International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

September 2023 | Issue 56 | Volume 15 | Number 3 | Pages 126-134

PHISHING IDENTIFICATION THROUGH UP-TO-DATE FEATURES GENERATION AND EXPLORATION

L.H. Abed [1] H.J. Mohammed [2] Y.S. Yaseen [2]

1. Department of Computer Systems Techniques, Anbar Technical Institute, Middle Technical University, Baghdad, Iraq laithhamed@mtu.edu.iq
2. Computer Center, University of Anbar, Anbar, Iraq, hussamjasim@uoanbar.edu.iq, yaseen.yaseen@uoanbar.edu.iq

Abstract- With the growing developments of digital technologies, advanced web applications have offered a convenient way to undertake daily vital services for individuals and enterprises. Nevertheless, cyberattacks have immensely led to privacy breaches via fraudulence, the most damaging being phishing attacks. Although blacklisting is deemed inadequate for overcoming such attacks, classification techniques have manifested their competence in beating them. However, the fresh phishing URLs in particular the ones appearing trickily genuine can undermine the detection accuracy. Furthermore, given the attempts of increasingly deceiving users, the ineffectiveness will exacerbate as phishing patterns would be variant throughout. Feature generation and exploration play an indispensable part if not essential pillar of identification to tackle such vulnerabilities. This research, therefore, presents an advanced anti-phishing framework using superior URLs features to defeat potential phishing attacks. With a view to reflecting a real-life scenario, an expanded dataset, including benign and phishing URLs is designed across a range of ongoing data sources. Accordingly, up-to-date proactive URLs features are generated and explored using a feature selection approach to determine the extent to which such features can be effective. Experiments are consequently implemented using the merits of various machine learning models to determine the reliability of detecting phishing websites. The results and analysis of experiments showed that the proposed anti-phishing approach was promisingly credible in detecting newly published phishing websites with an accuracy rate of 88.3%.

Keywords: Phishing Attacks, URLs Features, Naïve Bayes, AdaBoost Classification, Bagged Trees Classifier.

1. INTRODUCTION

Given The quick advancements in digital technologies have posed many Internet-based applications which have transformed the perception of doing many conventional daily routines into cyberspace [1]. Cyber-criminals, consequently, exploit this transformation in deceiving legitimate people and organizations to leak sensitive private/secret information. As a result, cyberattacks have immensely led to privacy breaches via fraudulence [2], the most damaging being phishing attacks [3]. Based upon the survey of Cisco Umbrella (i.e., the industry leader in threat detection) regarding cybersecurity threats, one user at least pressed a cyber-phishing link within nearly 86% of enterprises, and the available figures indicate that approximately 90% of privacy leaks represent phishing attacks [4]. Overall, cyber phishing attacks aim to obtain the user's information, such as their login details and credit card information. It takes place when a phisher masquerades as a trustworthy party and deceives a genuine person by clicking a malicious link through which malware is installed, or sensitive information is leaked [5].

With the purpose of coping with such attacks, a number of techniques are presented for resisting and beating them, including blacklisting, dynamic, and static techniques [6, 7]. Although blacklisting methods prevent users from accessing the priorly detected phishing websites using particular attributes, including domain names, IP addresses, and URLs, they still have no capacity for blocking zero-day/new phishing websites [7]. Dynamic methods inspect harmful scripts designed with webpages as well as analyze abnormal behaviors in web applications for phishing identification. While these methods are considered very effective versus phishing, they consume huge resources and could not probably act as a frontline of protection [6].

Static methods, on the other hand, exploit machine learning algorithms to identify phishing websites by training observed samples of phishing and benign and accordingly predict them in the meantime; therefore, they are considered robust proactive means of protection [6, 8]. Whilst phishing identification using machine learning techniques has demonstrated its competence in defeating phishing attacks, the newly deceived phishing URLs in particular the ones appearing genuine would undermine the system accuracy. In addition, due to the attempts of increasingly tricking legitimate people and organizations, the ineffectiveness will exacerbate as phishing patterns would be inconsistent over time. Feature generation and exploration play an indispensable part [9] if not essential pillar of identification to tackle such vulnerabilities.

What is more, some of the existing phishing identification methods have been implemented using limited and old-fashioned URL datasets. Other methods also overwhelmingly utilized very common machine learning algorithms. Furthermore, whilst many phishing detection techniques have concentrated on detecting specific types of phishing websites, little attention has been paid to detecting newly deceived phishing URLs. On this basis, the contribution of this research lies in collecting large amounts of zero-day phishing data. and accordingly, using a credible mechanism to generate up-to-date URL features and investigate their robustness in detecting fresh phishing attacks. This would require an intelligent technique under which the effectiveness of a URL feature can be determined and prioritized. As such, this research presents an advanced anti-phishing framework through the experimentation of considerable newly collected URL samples using superior textual features and a well-considered diversity of machine learning techniques.

The rest of this article sets out some of the real-world use cases and applications of machine learning and introduces the review of the previous works within sections two and three. The proposed framework of phishing detection is discussed in section four. Thereafter, the materials and methods for implementing the methodological approach are presented and explained within section five. Then, section six describes and analyses the empirical results, followed by a comparison with previous techniques, conclusions, and future work in sections seven and eight.

2. MACHINE LEARNING APPLICATIONS AND REAL-WORLD USE CASES

Machine learning has immensely evolved forward in improving different day-to-day industries, with intelligent and superior capacities being afforded using sophisticated algorithms enabling computing systems to train from previously identified data and produce predictions for new contexts. Thus, various everyday machine learning applications and real-world use cases have widely spread everywhere ranging from phishing, financial fraudulence, image recognition, self-driving cars to service and product recommendation. These applications and real-world use cases are explained in following subsections [3], [6], [10]

2.1. Phishing

As discussed earlier, classical phishing identification methods are not powerful enough to identify genuine and harmful websites. Advanced machine learning models can consequently detect patterns that unveil malicious URLs. In order to accomplish this, the classification algorithms can be learned using textual features to recognize the harmful ones in a realistic environment [1], [10].

2.2. Financial Fraudulence

Fraud identification is one of the most vital applications of machine learning. Smart and advanced machine learning algorithms cautiously check the authentic users' accounts each and every time a financial transaction is completed to catch any suspicious action in the meantime [10].

2.3. Image Recognition

Digital image recognition is a very remarkable machine-learning real-world application. In this approach, digital image features are generated and recognized using classification methods. This approach is widely used in biometric recognition [2], [9-10].

2.4. Self-Driving Cars

Another type of machine learning real-world use case is the evolutionary use of self-driving cars which massively depend on machine learning merits. Within these applications, smart cars gather data from different data sources, such as sensors and digital cameras to be input into very novel and intelligent machine-learning models and accordingly the latter would train, and adjust to manage the vehicle actions [10].

2.5. Service and Product Recommendation

Innovative machine-learning techniques are massively exploited by very famous technology firms, such as Google, Amazon, and Netflix for recommending the use of many applications, services, and products to clients. Advanced machine-learning approaches study the inquiry of the client, compare it to the numerous corresponding patterns, and then match it with the very close ones to suggest what she/he can order or watch in the meantime. [10].

3. PREVIOUS WORKS

A large number of studies have introduced various solutions for defeating cyber phishing attacks including blacklisting, dynamic, and static detection approaches. As mentioned earlier, blacklisting cannot block newly tricked phishing URLs, and dynamic detection techniques consume considerable resources as well as can only act as reactive security measures [6, 7]. Since this article concentrates upon accomplishing a powerful proactive frontline of protection using low power consumption resources, the review of the previous studies has been narrowed down to static phishing detection approaches. Amongst these, Basnet, et al. [11] employed 24 lexical URL features using the most effective techniques of machine learning.

However, the random forest technique revealed the superiority in identifying phishing attacks. Despite this, the authors experimented limited URLs data which could not be broad and enough for developing a robust framework of protection against phishers attempts. In other contribution, Mamun, et al. [12] generated 79 textual URL features to identify different forms of website malicious attacks, the phishing attacks being one of the well-studied ones. In this study, the mechanisms of both information gain and correlation were employed to select the important features. A number of the machine learning approaches including k-nearest neighbor algorithm, random forest algorithm and decision tree algorithm were consequently implemented for malicious website (e.g., phishing) detection. The experimental results revealed that the accuracy of the presented framework was all in all 99%.

Alshira'h and Al Fawa'reh [13] also created textual URL attributes from considerable amounts of data for defeating phishing websites. The contributors applied seven machine learning algorithms (i.e., support vector machine, Gaussian Naive Bayes, k-nearest neighbor, quadratic discriminant analysis, perceptron, random forest, and decision tree) to figure out their performance in phishing detection.

Subsequently, owing to the presence of unequal instances of phishing and non-phishing samples, the approach of synthetic minority oversampling was adopted to duplicate the minority class for a fair evaluation. With 98% accuracy, the random forest algorithm accomplished a better performance against the other machine learning techniques used in this study. In a different review, Kumar, et al. [14] analyzed how an approach could differentiate between the URLs of phishing and legitimate with regard to individuality, differences, and uncommon entities. On this basis, a number of lexical and statistical URL features were created. The authors also devised a manner of eliminating specific keywords from the URL with the aim of facilitating the course of identification. After which, a variety of machine learning techniques were performed in order to figure out the superior technique for detecting phishing websites. The empirical results showed that there was nearly no difference in the performance of the applied classification models. On the other hand, this study did not utilize the entire set of the created URL features which can give rise to more accurate results. What is more, there were a number of outdated URLs included in the selected dataset, and this could weaken the effectiveness of detection.

With regard to the manners of feature selection, Zaini, et al. [15] employed the swarm optimization technique for feature selection in order to pick 15 features from the generated 30 URL features. The researchers, thereafter, applied various machine-learning approaches to investigate their effectiveness for detecting phishing attacks. The experimental results confirmed that the random forest achieved the best performance. In the same context, Gupta, et al. [16] claimed that using many URL features would need loads of resources for reliably identifying phishing websites. Therefore, the writers developed a feature selection manner in which only nine features were adopted within the detection system. Accordingly, experiments were conducted using the dataset of ISCXURL-2016 and applying various techniques of machine learning algorithms. The random forest technique accomplished the greatest performance (i.e., 99.57%).

Likewise, Gandotra and Gupta [17] demonstrated that using numerous URL features needs time for developing a classification model and accordingly can impact the performance of phishing detection on a timely basis. This was demonstrated by evaluating the performance of some machine learning techniques across the entire URL features and the selective URL features, which were selected using a feature selection approach, respectively.

The results showed that the random forest algorithm outperformed the other classification models through both the entire and the selective URL features, although the latter was considered more efficient. In a different study, Abutaha, et al. [18] derived 22 textual features from numerous URLs to identify phishing attacks using various machine learning approaches. The results reported that SVM algorithm outperformed the other approaches with a 99.89% accuracy figure. From a different perspective, Purbay and Kumar [19] determined the impact of various splitting data approaches upon the accuracy of the phishing detection system using four machine learning methods. In order to explore this hypothesis, the researchers generated additional attributes, such as page rank, traffic rank information in addition to the textual URL features. Another study was presented by Dutta [20] who argued that traditional machine learning techniques can recognize restricted volumes of URLs in real-life, and accordingly there was a need to explore a sophisticated intelligent approach to cope with this issue. Having done an extensive search, the author come up with a recurrent neural network model called long short-term memory to combat phishing attacks occurred in cyberspace. This model was developed using 13700 of phishing and genuine URL samples. The results demonstrated that the applied intelligent approach of proposed model outdid traditional methods of phishing identification.

It is obvious from the previous discussion and analysis that there is yet a number of limitations within the existing phishing detection approaches that should be overcome. Whilst some authors generally tested their proposed anti-phishing techniques through outdated phishing URLs, others implemented their experimental designs using limited datasets. There are also specific machine learning techniques that were overly applied in the literature. Therefore, there is a need to present an advanced anti-phishing framework that concentrates upon exploring huge newly tricked phishing patterns using a well-studied variety of machine learning models.

4. THE PRESENTED FRAMEWORK OF PHISHING IDENTIFICATION

Given the incidence of recently damaging phishing breaches, an advanced, proactive, and reliable anti-phishing framework has been schemed and developed as described in Figure 1. With the phishing identification approach being introduced in this research to defeat zero-day cyberattacks, there is a number of key requirements that have to be accomplished. One of these requirements is that newly tricked phishing data has to be highly pragmatically collected with a view to having a real-life and fair evaluation. In particular, URL acquisition is undertaken to gather realistic row data, including zero-day phishing URLs and legitimate URLs. This is the core foundation under which the presented framework reliant on pattern classification is set out for phishing detection. That is, the captured URL strings are transformed and utilized as inputs for the identification process. Collecting and organizing considerable amounts of URL strings are also sought with the aim of really truly reflecting a practical and functional scenario that should be close to reality as much as possible.

In addition to this, there is a need to create up-to-date URL features and subsequently determine their superiority in order to facilitate the process of detecting fresh damaging attacks. On the whole, the feature creation procedure is used to carry out specific calculations upon the raw URL data collected beforehand to produce distinctive values (i.e., attributes) through which a phishing or benign URL would be recognized. For the time being, as phishing patterns do seem to be highly similar to genuine ones, the generation of up-to-date URL features is demandingly sought within the proposed anti-phishing framework to escalate the performance of phishing detection. Expanding upon this, effective URL features are pinpointed via modelling and conducting a number of tests upon these generated URL features to figure out those which would have an important role or vital contribution in aiding the process of phishing identification. With the purpose of determining this, the community of machine learning has a number of tactical and intelligent means (e.g., extra trees classification technique) which can possibly drill into the details of data to reveal whether the URL features are significant or not.

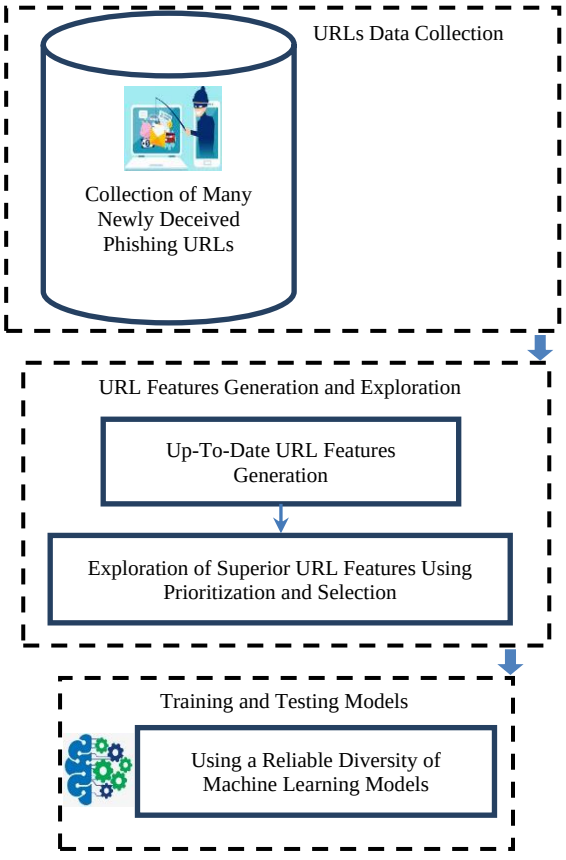

Figure 1. The proposed framework of phishing identification

What is more, a selection of diverse machine learning techniques has to be established with a view to applying the potential ones that can be robust against newly published phishing URLs. Of course, the machine learning pipeline is growingly developing every day in a very quick manner. In essence, the machine learning pipeline aims to computerize the learning workflow by transforming the series of data and correlating them together within a model for establishing predictions. In addition, various machine learning pipelines concentrate upon strategic aspects within their operation to raise up their effectiveness.

Accordingly, a robust variety of machine learning techniques are picked in order to find out their effectiveness in catching newly published phishing attacks. Thus, the presented anti-phishing framework is developed and implemented depending upon the prior insights with the aim of combating the very recent phishing URLs.

5. MATERIALS AND METHODS

A number of materials and methods were used for generating up-to-date features, investigating their effectiveness, and unveiling the performance of classification in detecting zero-day phishing attacks. These materials and methods are thoroughly illustrated by the following subsections:

5.1. Data Collection and Organization

With the aim of fully investigating the key hypothesis of this research, ongoing data collection sources were practically required to capture and organize a satisfactory expanded dataset of benign and fresh-phishing URLs for realistic experimentation purposes. As a consequence, the web-based platform of PhishTank in which considerable zero-day phishing URLs can be collected in an ongoing fashion was utilized for illegitimate URLs collection [21]. In particular, 6325 phishing samples are captured through the PhishTank technology. Samples within PhishTank are freely obtainable for members where an enrolment for a license key was undertaken to finalize the process of data acquisition.

On the other hand, it is believed that collecting new legitimate URLs was not a very demanding need in the context of this contribution. This is because phishers always almost impersonate their URLs as legitimate ones with the aim of deceiving victims. Furthermore, the authentic popular websites arguably would not tend to change their URLs because of fraudulent impersonation. For instance, the URLs of Google and PayPal have been forged by phishers and once a victim is pressed on those URLs, he/she would be directed to an illegitimate website to be scammed. However, those original URLs were not changed because of this scam. On this basis, selecting the legitimate URLs was done from Alexa top websites which were gathered by Mamun, et al., 2016 [12] where a million of the most popular websites around the world can be archived. As Mamun, et al. [12] stated, more than 35310 benign URL samples were reliably assembled dependent upon their genuineness.

As such, both benign and phishing samples were unified to become 6000 samples each and then combined all together (i.e., the whole dataset included 12000 samples) in order to apply fair balanced classification later on. Having organized the dataset in line with the context of this research, discriminative features were needed as inputs into a machine learning model. These URL features are explained within the next subsection. Table 1 depicts a preview on the nature of both the phishing [18] and benign [9] datasets.

Table 1. Preview on the nature of the data collection and organization. (a) phishing dataset [18], and (b) legitimate dataset [9]

(a)

URL	URL Details	Verified	Online	Target
https://yahooaccountsupp	www.phishtank	yes	yes	Other
https://etqwewqreeq.wee	www.phishtank	yes	yes	Other
https://ffmemberrshipvn-	www.phishtank	yes	yes	Other
https://bper.zaparetech.co	www.phishtank	yes	yes	Other
https://allegrolokalnie-	www.phishtank	yes	yes	Allegro
https://inpost.id7436457.	www.phishtank	yes	yes	Other
https://e-wizink-acesso-	www.phishtank	yes	yes	eBay
https://pl-olx.879456.site	www.phishtank	yes	yes	Other
https://leboncoinenligne.s	www.phishtank	yes	yes	Other
https://cw49455.tmweb.	www.phishtank	yes	yes	Other
https://leboncoin-achats	www.phishtank	yes	yes	Other
https://chpost.tempurl.h	www.phishtank	yes	yes	Other
https://allegrolokalnie-pl	www.phishtank	yes	yes	Allegro
http://steampoweredtrade	www.phishtank	yes	yes	Steam
http://kensingtonmarath	www.phishtank	yes	yes	Optus

(b)

ID	Genuine URLs
1	http://1337x.to/torrent/1048648/American-Sniper-2014-MD-i
2	http://1337x.to/torrent/1110018/Blackhat-2015-RUSSIAN-720p
3	http://1337x.to/torrent/1122940/Blackhat-2015-x264-1080p-
4	http://1337x.to/torrent/1124395/Fast-and-Furious-7-2015-HD-T
5	http://1337x.to/torrent/1145504/Avengers-Age-of-Ultron-2015-
6	http://1337x.to/torrent/1160078/Avengers-age-of-Ultron-2015-H
7	http://1337x.to/torrent/294349/American-Idol-S11E04-Audition
8	http://189.cn/dqmh/userCenter/myOrderInfoList.do?method=list
9	http://2gis.ru/moscow/search/%D0%9F%D0%BE%D0%B5%D1
10	http://abc.go.com/shows/general-hospital/episode-guide/2015-05
11	http://abc.go.com/shows/the-muppets/video/new-abc-comedy-tr
12	http://abcnews.go.com/US/wireStory/regulators-delays-georgia-n

5.2. Up-To-Date URL Features Generation

Feature generation plays a vital part if not essential pole in detecting malicious websites. From a methodological standpoint, this study employed the lexical or textual ULR features because they can act as a proactive frontline of protection prior to occurring any cyber breaches. In particular, creating and exploring up-to-date features was sought in this investigation in order to take advantage of their robustness in defeating phishing attacks. A number of recent research papers were consulted for this objective with a view to taking textual attributes out of the collected URL strings [6, 8, 12, 22, 23]. The extracted features mainly represented the number of characters within the whole URL, domain, directory, file, file extension, path, and query (i.e., the length of each section) [22, 23]. Other URL features were the number of special characters, digits, and letters and the number of each special character (e.g., "+", "?", "/", "@", "#", "_" and so forth) within the complete URL itself and its components previously aforementioned. This was methodologically done because an adversarial URL would highly likely carry a greater number of characters than its opposite URL (i.e., the benign one) [23].

For further URL features generation, Christou, et al. [8] claimed that specific symbols of a URL would be normally replaced by phishers for legitimizing them. For instance, the letter o of the google popular domain name can be faked by replacing it with the digit zero in order to victimize users. To this end, had the entropy of both phishing and benign URLs calculated, they will be arguably different. As a result, the entropy of the whole URL and its auxiliary sections was calculated as URL features [8, 12]. In addition to this, given the fact that shorter domains are costly, and their owners would buy them for underpinning the usability requirement, the attribute of the symbol continuity average was taken as a URL feature. Costly domains cannot be afforded by phishers; thereby, they would fake longer domains.

Exploiting such a feature could possibly aid in identifying benign and zero-day phishing URLs. In order to determine the symbol continuity average, tokens of consecutive special characters, digits, and letters were divided from domain names as they could. Then, the length of each token was calculated and the longest one from each respective class was chosen. The entire length of all the selected classes was subsequently found and then divided by the length of the entire domain [22]. For example, the symbol continuity average of a domain name, such as xwzy1346@#?201ab) is calculated as (4+4+3)/17=8.176. Boolean features were also created to check whether the internet protocol within a URL is HTTP or HTTPS under which a URL could be benign if a layer of transmission security was exploited; otherwise, it might be phishing [6].

5.3. Superior URL Features Exploration

Having created all the former 150 URL features, experiments were designed and carried out to discover the effective ones in order to escalate the accuracy of detecting new phishing attacks. With the purpose of accomplishing this, a machine learning model called extra trees was exploited to prioritize the generated URL features reliant upon the feature importance property being built-in within python platform programming. This technique can reorder the features according to their significance in predicting the target. As such, the model was set and fitted by its determined parameters (i.e., the whole group of 150 features and the target including phishing/benign), aiming ultimately at giving a score for each feature.

The higher the score, the more significant the feature towards identifying the target. Following this, a feature selection procedure was applied with the aim of having a tangible foundation regarding the effective or superior URL features. This was established by checking the effectiveness of the earlier model of extra tree in recognizing the target using different groups of ranked features. This classifier was built upon 50% training data and 50% testing data of the entire dataset using its default parameters. It was also initiated by the highest 30 ranked features and the target in the first test; and accordingly, the feature group was increased in a progressive way by 30 features within each subsequent experiment. The group of ranked features with the highest accuracy was considered the most effective one.

5.4. URL Features Standardization

URL features standardization is a pre-processing procedure that was used to reduce the data variances. This principle is meant to convert numeric features into a uniform range; thereby, the complications occurring during the workflow of the machine learning model are

diminished. Using such a technique had a very important part for an efficient machine learning process. That is, the biased classification to the features of large numeric values was eliminated. In addition to this, the speed of learning and producing estimators was also expedited. In seeking the available techniques for sorting this task out, the tool of scikit-learn within the platform of python programming provides a built-in scalar (i.e., StandardScaler) for data standardization. Therefore, this transformer was used to accomplish the purpose of this stage.

5.5. Machine Learning Models

With the merits of the machine learning techniques being developed and modeled, further experiments were implemented using a well-considered spectrum of statistical and ensemble models, including Naive Bayes (NB), AdaBoost (AB), Bagged Trees (BT) to explore their effectiveness in detecting phishing URLs.

It is believed that applying such a diverse combination of models would give a closer insight into the performance of the presented anti-phishing framework. The NB has the capacity to effectively classify considerable data and identifies the target through the probability of each URL feature. The ADB contrarily develops multiple decision tree models across a number of layers to adjust the errors from each one and reduce them accordingly until the target is correctly recognized. On the other hand, the BT combines the accuracies of multiple decision tree models to produce a more applicable accuracy. As such, a 50/50 sample splitting ratio was conducted to divide the dataset into two sets: the former for learning the classifier and the latter for testing its performance - thus conveying an acceptable real-life scenario of evaluation.

Based upon the dataset being collected and designed, the learning set consisted of 3000 phishing URLs and 3000 benign URLs and the testing set contained exactly the same. At the end of this phase, samples were randomly organized, and then each machine learning model selected within this investigation was configured using its own default parameters.

5.6. Performance Evaluation

This stage calculated the effectiveness of the machine learning model in identifying phishing websites through measuring the correct and wrong predictions using specific metrics. On the types of evolution front, three metrics including True Positive Rate (TPR), precision, and accuracy were measured to assess the performance of phishing detection. TPR (i.e., sensitivity or recall) measures the number of phishing URLs detected as phishing divided to the all-phishing URLs. The precision represents the ratio of the total number of correctly classified phishing URLs and the total number of predicted phishing URLs, where it explains the correctness achieved in phishing prediction. The accuracy is evaluated by the division of the total number of correct predictions to the whole population. The metrics of recall, precision, and accuracy can be calculated by the following Equations (1-3) [22]:

$$Recall = TP / (TP + FN) \quad (1)$$

$$Precision = TP / (TP + FP) \quad (2)$$

$$Accuarcy = (TP + TN) / (TP + TN + FP + FN) \quad (3)$$

In Equations (1)-(3), *TN* represents true negative of identifying benign websites, whereas *FN* is false negative of unidentified genuine websites, and, and *FP* is false positive of unpredicted phishing websites, whilst *TP* represents true positive of identifying phishing websites.

6. TEST RESULTS AND ANALYSIS

Generally, the core findings of this research demonstrate that the presented anti-phishing approach is sufficiently promising with the aid of creating and exploring up-to-date URL features. The earlier methods were implemented by the Spyder platform of python programming (version 3.8) where a number of python scripts were programmed and run upon a portable computer of Intel Core-i7 4610M, 32GB RAM, and 3GH CPU with a Windows 8.1 (64-bit operating system).

The preliminary results of exploring the effectiveness of the URL features generated in this study indicate that a considerable number of features have an important contribution in detecting phishing attacks. Table 2 compares the accuracy of phishing detection by dividing the topmost ranked URL features into a number of groups using the extra tree classification approach:

Table 2. Phishing detection accuracy across several groups of topmost-ranked URL features

URL Feature Groups	Accuracy
30	79%
60	82%
90	80%
120	86%
150	83%

The accuracy figures tabulated above evidently unveil that there is an observable change amongst the ranked feature groups. Based upon the empirical results, the approach of the URL feature prioritization using the extra tree classifier illustrates that some of the ranked feature groups have highly distinctive attributes, whereas others carry less distinguishable ones. The feature group of 120 achieves the highest accuracy (i.e., 86%) which indicates that these features have the potential of constituting more discriminative information. On the other hand, the feature group of 30 accomplishes the least accuracy figure (i.e., 79%) although this might be possible due to the insufficient learning of representing the only topmost 30 features quite well in detecting the target. This could be corroborated by the accomplished accuracy using the next feature group, where the accuracy escalated to 82% when expanding the feature group into 60.

It is worth stating that the feature group of 60, 90, and 150 report rather similar accuracy figures, thus showing that each feature group among the latter does appear to have the same contribution within the anti-phishing framework. In contrast, the feature groups of 30, 60, and 90 can together contribute to significantly raising the accuracy of phishing identification.

Particularly, the feature group of 150 gives lower performance (83%) than the feature group of 120 which accomplishes an accuracy of 86% because the former feature set appears to have less discriminative information. In accordance with this, the feature group of 120 was utilized to explore the effectiveness of the utilized machine learning techniques.

Having applied the selected classifiers in this study upon the testing data (i.e., 6000 samples of benign and phishing URLs), the experimental results reveal that the approach of pattern classification all in all can be robust enough against phishers' attempts in deceiving users. Table 3 presents the performance of the Naive Bayes, AdaBoost, and bagged trees in detecting phishing URLs as follows:

Table 3. The performance of the applied machine learning techniques

Classifier	Recall	Precision	Accuracy
Naive Bayes	86.9%	81.9%	84.5%
AdaBoost	86.8%	89.9%	88.3%
Bagged Trees	84.3%	87.4%	85.8%

According to the empirical results above, the boosting classification approach of AdaBoost outperforms the bagging technique of bagged trees and the Naive Bayes classifier with an overall performance of 86.8% recall, 89.9% precision, and 88.3% accuracy.

The high performance of AdaBoost could be interpreted due to its capacity in obtaining a generalized classification model with fewer faults. Such performance can be accomplished by diminishing the flaws within each sub-classification model. This explanation is supported by Freund, et al. [24] who elaborated that AdaBoost concentrates upon modifying the learning set to empower the fragile classifiers on the training phase via incorporating them and accordingly they can be turned out into a sturdy classification framework. Thereby, the AdaBoost learner approach would more than likely avoid the overfitting classification. In particular, it achieves a 93.2% accuracy result during the stage of learning which confirms that this approach can overcome the concerns of overfitting – thus establishing the best-fitting of classification in the context of this work.

On the other hand, the accuracy results of the bagged trees and the Naive Bayes classifiers do appear to have rather similar effectiveness in detecting phishing URLs. In terms of the bagged trees, the reason why it accomplished less performance (i.e., 84.3% recall, 87.4% precision, and 85.8% accuracy) might be possible because of losing the readability of data within particular instances recognized by a weak combination of sub-classifiers [25]. As a result, the weakness of combining multiple classifications can probably underestimate the role of the URLs features to correctly predicate phishing attacks. This would lead to many challenges and accordingly impact the overall performance in a negative manner. However, with the Naive Bayes classifier being achieved 86.9% recall, 81.9% precision, and 84.5% accuracy, the performance could be lower owing to the presence of the zero-frequency issue. That is, the Naive Bayes classifier technically sets zero probability for patterns on validation if they are not existing within the learning data [26], and this would certainly decrease the average performance of the phishing identification system.

7. COMPARISON WITH PREVIOUS TECHNIQUES

Numerous techniques have been previously introduced to detect cyber-phishing attacks. Accordingly, a well-studied comparison is required to have a concrete basis for describing how reliable the presented anti-phishing framework is in comparison with the existing phishing identification approaches. Table 4 shows the performance of the proposed framework versus the methods applied in [27], [28], and [29] in terms of phishing identification accuracy.

According to the tabulated figures, it is clear, on the one hand, that the proposed phishing detection has accomplished a quite better accuracy than what it has obtained in [29]. On the other hand, the achieved accuracy in [27], [28], and in the suggested anti-phishing within this research does appear to accomplish somewhat alike numbers, which are completely acceptable. Nevertheless, the previous phishing detection approaches do seem to exploit commercial technologies of URL feature generation for identifying specific types of phishing patterns, and could not detect the newly deceived phishing URLs, and this is not the case within the presented phishing identification framework.

Table 4. Comparison between the presented anti-phishing and the related approaches in terms of accuracy

Technique	Accuracy
Presented Method	88.3%
Ref. [27]	88.4%
Ref. [28]	88%
Ref. [29]	84

8. CONCLUSION AND FUTURE WORK

This study has addressed some issues surrounding the research area of phishing identification and accordingly presented a credible and promising anti-phishing framework for tackling them. Experimenting newly collected URLs in this article has clearly stated that cyber phishing attacks are increasingly exacerbating. Although the generation and investigation of up-to-date URL features have unveiled encouraging indications to detect newly deceived phishing attacks, there is a need to study and analyze the forthcoming tricks of phishers, and accordingly engineer advanced/innovative lexical features capable of resisting such scams. Some machine learning techniques can be also robust in detecting phishing URLs; however, others can be ineffective. In this research, the boosting classification approach of AdaBoost has outdone the bagging technique of bagged trees and the Naive Bayes classifier. As such, future research would concentrate upon scrutinizing the newly phishing patterns and seek a way to engineer viable URL features for effectively defeating phishers using a different spectrum of machine learning algorithms. Further work would be also undertaken with the purpose of exploring an approach in which the presented anti-phishing framework will apply online classification rather than batch/offline classification to detect phishing cyber-attacks in real-time throughout.

REFERENCES

[1] M. M. Yadollahi, F. Shoeleh, E. Serkani, A. Madani, H. Gharaee, "An Adaptive Machine Learning Based Approach for Phishing Detection Using Hybrid Features", IEEE Conference on Web Research, pp. 281-286, 2019.

[2] L.H. Abed, M.N. Rashid, O.M. Al Okashi, "Partial Crypto-Compression for Cloud-Based Photo Storage Using DCT and Daubechies 4 Wavelet", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 52, Vol. 14, No. 3, pp. 193-201, September 2022.

[3] B. Banik, A. Sarma, "Lexical Feature Based Feature Selection and Phishing URL Classification Using Machine Learning Techniques", Springer Conference on Machine Learning, Image Processing, Network Security and Data Sciences, pp. 93-105, 2020.

[4] M. Rosenthal, "Must-Know Phishing Statistics", 2022, www.tessian.com/blog/phishing-statistics-2020/

[5] R. Alabdan, "Phishing Attacks Survey: Types, Vectors, and Technical Approaches", Future Internet, Vol. 12, pp. 1-39, October 2020.

[6] T. Li, G. Kou, Y. Peng, "Improving Malicious URLs Detection via Feature Engineering: Linear and Nonlinear Space Transformation Methods", Information Systems, Vol. 91, pp. 1-18, 2020.

[7] S.A. Onashoga, A. Abayomi Alli, O. Idowu, J.O. Okesola, "A Hybrid Approach for Detecting Malicious Web Pages Using Decision Tree and Naive Bayes Algorithms", Georgian Electronic Scientific Journal, Vol. 48, pp. 9-17, 2016.

[8] O. Chistou, N. Pitropakis, P. Papadopoulos, S. McKeown, W.J. Buchanan, "Phishing URL Detection through Top-Level Domain Analysis: A Descriptive Approach", ICISSP, pp. 289-298, 2020.

[9] S. Sarabi, M. Asadnejad, S.A. Tabatabaei Hosseini, S. Rajebi, "Using Artificial Intelligence for Detection of Lymphatic Disease and Investigation on Various Methods of Its Classifications", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 43, Vol. 12, No. 2, pp. 58-65, June 2020.

[10] Bernard Marr, "The Top 10 AI and Machine Learning Use Cases Everyone Should Know about", Forbes, 2016. www.forbes.com/sites/bernardmarr/2016/09/30/what-are-the-top-10-use-cases-for-machine-learning-and-ai/?sh=12e6914d94c9.

[11] R.B. Basnet, A.H. Sung, Q. Liu, "Learning to Detect Phishing URLs", International Journal of Research in Engineering and Technology, Vol. 3, pp. 11-24, 2014.

[12] M.S.I. Mamun, M.A. Rathore, A.H. Lashkari, N. Stakhanova, A.A. Ghorbani, "Detecting Malicious URLs Using Lexical Analysis", Springer Conference on Network and System Security, Vol. 9, pp. 467-482, 2020.

[13] M. Alshira'h, M. Al Fawa'reh, "Detecting Phishing URLs Using Machine Learning Lexical Feature-Based Analysis", Int. J. Adv. Trends Comput. Sci. Eng., Vol. 9, pp. 5828-5837, 2020.

[14] J. Kumar, A. Santhanavijayan, B. Janet, B. Rajendran, B.S. Bindhumadhava, "Phishing Website Classification and Detection Using Machine Learning", IEEE International Conference on Computer Communication and Informatics, pp. 1-6, 2020.

[15] N.S. Zaini, D. Stiawan, M.F. Ab Razak, A. Firdaus, W.I.S. Wan Din, S. Kasim, T. Sutikno, "Phishing Detection System Using Machine Learning Classifiers", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 17, No. 3, pp. 1165-1171, 2020.

[16] B.B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, X. Chang, "A Novel Approach for Phishing URLs Detection Using Lexical Based Machine Learning in a Real-Time Environment", Computer Communications, Vol. 175, pp. 47-57, 2021.

[17] E. Gandotra, D. Gupta, "An Efficient Approach for Phishing Detection Using Machine Learning", Springer Conference in Multimedia Security, pp. 239-253, 2021.

[18] M. Abutaha, M. Ababneh, K. Mahmoud, S.A.H. Baddar, "URL Phishing Detection Using Machine Learning Techniques Based on URLs Lexical Analysis", The IEEE Conference on Information and Communication Systems, pp. 147-152, 2021.

[19] M. Purbay, D. Kumar, "Split Behavior of Supervised Machine Learning Algorithms for Phishing URL Detection", Springer Conference in Advances in VLSI, Communication, and Signal Processing, pp. 497-505, 2021.

[20] A.K. Dutta, "Detecting Phishing Websites Using Machine Learning Technique", PloS One, Vol. 16, pp. 1-17, 2021.

[21] PhishTank, www.phishtank.com/developer_info.php 2022.

[22] X. Zhang, A.H. Lashkari, A.A. Ghorbani, "A Lightweight Online Advertising Classification System Using Lexical-Based Features", SCITEPRESS Conference on e-Business and Telecommunications, Vol. 4, pp. 486-494, 2017.

[23] G. Vrbancic, Jr.I. Fister, V. Podgorelec, "Datasets for Phishing Websites Detection", Data in Brief, Vol. 33, pp. 1-7, 2020.

[24] Y. Freund, R. Schapire, N. Abe, "A Short Introduction to Boosting", Journal-Japanese Society for Artificial Intelligence, Vol. 14, pp. 771-780, 1999.

[25] M. Foley, "My Data Science Notes", 2020, https://bookdown.org/mpfoley1973/data-sci/.

[26] P. Kaviani, S. Dhotre, "Short Survey on Naive Bayes Algorithm", International Journal of Advance Engineering and Research Development, Vol. 11, pp. 607-611, 2017.

[27] M. Aburrous, M. A. Hossain, K. Dahal, F. Thabtah, "Intelligent Phishing Detection System for E-banking Using Fuzzy Data Mining", Expert Systems with Applications, Vol. 37, No. 12, pp. 7913-7921, 2010.

[28] G.A. Montazer, S. Arab Yarmohammadi, "Detection of Phishing Attacks in Iranian E-banking Using a Fuzzy–Rough Hybrid System", Applied Soft Computing, Vol. 35, pp. 482-492, 2015.

[29] Y. Pan, X. Ding, "Anomaly Based Web Phishing Page Detection", The 22nd Annual Computer Security Applications Conference, IEEE, pp. 381-392, December 2006.

BIOGRAPHIES

Name: **Leith**
Middle Name: **Hamid**
Surname: **Abed**
Birthday: 12.5.1985
Birth Place: Fallujah, Iraq
Bachelor: Computer Science, Department of Computer Science, Computer Collage, University of Anbar, Ramadi, Iraq, 2009
Master: Computer Science, Department of Computer Science, Computer Collage, University of Anbar, Ramadi, Iraq, 2012
Doctorate: Cyber Security, Department of Computing, School of Computing, Electronics, and Mathematics, University of Plymouth, Plymouth, United Kingdom, 2019
The Last Scientific Position: Lecturer, Department of Computer Systems Techniques, Anbar Technical Institute, Middle Technical University, Baghdad, Iraq, Since 2019
Research Interests: Bio-Cryptography, Malware Analysis and Detection, Security Management Using Self-Data Destruction and Secret Sharing
Scientific Publications: 8 Papers, 2 Theses

Name: **Hussam**
Middle Name: **Jasim**
Surname: **Mohammed**
Birthday: 27.6.1987
Birth Place: Ramadi, Iraq
Bachelor: Computer Science, Department of Computer Science, Computer Collage, University of Anbar, Ramadi, Iraq, 2009
Master: Computer Science, Department of Computer Science, Computer Collage, University of Anbar, Ramadi, Iraq, 2012
Doctorate: Computing, Department of Computing, School of Computing, Electronics, and Mathematics, University of Plymouth, Plymouth, United Kingdom, 2018
The Last Scientific Position: Lecturer, Computer Center, University of Anbar, Ramadi, Iraq, Since 2018
Research Interests: Cyber Security, Digital Forensics and AI
Scientific Publications: 9 Papers, 2 Theses

Name: **Yaseen**
Middle Name: **Saleem**
Surname: **Yaseen**
Birthday: 29.11.1987
Birth Place: Hit, Iraq
Bachelor: Computer Science, Department of Computer Science, Computer Collage, University of Anbar, Ramadi, Iraq, 2009
Master: Computer Science, Department of Computer Science, Computer Collage, University of Anbar, Ramadi, Iraq, 2013
Doctorate: Computing, Department of Computing, School of Computing, Electronics, and Mathematics, University of Plymouth, Plymouth, United Kingdom, 2019
The Last Scientific Position: Lecturer, Computer Center, University of Anbar, Ramadi, Iraq, Since 2019
Research Interests: Networks, Smart Homes
Scientific Publications: 6 Papers, 2 Theses