AdaBoost, bagging, rotation forest, and LogitBoost-Extra Tree. The suggested models outperformed current ML-based phishing attack detection models, and, as a result, the authors recommended using meta-algorithms to create phishing attack detection models.

To improve the detection of phishing websites, the authors of [46] suggested a phishing detection model based on a particle swarm optimization (PSO) algorithm. Their proposed method used PSO to weigh distinct websites, resulting in increased accuracy for classifying abnormal phishing websites. PSO weighting distinguishes different aspects of a website, considering how important they are in detecting phishing from legitimate websites. According to the findings, their proposed PSO-based component weighting improved the ML model's ability to recognize and monitor both phishing and legitimate websites individually.

The authors of [47] employed an evolutionary neuro-fuzzy intelligence system-based resilient approach with integrated features to identify and guard against phishing attacks.

The authors of [48] introduced the PhishBench benchmarking structure, which permits researchers to evaluate the characteristics of phishing attacks and fully comprehend different evaluation circumstances, unified framework specifications, data, machine learning algorithms, and evaluation metrics. When the proportion of phishing and authentic traffic fell from one to 10, the classification execution was reduced. In terms of the F1 score, the drop in execution ranged from 5.9% to 42%.

An intelligent phishing website identification method was proposed by Subasi and Kremic [49]. They used proprietary machine learning approaches to differentiate phishing websites. Several classifier approaches were applied to create a reliable and intelligent phishing detection system. The performances of their ML approaches were evaluated using ROC area, F-measure, and AUC. With a 97.61% accuracy, Adaboost with SVM outperformed all other classification approaches.

Alternatively, Mao et al. [50] developed a learning-based technique for determining page design comparability, which might be utilized to identify phishing attack pages. They built a phishing classifier using dual ML algorithms, a support vector machine, and a decision tree for effective page layout aspects. They used genuine website page testing from phishtank.com and alexa.com to validate their methodology.

Tyagi et al. [51] employed a dataset from the University of California at Irvine's machine learning repository, which had 2456 unique URLs and more than 11,000 URLs, with 6157 phishing and 4898 normal URL. They took 30 characteristics from the URLs and utilized them to forecast attacks. They employed DT, RF, gradient boosting, generalized linear, and PCA as machine learning techniques.

Chen and Chen [52] employed the SMOTE approach to increase their model's detection coverage. They trained machine learning models such as bagging, RF, and XGboost. The XGboost approach, which they proposed, yielded the maximum accuracy. They utilized the Phishtank database, which contained over 24,000 phishing and 4000 legitimate websites.

Alternatively, Abdelhamid et al. [53] developed a model content and feature comparison to detect attacks. They used a PhishTank dataset with approximately 11,000 samples. They utilized a technique called enhanced dynamic rule induction, which they said was the first machine learning and deep learning algorithm to be used as an anti-phishing tool. With two major threshold frequencies and rule strength, this algorithm passed datasets. Only "strong" characteristics were stored in the training dataset, and these features became part of the rule, while others were eliminated.

A study by Jain and Gupta [54] tested two databases. Their model was more accurate on Phishtank, which has over 1500 phishing URLs, followed by Openphish, which has over 600 phishing URLs and 1600 real URLs, as well as 66 valid URLs and 252 legal URLs. They enhanced phishing detection accuracy using machine learning methods such as RF, SVM, NN, logistic regression (LR), and NB. On the client side, they employed a successful feature extraction approach.

Lakshmi et al. [55] suggested a novel method for detecting phishing websites by looking for hyperlinks in the source code of the corresponding website's HTML page. The suggested method employed a feature vector with 30 parameters to detect malicious online pages. These characteristics were used to train a supervised DNN model with an Adam optimizer to distinguish between fraudulent and legitimate websites. To do so, the model employed a listwise process. When compared to other traditional ML algorithms such as SVM, Adaboost, and AdaRank, the proposed model outperformed the others, with a 96% accuracy rate.

Table 1 presents the summary of ML approaches for phishing website detection. The next table shows that some studies provide highly efficient results using ML for phishing attack detection.

Table 1. Comparison table of the latest research focusing on machine learning phishing detection techniques.

Model	Dataset	Algorithm	Accuracy
James et al. [37]	URLs	IBK, SVM, NB	89.75%
Subasi et al. [17]	website	ANN, KNN, RF, SVM, C4.5, RF	97.36%
Mao et al. [50]	Websites	SVM, RF, DT, AB	93%
Tyagi et al. [51]	URLs	DT, RF, GBM	98.40%
Chen and Chen [52]	websites	ELM, SVM, LR, C4.5, LC-ELM, KNN, XGB	99.2%
Joshi et al. [41]	Websites	RF	97.63%
Ubing et al. [42]	UCI	Ensemble bagging, boosting, stacking	95.4%
Sahingoz et al. [56]	Websites	SVM, DT, RF, KNN, KS, NB	97.98%
Abdelhamid et al. [53]	URLs	eDRI	93.5%
Patil et al. [40]	URLs	LR, DT, RF	96.58%
Jain and Gupta [54]	Websites	RF	99.57%
Jagadeesan et al. [57]	URLs	RF, SVM	95.11%
Niranjan et al. [58]	Websites	RC, kNN, IBK, LR, PART	97.3%
Chiew et al. [59]	URLs	RF, C4.5, PART, SVM, NB	96.17%
Pandey et al. [60]	Websites	SVM, RF	94%
Ali and Ahmed [61]	Websites	Genetic algorithm (GA) + DNN	89.50%
Aljofey et al. [62]	Websites	CNN	95.02%
Shie [63]	Websites	Convolutional auto encoder + DNN	89.00%
Maurya and Jain [64]	Websites	PSL 1 + PART	99.30%
Wang et al. [65]	Websites	RNN + CNN	95.79%
Lakshmi et al. [55]	UCI	DNN +Adam	96.00%
Li et al. [43]	URLs	GBDT, XGBoost and LightGBM	98.60%
Yang et al. [66]	Websites	Auto encoder + NIOSELM	94.60%
Anupam and Arpan [67]	Websites	Grey wolf optimizer + SVM	90.38%

4. Methodology

Utilizing the UCI dataset, four phishing detection models were developed using ANN, SVM, DTs, and RF algorithms. The MinMax normalization feature was employed as a preprocessing strategy to improve the models' accuracy. The proposed models were able to detect different types of attacks from the UCI dataset.The following subsections discuss the dataset used and implemented algorithms; Sections 4.1 and 4.2, respectfully.

4.1. Dataset Used: UCI Phishing Websites

Standard datasets already exist for the development of phishing website detection algorithms. Other studies classified websites to establish a list of legitimate and phishing sites for further consideration. This work, on the other hand, utilizes the freely accessible phishing dataset from UCI machine learning repository that can be found in [68], and was prepared by [69]. This dataset was created to build machine learning-based phishing website detection algorithms. It is comprised of extensive properties that span four distinct categories [70]. They designed and extracted characteristics from the following categories: Address Bar, HTML and JavaScript, Abnormal, and Domain. This study was performed using a phishing domain dataset with 31 attributes that can either take a binary or ternary

value. This dataset has 11,055 records, and each record includes 31 characteristics. The characteristics of the collection are identified by names, such as URL Length, Submitting to Email, Shortening Service, Abnormal URL, Having an At Symbol, and Redirect.

4.2. Implemented Algorithm

To increase accuracy, this paper utilized the MinMax normalization feature as a preprocessing step in each proposed model. Normalization is a useful strategy for improving the accuracy of machine learning models, and it is required for some models to work properly. The MinMax normalization technique in the suggested model compresses the data to a domain of [0, 1], which improves the model training input quality (see Equations (1) and (2)).

$$X\_std = (X - X.min)/(X.max - X.min) \quad (1)$$

$$X\_scalar = X\_std \times (max - min) + min \quad (2)$$

To enhance the model performance and complexities, we used a data normalization strategy, as shown in Table 2. The algorithm selects significant aspects from the initial dataset by determining the prediction outcome, which is performed by filtering it through 30 features. The UCI dataset is split 80/20 into training and testing sets, respectively, by using c5-fold cross-validation, which presented the best performance in the latest research. The prediction model is then taught using machine learning, which employs various learning models. This is particularly useful for making predictions, as utilizing many models ensures that the results are not biased toward a single model. To account for this, we present the results of all the models combined and totaled to establish their maximum accuracies. If most of the models indicate that a domain is phishing, then the model's prediction accuracy confirms that the domain is a phishing attempt.

Table 2. The performance results before and after using the normalization technique.

Classifier	Before Use Normalization	After Use Normalization
SVM	Accuracy: 94.46 Precision: 93.64 Recall: 96.62 F1-measure: 95.10	Accuracy: 94.66 Precision: 93.9 Recall: 96.6 F1-measure: 95.3
ANN	Accuracy: 95.5 Precision: 95.6 Recall: 96.3 F1-measure: 96	Accuracy: 96.2 Precision: 96 Recall: 97.2 F1-measure: 96.6
RF	Accuracy: 96.86 Precision: 96.56 Recall: 97.84 F1-measure: 97.20	Accuracy: 97.3 Precision: 96.9 Recall: 98.62 F1-measure: 97.6
DT	Accuracy: 95.4 Precision: 95.8 Recall: 95.9 F1-measure: 95.8	Accuracy: 96.3 Precision: 96.5 Recall: 96.8 F1-measure: 96.7

5. Model's Flowchart

Phishing is a concern to many individuals. However, existing methods, such as browser security indicators, cannot detect phishing websites. Due to the limits of current technology, users must evaluate whether a URL is phishing or not on their own. As a result, an automated technique for phishing website identification should be explored for increased cyber safety. This study shows how an implemented feature extraction approach and a prediction model based on a random forest classifier help increase the likelihood that a user will correctly identify a phishing website.

Each of the developed models, as shown in Figure 7, employs a feature selection technique to increase its accuracy. The data analysis heat map picks those that are most crucial in affecting the forecasted result by filtering the most interesting features out of the original dataset. As a result, irrelevant features have no effect on the model's efficiency or prediction.

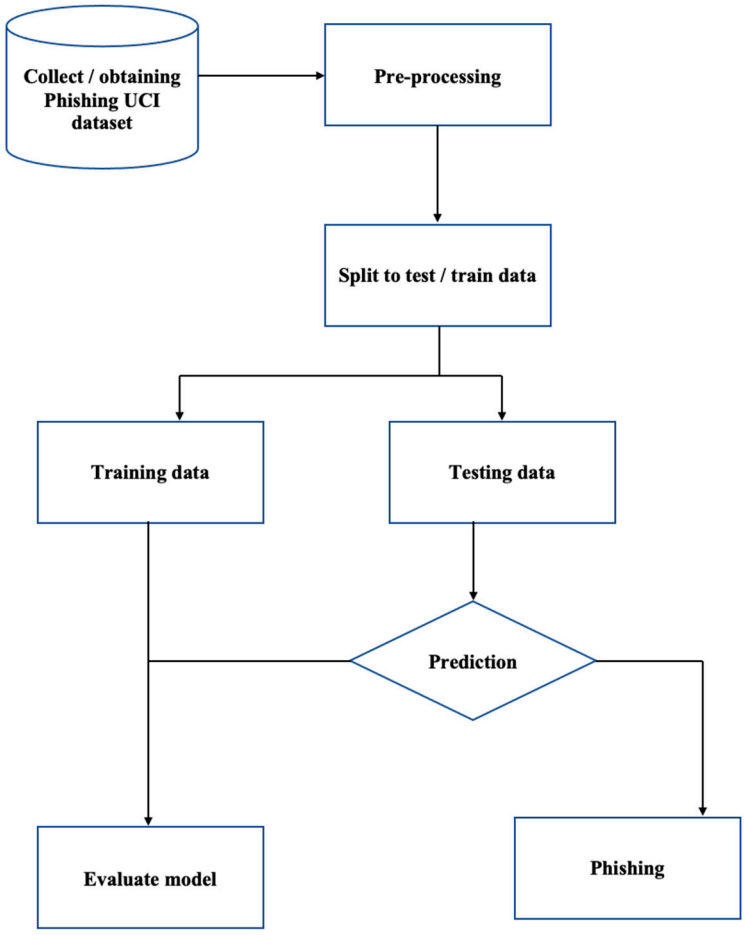

Figure 7. Model's flowchart.

A summary of models' flowchart steps follows:

1. Read the URL's UCI phishing websites dataset.
2. Check the data features.
3. Check the proposed data types.
4. Clean missing values from the data.
5. Split the data into training and testing sets.
6. Train the model using four machine-learning techniques: RF, SVM, DT, and ANN.
7. Evaluate the model's performance to estimate the accuracy and calculate the accuracy results.
8. Select the best model as the final model.

6. Findings and Analysis

To identify the most accurate machine learning model for detecting phishing domains, this paper employed an experimental approach using four ML techniques: SVM, ANN, RF, and DT. With a total of 11,055 data instances, the UCI dataset was utilized for experimentation. Thirty features were used for evaluating the dataset, and the 31st feature was used as the output. Table 3 displays the outcomes of the simulation with the true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), and false negative rate. Moreover, a five-fold cross-validation method was employed for the classification

procedure. The 10-fold cross-validation approach was used to locate a greater performance accuracy dataset. Cross-validation is a predictive performance model evaluation technique used to check a machine-learning algorithm's performance in generating predictions on newer data on which it has not been trained. The examination of the confusion matrix is the basis for the classification technique's result performance.

Table 3. Evaluation results and parameters used of the proposed classifiers.

Classifier	Parameters	TPR	TNR
SVM	Kernel function = rbf	0.92	0.96
ANN	Iterations = 500, Activation = Relu, Optimizer = Adam	0.94	0.96
RF	Trees = 100, Creation = gini	0.96	0.98
DT	Criterion = gini, Splitter = best	0.95	0.96

The results are shown in Table 4. The RF model provided the highest detection accuracy rate at 97%, followed by DT at 96%, ANNs at 95%, and SVM at 94%. Figure 8 depicts these results. Finally, Table 5 compares the RF model to the state-of-the-art results in the literature.

Table 4. Evaluation results in (%).

Classifier	Accuracy	Precision	Recall	F1-Measure
SVM	94.66	93.9	96.6	95.3
ANN	95.5	95.6	96.3	96
RF	97.3	96.9	98.2	97.6
DT	96.3	96.5	96.8	96.7

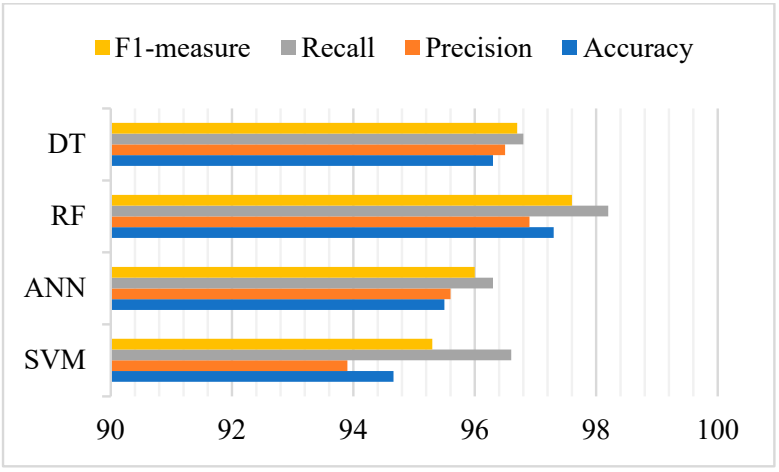

Figure 8. Proposal evaluation results.

Table 5 lists other research dealing with phishing attacks and crucial information about different machine-learning techniques. Three solutions based on ensemble learning, including the bagging, boosting, and stacking methods, were developed by Ubing et al. [42]. They combined their classifiers to attain a 95.4% accuracy rate in their results. Lakshmi et al. [55] proposed a new method for detecting phishing websites by scanning the source code of the related website's HTML page for linkages. They achieved a 96% accuracy rate. The researchers in [71] suggested three meta-learner models using ForestPA; the suggested meta-learners are efficient, according to their experimental data, with the lowest accuracy

at 97.4%. The accuracy values in this paper vary from 0.95 to 0.97%, except for Alsariera et al. [71], who got 97.4%, but this model takes longer to train and implement than RF and DT classifiers.

Table 5. Examining existing phishing domain detection model.

Schemes	Dataset	Algorithm	Accuracy
Ubing et al. [42]	UCI	Ensemble bagging, boosting, stacking	95.4%
Alsariera et al. [71]	UCI	ForestPA-PWDM, Bagged-ForestPA-PWDM, and Adab-ForestPA-PWDM	96.26% 96.5% 97.4%
Lakshmi et al. [55]	UCI	DNN +Adam	96.00%
Random Forest Model	UCI	Random Forest	97.3%

7. Conclusions and Future Works

In this work, we investigated the practicality and the efficiency of using machine learning for phishing detection. We developed four machine learning models based on artificial neural networks (ANNs), support vector machines (SVMs), decision trees (DTs), and random forest (RF) techniques. We then selected the most outperforming model of the fours and compared its performance with other solutions in the literature. The overall results show random forest (RF) model achieved the highest performance and outperforms other schemes in the literature.

Future work includes examining more machine learning algorithm techniques for phishing domains.

Author Contributions: Conceptualization, S.A. and M.A.; methodology, S.A. and M.A; software, S.A. and M.A; validation, S.A. and M.A.; formal analysis, S.A. and M.A.; investigation, S.A. and M.A.; resources, S.A. and M.A.; data curation, S.A. and M.A.; writing—original draft preparation, S.A. and M.A.; writing—review and editing, S.A. and M.A.; visualization, S.A. and M.A.; supervision, M.A.; project administration, M.A.; funding acquisition, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: The researchers would like to acknowledge the Deanship of Scientific Research, Taif University for funding this work.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This work utilizes the freely accessible phishing dataset from UCI machine learning repository that can be found in [68].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cabaj, K.; Domingos, D.; Kotulski, Z.; Respício, A. Cybersecurity Education: Evolution of the Discipline and Analysis of Master Programs. *Comput. Secur.* **2018**, *75*, 24–35. [CrossRef]
2. Iwendi, C.; Jalil, Z.; Javed, A.R.; Reddy, G.T.; Kaluri, R.; Srivastava, G.; Jo, O. KeySplitWatermark: Zero Watermarking Algorithm for Software Protection Against Cyber-Attacks. *IEEE Access* **2020**, *8*, 72650–72660. [CrossRef]
3. Rehman Javed, A.; Jalil, Z.; Atif Moqurrab, S.; Abbas, S.; Liu, X. Ensemble Adaboost Classifier for Accurate and Fast Detection of Botnet Attacks in Connected Vehicles. *Trans. Emerg. Telecommun. Technol.* **2020**, *33*, e4088. [CrossRef]
4. Conklin, W.A.; Cline, R.E.; Roosa, T. Re-Engineering Cybersecurity Education in the US: An Analysis of the Critical Factors. In Proceedings of the 2014 47th Hawaii International Conference on System Sciences, IEEE, Waikoloa, HI, USA, 6–9 January 2014; pp. 2006–2014.
5. Javed, A.R.; Usman, M.; Rehman, S.U.; Khan, M.U.; Haghighi, M.S. Anomaly Detection in Automated Vehicles Using Multistage Attention-Based Convolutional Neural Network. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 4291–4300. [CrossRef]
6. Mittal, M.; Iwendi, C.; Khan, S.; Rehman Javed, A. Analysis of Security and Energy Efficiency for Shortest Route Discovery in Low-energy Adaptive Clustering Hierarchy Protocol Using Levenberg-Marquardt Neural Network and Gated Recurrent Unit for Intrusion Detection System. *Trans. Emerg. Telecommun. Technol.* **2020**, *32*, e3997. [CrossRef]

7. Bleau, H.; Global Fraud and Cybercrime Forecast. Retrieved RSA 2017. Available online: https://www.rsa.com/en-us/resources/2017-global-fraud (accessed on 19 November 2021).
8. Computer Fraud & Security. APWG: Phishing Activity Trends Report Q4 2018. *Comput. Fraud Secur.* **2019**, *2019*, 4. [CrossRef]
9. Hulten, G.J.; Rehfuss, P.S.; Rounthwaite, R.; Goodman, J.T.; Seshadrinathan, G.; Penta, A.P.; Mishra, M.; Deyo, R.C.; Haber, E.J.; Snelling, D.A.W. *Finding Phishing Sites*; Google Patents: Microsoft Corporation, Redmond, WA, USA, 2014.
10. What Is Phishing and How to Spot a Potential Phishing Attack. PsycEXTRA Dataset. Available online: https://www.imperva.com/learn/application-security/phishing-attack-scam/ (accessed on 20 November 2021).
11. Gupta, B.B.; Tewari, A.; Jain, A.K.; Agrawal, D.P. Fighting against Phishing Attacks: State of the Art and Future Challenges. *Neural Comput. Appl.* **2016**, *28*, 3629–3654. [CrossRef]
12. Zhu, E.; Ju, Y.; Chen, Z.; Liu, F.; Fang, X. DTOF-ANN: An Artificial Neural Network Phishing Detection Model Based on Decision Tree and Optimal Features. *Appl. Soft Comput.* **2020**, *95*, 106505. [CrossRef]
13. Machine Learning Decision Tree Classification Algorithm—Javatpoint. Available online: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm (accessed on 25 November 2021).
14. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
15. Friedman, J.H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer Open: Berlin/Heidelberg, Germany, 2017.
16. Brownlee, J. Train-Test Split for Evaluating Machine Learning Algorithms. *Mach. Learn. Mastery* **2020**, *23*. Available online: https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/ (accessed on 25 December 2021).
17. Subasi, A.; Molah, E.; Almkallawi, F.; Chaudhery, T.J. Intelligent Phishing Website Detection Using Random Forest Classifier. In Proceedings of the 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), Ras Al Khaimah, United Arab Emirates, 21–23 November 2017; pp. 1–5.
18. Jeremybeauchamp English: A Visual Comparison between the Complexity of Decision Trees and Random Forests. 2020. Available online: https://commons.wikimedia.org/wiki/File:Decision_Tree_vs._Random_Forest.png (accessed on 27 December 2021).
19. Sönmez, Y.; Tuncer, T.; Gökal, H.; Avcı, E. Phishing Web Sites Features Classification Based on Extreme Learning Machine. In Proceedings of the 2018 6th International Symposium on Digital Forensic and Security (ISDFS), IEEE, Antalya, Turkey, 22–25 March 2018; pp. 1–5.
20. ResearchGate. Figure 2. Classification of Data by Support Vector Machine (SVM). Available online: https://www.researchgate.net/figure/Classification-of-data-by-support-vector-machine-SVM_fig8_304611323 (accessed on 6 October 2021).
21. Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*; Cambridge University Press: Cambridge, UK, 2000.
22. Gomes, H.M.; Barddal, J.P.; Enembreck, F.; Bifet, A. A Survey on Ensemble Learning for Data Stream Classification. *ACM Comput. Surv. CSUR* **2017**, *50*, 1–36. [CrossRef]
23. Zhou, Z.-H. *Ensemble Methods: Foundations and Algorithms*; Chapman and Hall/CRC: London, UK, 2019; ISBN 1-4398-3005-3.
24. Yaman, E.; Subasi, A. Comparison of Bagging and Boosting Ensemble Machine Learning Methods for Automated EMG Signal Classification. *BioMed Res. Int.* **2019**, *2019*, 9152506. [CrossRef]
25. Bagging (Bootstrap Aggregation)—Overview, How It Works, Advantages—Ro.Outletshop2021.Ru. Available online: https://corporatefinanceinstitute.com/resources/data-science/bagging-bootstrap-aggregation/#:~:text=Bagging%20offers%20the%20advantage%20of,of%20interpretability%20of%20a%20model. (accessed on 6 October 2021).
26. Junior, J.R.B.; do Carmo Nicoletti, M. An Iterative Boosting-Based Ensemble for Streaming Data Classification. *Inf. Fusion* **2019**, *45*, 66–78. [CrossRef]
27. Zhou, Z.-H. Ensemble Learning. In *Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 181–210.
28. AdaBoost Classifier in Python—DataCamp. Available online: https://www.datacamp.com/tutorial/adaboost-classifier-python (accessed on 6 October 2021).
29. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.; Arshad, H. State-of-the-Art in Artificial Neural Network Applications: A Survey. *Heliyon* **2018**, *4*, e00938. [CrossRef]
30. McCulloch, W.S.; Pitts, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
31. Jin, D.; Wang, P.; Bai, Z.; Wang, X.; Peng, H.; Qi, R.; Yu, Z.; Zhuang, G. Analysis of Bacterial Community in Bulking Sludge Using Culture-Dependent and-Independent Approaches. *J. Environ. Sci.* **2011**, *23*, 1880–1887. [CrossRef] [PubMed]
32. Liu, Z.-W.; Liang, F.-N.; Liu, Y.-Z. Artificial Neural Network Modeling of Biosorption Process Using Agricultural Wastes in a Rotating Packed Bed. *Appl. Therm. Eng.* **2018**, *140*, 95–101. [CrossRef]
33. Oliveira, V.; Sousa, V.; Dias-Ferreira, C. Artificial Neural Network Modelling of the Amount of Separately-Collected Household Packaging Waste. *J. Clean. Prod.* **2019**, *210*, 401–409. [CrossRef]
34. Basit, A.; Zafar, M.; Liu, X.; Javed, A.R.; Jalil, Z.; Kifayat, K. A Comprehensive Survey of AI-Enabled Phishing attacks detection techniques. *Telecommun. Syst.* **2021**, *76*, 139–154. Available online: https://link.springer.com/article/10.1007/s11235-020-00733-2 (accessed on 27 September 2021). [CrossRef]
35. A Comprehensive Guide to Understand and Implement Text Classification in Python. *Anal. Vidhya* **2018**. Available online: http://www.shivambansal.com/blog/text-classification-guide/ (accessed on 25 October 2021).
36. Sánchez-Paniagua, M.; Fernández, E.F.; Alegre, E.; Al-Nabki, W.; González-Castro, V. Phishing URL Detection: A Real-Case Scenario Through Login URLs. *IEEE Access* **2022**, *10*, 42949–42960. [CrossRef]

37. James, J.; Sandhya, L.; Thomas, C. Detection of Phishing URLs Using Machine Learning Techniques. In Proceedings of the 2013 International Conference on Control Communication and Computing (ICCC), Thiruvananthapuram, India, 13–15 December 2013; Available online: https://ieeexplore.ieee.org/abstract/document/6731669 (accessed on 26 September 2021).
38. Liew, S.W.; Sani NF, M.; Abdullah, M.T.; Yaakob, R.; Sharum, M.Y. An Effective Security Alert Mechanism for Real-Time Phishing Tweet Detection on Twitter—ScienceDirect. *Comput. Secur.* **2019**, *83*, 201–207. Available online: https://www.sciencedirect.com/science/article/pii/S0167404818309040 (accessed on 26 September 2021). [CrossRef]
39. Hutchinson, S.; Zhang, Z.; Liu, Q. Detecting Phishing Websites with Random Forest. In Proceedings of the Machine Learning and Intelligent Communications, Hangzhou, China, 6–8 July 2018; Meng, L., Zhang, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 470–479.
40. Patil, V.; Thakkar, P.; Shah, C.; Bhat, T.; Godse, S.P. Detection and Prevention of Phishing Websites Using Machine Learning Approach. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 19–18 August 2018; pp. 1–5.
41. Joshi, A.; Pattanshetti, P.T.R. *Phishing Attack Detection Using Feature Selection Techniques*; Social Science Research Network: Rochester, NY, USA, 2019.
42. Ubing, A.; Kamilia, S.; Abdullah, A.; Zaman, N.; Supramaniam, M. Phishing Website Detection: An Improved Accuracy through Feature Selection and Ensemble Learning. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 252–257. [CrossRef]
43. Li, Y.; Yang, Z.; Chen, X.; Yuan, H.; Liu, W. A Stacking Model Using URL and HTML Features for Phishing Webpage Detection. *Future Gener. Comput. Syst.* **2019**, *94*, 27–39. [CrossRef]
44. Zamir, A.; Khan, H.U.; Iqbal, T.; Yousaf, N.; Aslam, F.; Anjum, A.; Hamdani, M. Phishing Web Site Detection Using Diverse Machine Learning Algorithms. *Electron. Libr.* **2020**, *38*, 65–80. [CrossRef]
45. Alsariera, Y.A.; Adeyemo, V.E.; Balogun, A.O.; Alazzawi, A.K. AI Meta-Learners and Extra-Trees Algorithm for the Detection of Phishing Websites. *IEEE Access* **2020**, *8*, 142532–142542. [CrossRef]
46. Ali, W.; Malebary, S. Particle Swarm Optimization-Based Feature Weighting for Improving Intelligent Phishing Website Detection. *IEEE Access* **2020**, *8*, 116766–116780. [CrossRef]
47. Adebowale, M.A.; Lwin, K.T.; Sanchez, E.; Hossain, M.A. Intelligent Web-Phishing Detection and Protection Scheme Using Integrated Features of Images, Frames and Text—ScienceDirect. *Expert Syst. Appl.* **2019**, *115*, 300–313. Available online: https://www.sciencedirect.com/science/article/abs/pii/S0957417418304925 (accessed on 26 September 2021). [CrossRef]
48. El Aassal, A.; Baki, S.; Das, A.; Verma, R.M. An In-Depth Benchmarking and Evaluation of Phishing Detection Research for Security Needs. *IEEE Access* **2020**, *8*, 22170–22192. Available online: https://ieeexplore.ieee.org/abstract/document/8970564 (accessed on 27 September 2021). [CrossRef]
49. Subasi, A.; Kremic, E. Comparison of Adaboost with MultiBoosting for Phishing Website Detection—ScienceDirect. *Procedia Comput. Sci.* **2020**, *168*, 272–278. Available online: https://www.sciencedirect.com/science/article/pii/S1877050920303902 (accessed on 27 September 2021). [CrossRef]
50. Mao, J.; Bian, J.; Tian, W.; Zhu, S.; Wei, T.; Li, A.; Liang, Z. Phishing Page Detection via Learning Classifiers from Page Layout Feature. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 43. Available online: https://jwcn-eurasipjournals.springeropen.com/articles/10.1186/s13638-019-1361-0 (accessed on 27 September 2021). [CrossRef]
51. A Novel Machine Learning Approach to Detect Phishing Websites. Available online: https://ieeexplore.ieee.org/abstract/document/8474040/ (accessed on 27 September 2021).
52. Chen, Y.H.; Chen, J.L. AI@ntiPhish—Machine Learning Mechanisms for Cyber-Phishing Attack. *IEICE Trans. Inf. Syst.* **2019**, *102*, 878–887. Available online: https://www.jstage.jst.go.jp/article/transinf/E102.D/5/E102.D_2018NTI0001/_article/-char/ja/ (accessed on 27 September 2021). [CrossRef]
53. Abdelhamid, N.; Thabtah, F.; Abdel-Jaber, H. Phishing Detection: A Recent Intelligent Machine Learning Comparison Based on Models Content and Features. In Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics, Beijing, China, 22–24 July 2017; Available online: https://ieeexplore.ieee.org/abstract/document/8004877 (accessed on 27 September 2021).
54. Jain, A.K.; Gupta, B.B. Towards Detection of Phishing Websites on Client-Side Using Machine Learning Based Approach. *Telecommun. Syst.* **2018**, *68*, 687–700. Available online: https://link.springer.com/article/10.1007/s11235-017-0414-0 (accessed on 27 September 2021). [CrossRef]
55. Lakshmi, L.; Reddy, M.P.; Santhaiah, C.; Reddy, U.J. Smart Phishing Detection in Web Pages Using Supervised Deep Learning Classification and Optimization Technique ADAM. *Wirel. Pers. Commun.* **2021**, *118*, 3549–3564. [CrossRef]
56. Sahingoz, O.K.; Buber, E.; Demir, O.; Diri, B. Machine Learning Based Phishing Detection from URLs—ScienceDirect. *Expert Syst. Appl.* **2019**, *117*, 345–357. Available online: https://www.sciencedirect.com/science/article/abs/pii/S0957417418306067 (accessed on 27 September 2021). [CrossRef]
57. Jagadeesan, S. URL Phishing Analysis Using Random Forest. *Int. J. Pure Appl. Math.* **2018**, *118*, 4159–4163.
58. Niranjan, A.; Haripriya, D.K.; Pooja, R.; Sarah, S.; Deepa Shenoy, P.; Venugopal, K.R. *EKRV: Ensemble of KNN and Random Committee Using Voting for Efficient Classification of Phishing*; Springer: Singapore, 2019; Available online: https://link.springer.com/chapter/10.1007/978-981-13-1708-8_37 (accessed on 27 September 2021).

59. Chiew, K.L.; Tan, C.L.; Wong, K.; Yong, K.S.; Tiong, W.K. A New Hybrid Ensemble Feature Selection Framework for Machine Learning-Based Phishing Detection System—ScienceDirect. *Inf. Sci.* **2019**, *484*, 153–166. Available online: https://www.sciencedirect.com/science/article/abs/pii/S0020025519300763 (accessed on 27 September 2021). [CrossRef]
60. Pandey, A.; Gill, N.; Sai Prasad Nadendla, K.; Thaseen, I.S. Identification of Phishing Attack in Websites Using Random Forest-SVM Hybrid Model. In Proceedings of the Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018), Vellore, India, 6–8 December 2018; Springer International Publishing: Midtown Manhattan, NY, USA, 2020. Available online: https://link.springer.com/chapter/10.1007/978-3-030-16660-1_12 (accessed on 27 September 2021).
61. Ali, W.; Ahmed, A.A. Hybrid Intelligent Phishing Website Prediction Using Deep Neural Networks with Genetic Algorithm-Based Feature Selection and Weighting. *IET Inf. Secur.* **2019**, *13*, 659–669. [CrossRef]
62. Aljofey, A.; Jiang, Q.; Qu, Q.; Huang, M.; Niyigena, J.P. An Effective Phishing Detection Model Based on Character Level Convolutional Neural Network from URL. *Electronics* **2020**, *9*, 1514. Available online: https://www.mdpi.com/2079-9292/9/9/1514 (accessed on 27 September 2021). [CrossRef]
63. Shie, E.W.S. Critical Analysis of Current Research Aimed at Improving Detection of Phishing Attacks. *Sel. Comput. Res. Pap.* **2020**, *45*, 45–53.
64. Maurya, S.; Jain, A. Deep Learning to Combat Phishing. *J. Stat. Manag. Syst.* **2020**, *23*, 945–957. [CrossRef]
65. Mao, J.; Bian, J.; Tian, W.; Zhu, S.; Wei, T.; Li, A.; Liang, Z. Detecting Phishing Websites via Aggregation Analysis of Page Layouts—ScienceDirect. *Procedia Comput.* **2018**, *129*, 224–230. Available online: https://www.sciencedirect.com/science/article/pii/S187705091830276X (accessed on 27 September 2021). [CrossRef]
66. Yang, L.; Zhang, J.; Wang, X.; Li, Z.; Li, Z.; He, Y. An Improved ELM-Based and Data Preprocessing Integrated Approach for Phishing Detection Considering Comprehensive Features—ScienceDirect. *Expert Syst. Appl.* **2021**, *165*, 113863. Available online: https://www.sciencedirect.com/science/article/abs/pii/S0957417420306734 (accessed on 27 September 2021). [CrossRef]
67. Anupam, S.; Kar, A.K. Phishing Website Detection Using Support Vector Machines and Nature-Inspired Optimization Algorithms. *Telecommun. Syst.* **2021**, *76*, 17–32. Available online: https://link.springer.com/article/10.1007/s11235-020-00739-w (accessed on 27 September 2021). [CrossRef]
68. UCI Machine Learning Repository: Phishing Websites Data Set. Available online: https://archive.ics.uci.edu/ml/datasets/phishing+websites (accessed on 29 November 2021).
69. Ramesh, G.; Krishnamurthi, I.; Kumar, K.S.S. An Efficacious Method for Detecting Phishing Webpages through Target Domain Identification. *Decis. Support Syst.* **2014**, *61*, 12–22. [CrossRef]
70. Singh, C. Phishing Website Detection Based on Machine Learning: A Survey. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE, Coimbatore, India, 6–7 March 2020; pp. 398–404.
71. Alsariera, Y.A.; Elijah, A.V.; Balogun, A.O. Phishing Website Detection: Forest by Penalizing Attributes Algorithm and Its Enhanced Variations. *Arab. J. Sci. Eng.* **2020**, *45*, 10459–10470. [CrossRef]