Phishing Website Detection from URLs Using Classical Machine Learning ANN Model

Said Salloum[1,2(✉)], Tarek Gaber[1,3], Sunil Vadera[1], and Khaled Shaalan[4]

[1] School of Science, Engineering, and Environment, University of Salford, Salford, UK
S.A.S.Salloum@edu.salford.ac.uk
[2] Machine Learning and NLP Research Group, University of Sharjah, Sharjah, UAE
[3] Faculty of Computers and Informatics, Suez Canal University, Ismailia 41522, Egypt
[4] Faculty of Engineering and IT, The British University in Dubai, Dubai, UAE

Abstract. Phishing is a serious form of online fraud made up of spoofed websites that attempt to gain users' sensitive information by tricking them into believing that they are visiting a legitimate site. Phishing attacks can be detected many ways, including a user's awareness of fraud protection, blacklisting websites, analyzing the suspected characteristics, or comparing them to recent attempts that followed similar patterns. The purpose of this paper is to create classification models using features extracted from websites to study and classify phishing websites. In order to train the system, we use two datasets consisting of 58,645 and 88,647 URLs labeled as "Phishing" or "Legitimate". A diverse range of machine learning models such as "XGBOOST, Support Vector Machine (SVM), Random Forest (RF), k-nearest neighbor (KNN), Artificial neural network (ANN), Logistic Regression (LR), Decision tree (DT), and Gaussian naïve Bayes (NB)" classifiers are evaluated. ANN provided the best performance with 97.63% accuracy for detecting phishing URLs in experiments. Such a study would be valuable to the scientific community, especially to researchers who work on phishing attack detection and prevention.

Keywords: Fraud protection · Cybersecurity · Machine learning · Phishing Detection · URL

1 Introduction

As the magnitude and sophistication of cybersecurity assaults grows, social engineering is regarded one of the most effective and simple strategies for obtaining inside and private data [1]. Phishing, as defined by the Anti-Phishing Working Group (APWG) [2] is a crime that involves the theft of financial account records and identity through the use of technical deception and social engineering. Additionally, delusive email messages and email addresses are utilized in the plan of social engineering where they hunt individuals who are heedless of this fraud and make them accept that they are connected to a legitimate and trusted group. They are created in such a way that guide users towards fake websites and eventually misleading them to disclose their username and password

J. Garcia-Alfaro et al. (Eds.): SecureComm 2021, LNICST 399, pp. 509–523, 2021.
https://doi.org/10.1007/978-3-030-90022-9_28

which is essential financial information. Whereas, technical subterfuge plans place hostile software in a computer system with the aim of documents theft by usually utilizing techniques that misleads users to fake websites or capturing their account username and password [3]. Moreover, a software like HTTrack is easily accessible to consumers allowing them to make an identical match of any website and use them for any reason. At the same time, in order to shield users from above mentioned attacks, the organizations must make people aware of the different ways to identify phishing emails or links. Therefore, this is the reason that qualified consumers fall into this trap by accessing hostile website supposing it to be a legitimate one, eventually disclosing their personal and sensitive information. Hence, it is seen that user awareness is essential together with computer-based solutions in order to gain a protection from phishing strike. Allowing a computer to have potentiality recognizing malicious websites, this solution would help users from staying away from them. A URL presents a universal address of a document in the World Wide Web and can be utilized to identify original sites in the event of fake sites made by pretenders. Hence, a common way to identify illicit phishing websites depends on their Uniform Resource Locators (URLs).

A blacklist of malicious URLs that are created by the anti-virus class can be one problem-solving perspective; however, this perspective is not considered as all-inclusive because of recent malicious URLs coming up frequently. Therefore, a perspective that can classify a recent, unknown URL into a phishing or legitimate website immediately is required. Usually, this perspective uses machine-learning, where a model created by the training groups of familiar attacks helps in classifying recent phishing sites.

Although the correct technology combined with security knowledge can protect a person from phishing attacks, implementing these in one's daily life might be difficult. Not to mention the ever-evolving new phishing attack patterns against which internet users and even existing email security solutions are powerless. Therefore, there seems to be an urgent need for new and improved methods of detecting phishing websites despite the availability of existing ones. Machine learning classifiers come into play here. Machine learning which is part of artificial intelligence (AI), employs a data mining approach to extract both known and unknown features from a data set, which are then combined with a classification algorithm to detect cyber-attacks and classify them as phishing. The objective of this paper is to develop a method to identify phishing attacks on a website. The study presents a phishing detection system to detect phishing attacks and harmful websites to accomplish this. To identify the attacks, the study described in this paper leverages features from website URLs and a Machine Learning technique. The potentials of generally utilized machine-learning algorithms on a similar phishing data set have been contrasted as the main aim in this study. To categorize URLs, we tried out general machine learning algorithms for instance "XGBOOST, Support Vector Machine (SVM), Random Forest (RF), k-nearest neighbor (KNN), Artificial neural network (ANN), Logistic Regression (LR), Decision tree (DT), and Gaussian naïve Bayes (NB)" classifiers. However, training data sets that include phishing URLs that exist in the public domain are very few. Therefore, it is an issue while creating a machine-learning-based perspective for the above-mentioned problem. Hence, the machine learning approaches, which depend on the available datasets, are required to be assessed for their efficacy. Furthermore, utilization of these datasets is performed in

this paper with components from the data URLs been pulled out and the availability of class labels.

This paper is organized as follows: the related work in classifying phishing URLs is described in Sect. 2, the specific aspects of data set and methodology is presented in Sect. 3, the test results along with discussion is presented in Sect. 4, and the limitations of the recent work and the future work is outlined in Sect. 5.

2 Literature Review

This section focuses on the relevant research work done by predecessors on phishing attacks as a whole and summarises the classification techniques which were used for the detection of web phishing.

2.1 Phishing Attack Approaches

One of the types of phishing is spoofing emails, where a phisher emails users a fake email address to deceive people so that they end up opening the email [4–8]. This allows the phisher to influence the user in gaining access to their private information [9]. Typically, a simple mail transfer protocol (SMTP) is used for email spoofing [10]. The phisher does these kinds of attacks using a spoofed email address that looks like a legal identity. To steal the data of well-known organizations is the purpose of such attacks [11]. The phisher makes fake accounts on known social media sites like Facebook, YouTube, Twitter, WhatsApp, Gmail, Instagram, and LinkedIn, where people share their profile of private information with complete disregard of privacy issues that may arise, this allows the phishers to send out requests to people while claiming to be a legitimate identity [9, 12]. The provision of a set of features that allow the correct URL classification is paramount because a URL plays a huge role on a website. The financial assets of an organization are under threat of a phisher's attack as well. Detection of phishing URLs and blacklisting are a few ways that can be used to reduce this problem [13]. A threat to system privacy is a Trojan horse that implements an action when the user clicks on a file. Most of the sites require people to enter their personal information, including job advertisements and illegal banking sites, which gain the interest of the users. Once a person enters his/her information on the Trojan horse, they became a fall prey to phishing [5, 14].

2.2 Classification of web Phishing detection Schemes

To this day, there have been many techniques developed to stop phishing attacks and guarantee the safety of users online. The detection of fake URLs and spoofed email is very difficult, and they are irrepressible. Blocking harmful emails and fake URLs are the best ways to stop phishing. An approach to detect and classify harmful URLs was proposed by [15], where the detection of malicious URLs is done by using a proactive approach in which the lexical analysis is used; also the proposal of a feature set is made for categorizing harmful URLs; additionally, the obfuscation technique is analyzed for the mitigation of these URLs [15]. PLIFER is a method based on ML technique introduced

by [16], which uses ten features that are extracted, the application of Random Forest (RF), and the age of domains of the URLs in the detection of a phishing website. This method's success is that it can classify phishing emails 96% of the time. Labelled data sets are used in classification techniques to detect phishing.URL and textual based features are used by various classification techniques. Features such as IP address, domain name, geographic properties as input in the classification of phishing are URL-based features.

The classification of such features is done through ML algorithms [17]. The use of hyperlinks in the pages of a website was proposed by [18]. Hyperlink features set as well as multiple ML approaches are used in tandem. Or the detection of phishing website's hyperlink features is one such approach. This approach is able to detect phishing with up to 98% preciseness as wee as being language-independent [19]. A model based on the ML algorithms which are used for the detection of phishing websites was proposed by [20] and is called the feature selection model hybrid ensemble feature selection (HEFS). A cumulative distribution gradient algorithm is applied for the extraction of the primary set. To get the second feature set a function called the data perturbation ensemble is used. An ensemble learner RF is used for detecting phishing websites. HEFS was able to detect phishing attributes with an accuracy of 94.6% as shown in the results [20]. A twofold technique that used the ensemble ML model for the classification of phishing websites was proposed by [21]. In the first step, RF classifier was applied and the results were integrated with feedforward NN. For the validation of the performance of the ensemble, K-fold cross-validation was used. For a publicly available data set [21] the results showed 93.41% accuracy with the use of the RF_NN model.

Li et al. [22] proposed a stacking model which uses URL features and HTML for the detection of phishing websites. The stacking model consists of the combination of Gradient boosted decision tree, light boosting machine (LightGBM), and XGradient-Boost. This approach is able to show 97.3% accuracy when applied to publicly available data sets [22]. A real-time anti-phishing system that can detect phishing URLs is given by [23]. This method used 7 classification algorithms [decision tree, K-star, AdaBoost, kNN (n = 3), SMO, RF, and Naïve Bayes] and various natural language processing-based features. NLP-based features have also been used in combination with RF. A data set made by the authors comprising of 73,575 URLs was used for the validation of this technique, and it showed 97.98% accuracy better than all previous models [23].

Dogukan et al. [24] suggested an anti-phishing system based on URL features named PHISH-SAFE to evaluate the performance of the proposed solution. This approach, which has its roots in machine learning, is adapted utilizing over 33,000 phishing and legitimate URLs, as well as SVM and Naive Bayes classifiers. Furthermore, the system uses 14 different URL elements to determine whether a website is phishing or legitimate. The use of an SVM classifier resulted in a 90% accuracy rate in detecting phishing websites.

Sahingoz et al. [23] suggested a real-time anti-phishing system with language independence, use of a large amount of phishing and legitimate data, real-time execution, detection of new websites, independence from third-party services, and use of feature-rich classifiers among its unique features. This system employs seven different categorization methods and features based on natural language processing (NLP). According to the experimental and comparison findings of the developed classification methods,

the Random Forest approach with solely NLP-based characteristics performs best, with a 97.98% accuracy rate for phishing URL detection.

Chriwstou et al. [25] attempted to construct a machine learning model that detects fraudulent URLs by training the SVM and Random Forests algorithms on malicious and benign datasets. This model was built to be used in conjunction with the Splunk platform and was influenced by previous approaches in the literature. They assessed the algorithms' performance in terms of precision and recall, with Random Forests achieving up to 85% precision and 87% recall and SVM achieving up to 90% precision and 88% recall using only descriptive features.

Gupta et al. [26] built a model for phishing detection with the maximum accuracy of 99.57% using the Random Forest algorithm. The experimental model employed the ISCXURL-2016 dataset and 11964 examples of legitimate and phishing URLs to detect phishing attempts with only 9 lexical features. This model performed well when tested against a variety of machine learning classifiers, however, it did have certain drawbacks. This model, for example, lacked advanced deep learning algorithms to evaluate its approach. Furthermore, 9 lexical features are insufficient for a model to be classified as successful.

In terms of the detection rate, the results of this review indicate that there is a need for up-to-date learning software such as machine learning as well as extra features to improve the accuracy of phishing detection systems. Machine learning models allow datasets to be used in the development of knowledge bases, requiring more time for training. Consequently, some parallel processing approaches will be extremely beneficial to these systems.

This research is very useful and focuses on the supervised ML approach, which can be used to detect phishing websites that use publicly available resources. There have been various approaches that show success in phish detection with high accuracy in blocking the attempts of malicious websites in luring people, with the affirmation of reports that the algorithm used in these approaches can detect and filter the phishing scams. However, the threat is still not over because the previous approaches ignored the text inside websites, which led to people being deceived. To fix this, a robust model based on machine learning algorithms has been proposed that can detect and block phishing schemes in websites.

2.3 URLs and Attackers' Techniques

A variety of techniques are used by attackers in order to avoid being detected by system admins or security systems. Figure 1 represents a general design of a URL. Moreover, the components of URLs should be noted in order to understand the attitude of attackers. To approach the web page, a URL begins with its protocol name in a typical form. Subsequently, situated is the subdomain and the Second Level Domain (SLD) name that usually represents the organization name in the server hosting. In the last, the Top-Level Domain (TLD) name that displays the domains in the DNS root zone of the Internet occurs.

The path of the page in the server represents the inside address whereas, the domain name (hostname) of the web page is found in the earlier parts and in the HTML structure

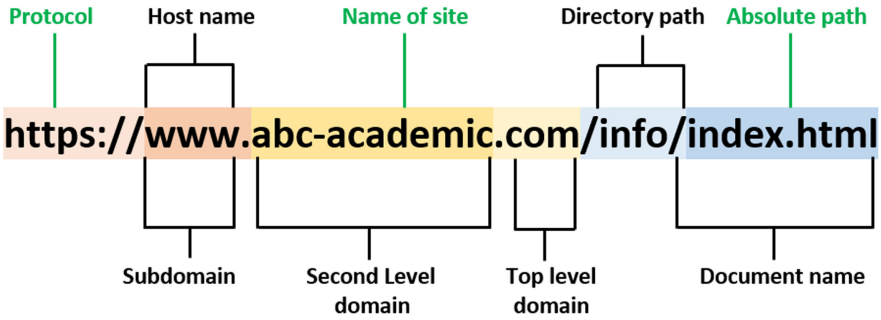

Fig. 1. The Parts of a URL

the page name is seen. Furthermore, the names utilized for phishing purposes, a considerable attempt is made by companies of cybersecurity to find out illicit domain via the name. This is because the constitution of SLD and TLD makes the distinctive and important part of a URL, called a domain name. However, the blocking of IP addresses can simply be done in order to halt access to web pages found in the domain name if it is recognized as phishing. Moreover, only at the initial time, the SLD name can be created, it is also seen that the attacker can locate or purchase it with ease for phishing purpose even though usually the company name and kind of activity is displayed on the SLD name. In addition, due to the reason that inside address structure relies on the attacker directly, by expanding the SLD via path and file names attackers can create limitless URLs. The detection system should consider the techniques of attack used by the attackers because there are certain essential ways adopted to weaken the users for instance cybersquatting, typosquatting, variable characters, and joint use of words. They do this to expand their attacking performance to steal additional sensitive information.

3 Methodology

This section details the proposed framework of phishing detection using URL features based on website properties. Feature selection techniques, selected data sets, ML algorithms, and performance evaluation measures are applied in experimenting with this proposal. The experimental setup for this suggested model is depicted in Fig. 2; there are multiple steps in the process. First, a dataset of phishing websites is selected; then a feature selection algorithm is used to analyze the top attributes. The features are fed into the ML classifiers after being normalized. The training of the features will be done using classical machine learning models, such as "XGBOOST, SVM, RF, KNN, ANN, LR, DT, and NB". Detection of phishing websites will be achieved by using best-performing algorithms (i.e., separating legitimate from phishing websites).

3.1 Dataset

Additional illustration and apprehension of phishing and legal actions can be obtained from the dataset. Hence, collecting a dataset is an important initial step. Sustaining the

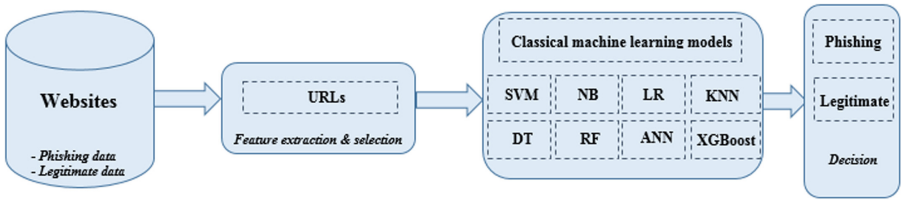

Fig. 2. The proposed approach

outcome validity, the dataset step is essential. For additional evaluation and to predict and anticipate the subsequent occurrence of phishing, the outcome obtained from the examined dataset is utilized. The entire features were gathered from [27]. Furthermore, Fig. 3 demonstrates the four group's value relies on the specific sub-strings; whereas, the first group is based on values of the features on the entire URL string [27]. The external services for instance Google search index and the URL resolve metrics are on which the last group attributes are based. After eliminating the target phishing attribute, there are 111 attributes in the dataset that indicate if the specific case is phishing (value 1) or legitimate (value 0). A total of 58,645 instances make up the dataset, of which 27,998 instances have been categorized as legitimate, whereas 30,647 instances have been categorized as phishing websites. [27]. The reason for these is to imitate the actuality of the original circumstances with the presence of additional legitimate websites. Since the 88,647 instances are present in the second variant of the dataset, where 58,000 instances are labeled as legitimate, and 30,647 instances labeled as phishing. Utilizing the website of Phishtank [28], at first, we gathered a list of 30,647 established phishing URLs for this preparing procedure. Secondly, 58,000 legitimate website URLs were collected from the list of legitimate URLs, attained from the Alexa ranking website [29]. Furthermore, a list of 27,998 community labeled along with organized URLs was attained as well [30], as these URLs are signifying impartially reported news they are legitimate in that aspect too. Moreover, the variants of the datasets mentioned above, we made those utilizing the URL lists of phishing and legitimate websites. All the instances from the *dataset_small* and more instances of extracted features by Alexa top sites list of URLs are found in the bigger and additional unstable dataset. While, the lesser, additional stable dataset *dataset_small* consists instances of features taken out from Phishtank URLs along with instances of extracted features via community labeled and organized URLs which are presented as to be legitimate [27].

Fig. 3. URL components [27].

3.2 Feature Extraction

The machine-learning algorithms and the features utilized are directly linked to the efficacy of the trained system. Hence, a detailed literature review is done in order to determine the important features. Furthermore, the research that utilizes features in various groups for example examining e-mail content and website analysis were studied along with the studies that just examine the URL. Moreover, within the hostname, path sections, and domain the attributes of the URL were individually assessed. The 111 various features found in our study were attained with Python programming language inscribed scripts.

Figure 4 shows the list of features utilized in our study as the top 85 features were selected following their classification from the RF Classifier, to attain a considerable accuracy rate.

URL	Domain	Directory	File name	Parameters	Format
qty_dot_url	qty_dot_domain	qty_dot_directory	qty_dot_file	qty_dot_params	Number of "." signs
qty_hyphen_url	qty_hyphen_domain	qty_hyphen_directory	qty_hyphen_file	qty_hyphen_params	Number of "-" signs
qty_underline_url	qty_underline_domain	qty_underline_directory	qty_underline_file	qty_underline_params	Number of "_" signs
qty_slash_url	qty_slash_domain	qty_slash_directory	qty_slash_file	qty_slash_params	Number of "/" signs
qty_questionmark_url	qty_questionmark_domain	qty_questionmark_directory	qty_questionmark_file	qty_questionmark_params	Number of "?" signs
qty_equal_url	qty_equal_domain	qty_equal_directory	qty_equal_file	qty_equal_ params	Number of "= " sings
qty_at_url	qty_at_domain	qty_at_directory	qty_at_file	qty_at_ params	Number of "@" signs
qty_and_url	qty_and_domain	qty_and_directory	qty_and_file	qty_and_ params	Number of "&" signs
qty_exclamation_url	qty_exclamation_domain	qty_exclamation_directory	qty_exclamation_file	qty_exclamation_ params	Number of "!" signs
qty_space_url	qty_space_domain	qty_space_directory	qty_space_file	qty_space_ params	Number of " " signs
qty_comma_url	qty_comma_domain	qty_comma_directory	qty_comma_file	qty_comma_ params	Number of "," signs
qty_plus_url	qty_plus_domain	qty_plus_directory	qty_plus_file	qty_plus_ params	Number of "+ " signs
qty_asterisk_url	qty_asterisk_domain	qty_asterisk_directory	qty_asterisk_file	qty_asterisk_ params	Number of "*" signs
qty_hashtag_url	qty_hashtag_domain	qty_hashtag_directory	qty_hashtag_file	qty_hashtag_ params	Number of "#" signs
qty_dollar_url	qty_dollar_domain	qty_dollar_directory	qty_dollar_file	qty_dollar_ params	Number of "$" signs
qty_percent_url	qty_percent_domain	qty_percent_directory	qty_percent_file	qty_percent_ params	Number of "%" signs
length_url	length_domain	length_ directory	length_file	length_ params	Number of characters

Fig. 4. Dataset attributes are based on URL, domain, file name, and URL parameters [27].

3.3 System Implementation and Performance Evaluation

Amongst the machine learning algorithms utilized in the experiment include: XGBOOST, SVM, RF, KNN, ANN, LR, DT, and NB. The models made with these algorithms trained via utilizing the Sklearn library within the Python programming language. The evaluation that depends on the distance of k neighbors is developed by the KNN algorithm which is quick and effective. Although, extensive memory is required in order to estimate the distance in huge data. While the accurate k value holds a great significance for the outcome. Additionally, when the dependent variable is gathered in two classes the algorithms that can generate productive predictions are called Logistic Regression. However, the adverse impact on prediction is seen by the factors for instance repeating of features, outlier values, and features inconsistency. Moreover, an algorithm that can operate with a huge group of independent variables and is simply executed is known as SVM. The drawbacks of this algorithm are that it does not work properly in

considerable noise and it is not appropriate for huge databases, although within non-linear issues it can provide efficient results by utilizing the basic trick. Furthermore, an algorithm that performs by splitting the dataset into sub-sections in order to set up a tree is known as a Decision Tree. Every node in this tree is allocated to a feature while every leaf is allocated to a class. Even though being simply illustrated is the plus point of this algorithm also contains less hyperparameter, but within smaller data sets and multi-class it does not work efficiently, and it is simply overfitted. An algorithm that places its pace and production at the forefront is known as XGBoost. It depends on gradient boosted decision trees. Although the plus point of this algorithm is its focus to lessen the errors present in the old tree, this is done by creating another tree. Yet, this procedure is time-consuming. Just like the earlier mentioned decision tree algorithm, this one can also be overfitted effortlessly. Additionally, a classification algorithm that performs as per Bayes' theorem and it depends on conditional probability is known as Gaussian Naive Bayes. The advantage of this algorithm is that it requires a shorter training time, and it has a simple application, but what makes it undesirable is the assumption that features are independent to all and its lesser estimation with low data. By producing the great number of trees in the dataset Random Forest is an algorithm that performs with the method of Ensemble Learning. The advantages of this algorithm are that it is slightly influenced by noise, does not need feature scaling, splits into subtrees, is strong, and resistant to detachment (overfit). Yet, it requires memory and processor power as it has been training for quite some time. Lastly, an algorithm having a formation such as biological neural networks and it performs with at the minimum of three layers is known as Artificial Neural Network (ANN), it has the ability to determine the link amongst the features and observe it properly as it utilizes less stat when training the model. However, it needs more processor power and memory which usually relies on the dimensions of the model along with its susceptibility to overfitting. In addition, a computer with Core™ i7-10750H 2.6 GHz processor having a memory of 16 GB Ram was utilized for the experiments. However, 10-Fold Cross-Validation was executed to attain the experimental outcome, in order to get the stats validated.

4 Experimental and Evaluation

For data splitting - training 70%, testing 30% - and algorithm evaluation, we have used the scikit learn library 12. The evaluation process for all the techniques includes a split of training 70%, test 30%, and 10-folds cross-validation. Furthermore, the entire experiments were performed on a Lenovo (LEGION 5 15IMH05H GAMING Core™ i7-10750H 2.6 GHz 1TB + 512 GB SSD 16GB 15.6" (1920x1080) 144 Hz BT WIN10). The loading of the entire dataset to Jupyter Notebook in the Anaconda environment is the initial step, afterward utilizing all URL features it is then classified using each of the 8 techniques. Subsequently, 8 of the methods having the highest accuracy were chosen and were assessed for their performance on Huddersfield phishing datasets, and the performance of every technique was evaluated with criteria of precision, accuracy, F-measure, and recall. The measurement of classifiers with respect to quality comprises employing Precision, Accuracy, Recall, and F1-measures. The confusion matrix given in Table 1 can be examined to understand these measures.

Table 1. Confusion matrix.

Confusion matrix	Predicted positive	Predicted negative
Actual positive	TP	FN
Actual negative	FP	TN

To evaluate the performance of our model, a confusion matrix is employed. A confusion matrix represents a table that presents an overview of the classification and segmentation performance. A two-class confusion matrix is used frequently to put forward the positive and negative classes for a few of the binary classification problems. It can be seen in Table 1 that the four cells of the matrix in this research are true positives (*TP*), false positives (*FP*), true negatives (*TN*), and false negatives (*FN*) [31]. *TP* refers to the total correct predictions which are positive, *FN* signifies the total incorrect predictions which are negative, *FP* refers to the total incorrect predictions which are positive, and *TN* signifies the total correct predictions which are negative. These four results can be used to obtain the four measures of classification performance.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F1-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{TN} + \text{FN})} \tag{4}$$

The traditional models used for comparison include "XGBOOST, SVM, RF, KNN, ANN, LR, DT, and NB". A succession of experiments has been performed using traditional machine learning methods to present equal parallel contrast between models. Models were selected based on their comparable and competitive outputs. The results have been recorded correctly without any bias in choosing models. The outcomes for the entire dataset utilized were attained in 8 various algorithms. The training time and precision, accuracy, F-measure, and recall rate are shown in Table 2. As per these data, the brief training time was seen with the NB algorithm whereas, utilizing the ANN classifier having a 97.63% of accuracy rate, the excessive test classification outcome was attained in the model.

Figure 5 illustrated the attained values of all the learning algorithms also the suitable execution time amongst the models tested are of "XGBOOST, SVM, RF, KNN, ANN, LR, DT, and NB" where the most suitable time was of the algorithm NB. The detection time of a web page by its URL is crucial in terms of utilizing the suggested system in runtime. Additionally, intending to avoid the third-party services available on the internet the execution time could be reduced. Moreover, the execution time of 100 URLs and 1 URL address does not have substantial variance because of the trained system formation. When

detecting Phishing URLs, the computers utilizing the algorithm NB or DT requires a shorter time as compared to using some others. However, with the addition of test time, only the algorithm SVM and RF show a difference in time as compared to the other algorithms. Furthermore, for the examination of 1 URL, it is desirable to utilize the higher accuracy rate algorithm at the time of modeling the system. Therefore, the NB algorithm will be suitable to utilize in the scenario. Also, the ANN algorithm can be utilized since the train time and high accuracy rate is seen when choosing an algorithm.

Table 2. Test results of classifiers on Dataset

Algorithm	Precision	Recall	F-Measure	Accuracy	Time (sec.)
ANN	0.969	0.975	0.971	**0.976**	43.1
DT	0.933	0.954	0.943	0.955	23.3
KNN	0.894	0.876	0.888	0.884	328.6
LR	0.882	0.896	0.890	0.892	39.4
NB	0.969	0.973	0.978	0.970	**10.2**
RF	0.934	0.962	0.962	0.968	658.2
SVM	0.925	0.925	0.962	0.855	989.7
XGBOOST	0.948	0.963	0.965	0.941	549.9

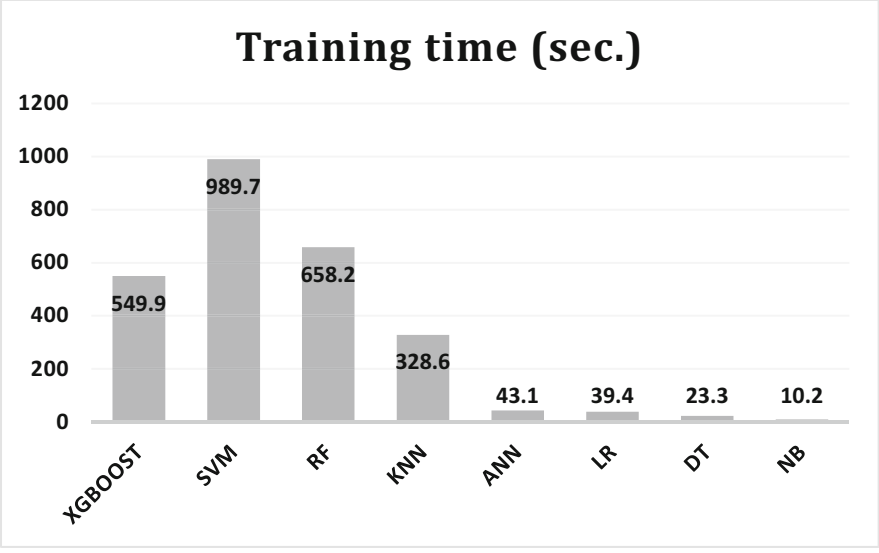

Fig. 5. Real-time execution of the classification algorithms

5 Comparison with Previous Work

The artificial neural networks' performance was highly encouraging, given that the modeled ANN was able to deliver a good conclusion despite the problem's great complexity. In the test phase, the ANN had a 97.6% accuracy rate. Table 3 shows a comparison of the ANN's accuracy with that of studies that used the ANN approach to detect phishing. Despite the good results obtained generally, Table 3 reveals that ANN's accuracy was among the best of the research analyzed. It is worth noting that an MLP was employed. In other words, in some circumstances, only a single strategy was used instead of two related procedures, indicating that the ANN-MLP is a good alternative for solving the problem [32]. It is also worth emphasizing that the comparison was done based on phishing detection accuracy, i.e., in the application of the problem rather than in the database because the bases employed in the two studies were different. In future experiments, the order of the qualities should be changed in order to find better groups for the ANNs to analyze. In future researches, the order of the qualities should be changed in order to find better groups for the ANNs to analyze. Its goal is to greatly expand the training and testing database in order to improve the ANN's generalization capability and, as a result, its performance in solving the classification issue.

Table 3. Comparative table

Study	Accuracy	Precision	Recall	F-Measure	FPR	TPR
[33]	–	0.966	0.966	0.966	–	–
[34]	0.933	0.933	0.933	0.933	7.0	–
[9]	0.958	0.967	0.958	0.960	–	–
[35]	0.889	–	–		–	–
[36]	0.955	–	–	–	–	–
[37]	0.969	0.969	0.969	0.969	–	–
[35]	0.849	–	–	–	15.91	85.61
[38]	0.920	–	–	–	–	–
[19]	0.972	0.974	–	0.975	–	–
[39]	0.972	–	–	–	–	–
[40]	0.955	0.952	0.961	0.957	–	
[32]	0.982	–	–	–	–	–
Our method	**0.976**	0.969	0.975	0.971	–	–

6 Conclusions and Future Works

The use of denounced phishers, lessons for beginners, and the development of methods of visualizing and integrating toolbars with the web browser have all been set in motion

to decrease phishing attacks. Even though the training of users on phishing costs a lot as well as the hopeful outcome of these procedures, yet phishing detection rates are low. Furthermore, machine-learning techniques have proved to be quite effective in defeating phishing. A strategy, employing ML and based on certain features, utilizes models in order to identify whether a website is legitimate or phishing. In this work, we execute a phishing detection system using certain machine learning algorithms. Further, the current datasets in the literature are used to evaluate the proposed system, and the results obtained are compared to the results of the most recent study. Our goal is to shortlist the best machine learning algorithm and detect phishing URLs, which is done by comparing the false negative, wrong positive, and accuracy rate of each algorithm.

As a result of this study, we investigated the strength of machine learning in spite of adverse learning techniques in terms of phishing detection. In addition, the study also looks at analyzing numerous features of both legitimate and phishing URLs through the technique of machine learning to detect phishing URLs. Machine learning is an auspicious method to differentiate between the websites that are legitimate or phishing. As seen that the purpose of phishing websites is to seize all the sensitive information of an individual for instance their credit card details, username and password and all various private data. All of this is accomplished by deceiving them into believing that these websites are legitimate. Although, machine learning can set out to downgrade the accuracy of a trained classifier model, as this method is liable to an adverse learning technique. In order to detect phishing websites, the following algorithms are utilized for instance; XGBOOST, Support Vector Machine (SVM), Random Forest (RF), k-nearest neighbor (KNN), Artificial neural network (ANN), Logistic Regression (LR), Decision tree (DT), and Gaussian naïve Bayes (NB)" algorithms. When evaluating our model by utilizing eight different machine learning algorithms, the ANN was found to produce the highest accuracy rate at 97.63%. In addition, for phishing website detection, experiments were repeatedly carried out using various (orthogonal and oblique) random forest classifiers. According to the contrasted outcome, it was found that the proposed system had high accuracy rates and increased phishing detection efficacy.

As a result, as far as future work is concerned, we should first develop a large and current dataset of Phishing Detection System-dependent URLs. Also utilizing certain hybrid algorithms along with NLP based features models stated in [23] we must utilize this dataset and aim to strengthen our system. The next step involves combining SVM with a web browser and employing large numbers of beginners in the pilot study. At last, we combine the proposed technique with different feature extraction models to evaluate its use in a real-world setting.

References

1. Salloum, S., Gaber, T., Vadera, S., Shaalan, K.: Phishing email detection using natural language processing techniques: a literature survey. Procedia Comput. Sci. **189**, 19–28 (2021)
2. Anti-Phishing Working Group. Phishing Activity Trends Report 1st Quarter 2020. https://docs.apwg.org/reports/apwg_trends_report_q3_2020.pdf.
3. Anti-Phishing Working Group. Phishing Activity Trends Report 3rd Quarter 2020 (2020). https://docs.apwg.org/reports/apwg_trends_report_q3_2020.pdf.

4. Gunawardena, S., Kulkarni, D., Gnanasekaraiyer, B.: A steganography-based framework to prevent active attacks during user authentication. In: 2013 8th International Conference on Computer Science & Education, pp. 383–388 (2013)
5. Gupta, S., Singhal, A., Kapoor, A.: A literature survey on social engineering attacks: phishing attack. In: 2016 international conference on computing, communication and automation (ICCCA), pp. 537–540 (2016)
6. Mujtaba, G., Shuib, L., Raj, R.G., Majeed, N., Al-Garadi, M.A.: Email classification research trends: review and open issues. IEEE Access **5**, 9044–9064 (2017)
7. Gualberto, E.S., De Sousa, R.T., Thiago, P.D.B., Da Costa, J.P.C.L., Duque, C.G.: From feature engineering and topics models to enhanced prediction rates in phishing detection. IEEE Access **8**, 76368–76385 (2020)
8. Sonowal, G., Kuppusamy, K.S.: PhiDMA–a phishing detection model with multi-filter approach. J. King Saud Univ. Inf. Sci. **32**(1), 99–112 (2020)
9. Zamir, A., et al.: Phishing web site detection using diverse machine learning algorithms. Electron. Libr. **38**, 65–80 (2020)
10. Salloum, S.A., Alshurideh, M., Elnagar, A., Shaalan, K.: Machine learning and deep learning techniques for cybersecurity: a review. In: Hassanien, A.-E., Azar, A.T., Gaber, T., Oliva, D., Tolba, F.M. (eds.) AICV 2020. AISC, vol. 1153, pp. 50–57. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-44289-7_5
11. Caputo, D.D., Pfleeger, S.L., Freeman, J.D., Johnson, M.E.: Going spear phishing: Exploring embedded training and awareness. IEEE Secur. Priv. **12**(1), 28–38 (2013)
12. Allen, J., Gomez, L., Green, M., Ricciardi, P., Sanabria, C., Kim, S.: Social network security issues: social engineering and phishing attacks. In: Proceedings Student-Faculty Research Day, CSIS, Pace University (2012)
13. Xiang, G., Hong, J., Rose, C.P., Cranor, L.: Cantina+ a feature-rich machine learning framework for detecting phishing web sites. ACM Trans. Inf. Syst. Secur. **14**(2), 1–28 (2011)
14. Wadhwa, A., Arora, N.: A review on cyber crime: major threats and solutions. Int. J. Adv. Res. Comput. Sci. **8**(5) (2017)
15. Mamun, M., Rathore, M., Lashkari, A., Stakhanova, N., Ghorbani, A.: Detecting malicious urls using lexical analysis. In: Chen, J., Piuri, V., Chunhua, S., Yung, M. (eds.) NSS 2016. LNCS, vol. 9955, pp. 467–482. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46298-1_30
16. Fette, I., Sadeh, N., Tomasic, A.: Learning to detect phishing emails. In: Proceedings of the 16th International Conference on World Wide Web, pp. 649–656 (2007)
17. Das Bhattacharjee, S., Talukder, A., Al-Shaer, E., Doshi, P.: Prioritized active learning for malicious url detection using weighted text-based features. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 107–112 (2017)
18. Kumar, N., Chaudhary, P.: Mobile phishing detection using naive Bayesian algorithm. Int. J. Comput. Sci. Netw. Secur. **17**(7), 142–147 (2017)
19. Jain, A.K., Gupta, B.B.: A machine learning based approach for phishing detection using hyperlinks information. J. Ambient. Intell. Humaniz. Comput. **10**(5), 2015–2028 (2018). https://doi.org/10.1007/s12652-018-0798-z
20. Chiew, K.L., Tan, C.L., Wong, K., Yong, K.S.C., Tiong, W.K.: A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. Inf. Sci. (Ny) **484**, 153–166 (2019)
21. Nagaraj, K., Bhattacharjee, B., Sridhar, A., Sharvani, G.: Detection of phishing websites using a novel twofold ensemble model. J. Syst. Inf. Technol. **20**, 321–357 (2018)
22. Li, Y., Yang, Z., Chen, X., Yuan, H., Liu, W.: A stacking model using URL and HTML features for phishing webpage detection. Futur. Gener. Comput. Syst. **94**, 27–39 (2019)
23. Sahingoz, O.K., Buber, E., Demir, O., Diri, B.: Machine learning based phishing detection from URLs. Expert Syst. Appl. **117**, 345–357 (2019)

24. Jain, A., Gupta, B.B.: PHISH-SAFE: URL features-based phishing detection system using machine learning. In: Bokhari, M., Agrawal, N., Saini, D. (eds.) Cyber Security. AISC, vol. 729, pp. 467–474. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-8536-9_44
25. Christou, O., Pitropakis, N., Papadopoulos, P., McKeown, S., Buchanan, W.J.: Phishing url detection through top-level domain analysis: a descriptive approach (2020). arXiv Prepr. arXiv2005.06599
26. Gupta, B.B., Yadav, K., Razzak, I., Psannis, K., Castiglione, A., Chang, X.: A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. Comput. Commun. **175**, 47–57 (2021)
27. Vrbančič, G., Fister, I., Jr., Podgorelec, V.: Datasets for phishing websites detection. Data Br. **33**, 10643 (2020)
28. http://www.phishtank.com
29. https://www.alexa.com
30. Lab, O.C.: Url testing lists intended for discovering website. In: Censorship (2014)
31. Sammut, Claude, Webb, Geoffrey I. (eds.): Encyclopedia of Machine Learning and Data Mining. Springer, Boston (2017). https://doi.org/10.1007/978-1-4899-7687-1
32. Ferreira, R.P., et al.: Artificial neural network for websites classification with phishing characteristics. Soc. Netw. **7**(02), 97 (2018)
33. Sameen, M., Han, K., Hwang, S.O.: PhishHaven—an efficient real-time ai phishing URLs detection system. IEEE Access **8**, 83425–83443 (2020)
34. Zaini, N.S., et al.: Phishing detection system using machine learning classifiers. Indones. J. Electr. Eng. Comput. Sci **17**(3), 1165–1171 (2019)
35. Korkmaz, M., Sahingoz, O.K., Diri, B.: Detection of phishing websites by using machine learning-based URL analysis. In: 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–7 (2020)
36. Pradeepthi, K.V., Kannan, A..: Performance study of classification techniques for phishing URL detection. In: 2014 Sixth International Conference on Advanced Computing (ICoAC), pp. 135–139 (2014)
37. Osho, O., Oluyomi, A., Misra, S., Ahuja, R., Damasevicius, R., Maskeliunas, R.: Comparative evaluation of techniques for detection of phishing URLs. In: Florez, H., Leon, M., Diaz-Nafria, J.M., Belli, S. (eds.) ICAI 2019. CCIS, vol. 1051, pp. 385–394. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32475-9_28
38. Sahingoz, O.K., Baykal, S.I., Bulut, D.: Phishing detection from urls by using neural networks. Comput. Sci. Inf. Technol. **8**(17), 41–54 (2018)
39. Sindhu, S., Patil, S.P., Sreevalsan, A., Rahman, F., An, M.S.: Phishing detection using random forest, SVM and neural network with backpropagation. In: 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), pp. 391–394 (2020)
40. Zhang, N., Yuan, Y.: Phishing detection using neural network. In: CS229 Lect. notes (2012)