Detection of Phishing Websites by Using Machine Learning-Based URL Analysis

Mehmet Korkmaz
Yildiz Technical University
Computer Engineering Department
Istanbul/Turkey
mkorkmazzz@gmail.com

Ozgur Koray Sahingoz
Istanbul Kultur University
Computer Engineering Department
Istanbul/Turkey
sahingoz@gmail.com

Banu Diri
Yildiz Technical University
Computer Engineering Department
Istanbul/Turkey
diri@yildiz.edu.tr

***Abstract*— In recent years, with the increasing use of mobile devices, there is a growing trend to move almost all real-world operations to the cyberworld. Although this makes easy our daily lives, it also brings many security breaches due to the anonymous structure of the Internet. Used antivirus programs and firewall systems can prevent most of the attacks. However, experienced attackers target on the weakness of the computer users by trying to phish them with bogus webpages. These pages imitate some popular banking, social media, e-commerce, etc. sites to steal some sensitive information such as, user-ids, passwords, bank account, credit card numbers, etc. Phishing detection is a challenging problem, and many different solutions are proposed in the market as a blacklist, rule-based detection, anomaly-based detection, etc. In the literature, it is seen that current works tend on the use of machine learning-based anomaly detection due to its dynamic structure, especially for catching the "zero-day" attacks. In this paper, we proposed a machine learning-based phishing detection system by using eight different algorithms to analyze the URLs, and three different datasets to compare the results with other works. The experimental results depict that the proposed models have an outstanding performance with a success rate.**

Keywords*— *cybersecurity, phishing, machine learning, website classification

I. INTRODUCTION

In our daily life, we carry out most of our work on digital platforms. Using a computer and the internet in many areas facilitates our business and private life. It allows us to complete our transaction and operations quickly in areas such as trade, health, education, communication, banking, aviation, research, engineering, entertainment, and public services. The users who need to access a local network have been able to easily connect to the Internet anywhere and anytime with the development of mobile and wireless technologies. Although this situation provides great convenience, it has revealed serious deficits in terms of information security. Thus, the need for users in cyberspace to take measures against possible cyber-attacks has emerged.

Attacks can be carried out by people such as cybercriminals, pirates, or non-malicious (white-capped) attackers and hacktivists [1]. The aim is to reach the computer or the information it contains or to capture personal information in different ways. The attacks, as internet worms (Morris Worm), started in 1988, and they have been carried out until today. These attacks are mainly targeted in the following areas: fraud, forgery, force, shakedown, hacking, service blocking, malware applications, illegal digital contents and social engineering [2]. Reaching with a wide range of target users, attackers aim to get a lot of information and/or money. According to Kaspersky's data, the average cost of an attack in 2019 (depending on the size of the attack) is between $ 108K and $ 1.4 billion. In addition, the money spent on global security products and services is around $ 124 billion [3].

Among these attacks, the most widespread and also critical one is "phishing attacks". In this type of attack, cybercriminals especially use an email or other social networking communication channels. Attackers reach the victim users by giving the impression that the post was sent from a reliable source, such as a bank, e-commerce site, or similar. Thus, they try to access sensitive information of them [4]. Attackers then access their victims' accounts by using this information. Thus, it causes pecuniary loss and intangible damages.

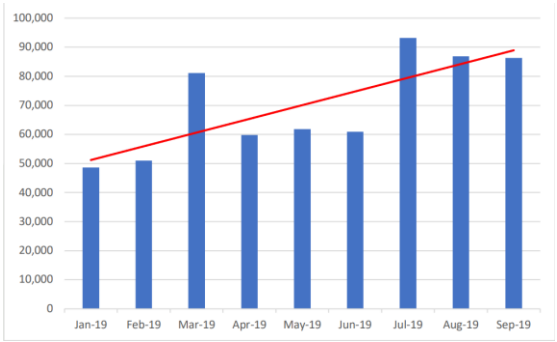

Fig. 1. Number of Phishing Sites [6]

The method of reaching target users in phishing attacks has continuously increased since the last decade. This method has been carried out in the 1990s as an algorithm-based, in the early 2000s based on e-mail, then as Domain Spoofing and in recent years via HTTPs. Due to the size of the mass attacked in recent years, the cost and effect of the attacks on the users have been high. The average financial cost of the data breach as part of the phishing attacks in 2019 is $ 3.86 million, and the approximate cost of the BEC (Business Email Compromise) phrases is estimated to be around $ 12 billion. Also, it is known that about 15% of people who are attacked are at least one more target [5]. With this result, it can be said that phishing attacks will continue to being carried out in the ongoing years. Figure 1 also supports this idea and show the number of phishing sites in 2019, and as can be seen from it, there is an increasing trend in this type of attack. In this regard, regular reports published by APWG (Anti Phishing

Working Group) are an important guide for the researchers. According to the reports, the number of phishing sites is reached to approximately 640,000 sites were determined in 2018, and in the first three quarters of 2019, this number was reported as 629,611 [6]. Reports for the last quarter of 2019 have not been published yet. However, it can be said that the phishing attacks not only continue, but also there will be an increase in the number of attack types compared to the previous year.

This increase indicates that phishing attacks are used more by attackers. Because they are easy to design. Phishing attacks are based on the attacker's creation of a fake website, as depicted in Figure 2. First, a phisher makes fake websites, including a phishing kit. Then, the victim is directed to the fake website with the prepared email. Believing that the e-mail and URL are secure, the victim uses the fake website by clicking on the URL. After this moment, the Phishing kit receives the victim's credentials and sends it to the phisher. Finally, Phisher makes fake earning from the legitimate website using the victim's credentials. These sites generally have very similar or even identical visuals. In an e-mail that is thought to be sent from a trusted source, the target is directed to this fake website. The target accesses the website at the relevant URL via e-mail, which she/he finds reliable and writes the information that the attacker wants to obtain. The attacker receives the necessary information and uses it in the real system.

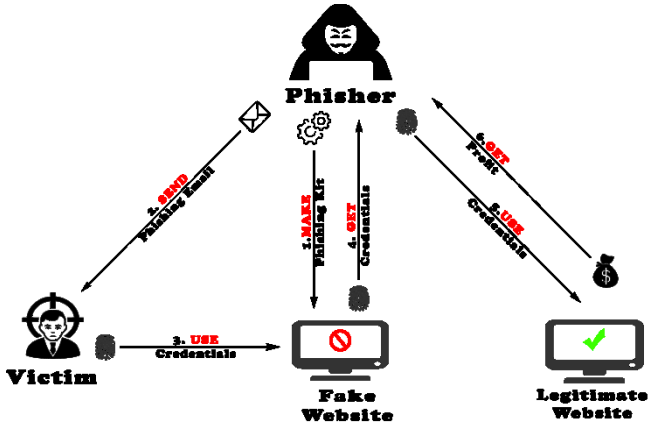

Fig. 2. Life Cycle of a Phishing Attack

In this way, the attacker gets information and / or earnings. Reliable e-mail contents are created in different ways for the victim to believe. Previously, e-mails with low probability offers, urgent texts, links, or attachments that may be relevant and unusual senders were used. Today, reliable organizations or similar links to these organizations are preferred. Attackers prefer reaching to victims by using a secure communication protocol, and the real URL is served by changing in a way that is close to the original. At this stage, if the victim knows the website is fake, he can protect himself from the attack. It is very difficult for the victim to detect the attack by himself, because mainly this type of messages gave some alert messages to the users, and aims to make panic for entering his confidential data to the forwarded page.

Therefore, different decision support or detection systems have been developed to protect the end user against phishing attacks. Different approaches are used in these systems, such as Blacklists, Rule-based systems, Similarity-based systems, and Machine Learning based systems, etc. The literature was reviewed in detail, and the studies in this context were examined carefully. Currently, machine learning-based systems are especially preferred for its protection mechanism to the zero-day attacks. Therefore, in this paper, it is aimed to implement a phishing detection system based on a machine learning algorithm for investigating the URL address of the target web page. With the idea of existing improvable ways of the designed system, it is aimed at the detection of phishing attacks in a short time, without the need for third-party services, and also without waiting for the blacklists to be updated.

The paper is organized as follows: in the next section, the literature review is included. In the third section, the details of the designed system are explained. In the fourth section and fifth section, the results obtained in the experiments are shared, and conclusion and future studies are drawn, respectively.

II. Literature Survey

In this section, it was discussed some of the techniques which based on list, rule, visual similarity, and machine learning.

A. List Based Phishing Detection Systems

These systems use two lists to classify phishing and non-phishing websites. These are called whitelist and blacklist. The whitelist contains safe and legitimate websites, while the blacklist includes websites classified as phishing.

In [7], researchers used the whitelist to identify phishing sites. In the study, access to websites takes place only on the condition that the URL is in the whitelist. Another method is the blacklist approach. In the literature, apart from applications such as Google Safe Browsing API, PhishNet, there are also some studies using blacklists like [8]. In blacklist-based systems, the URL is checked from the list and access to the URL if it is not included in the list. The biggest disadvantage of these systems is that the small change in the URL prevents matching in the list. Additionally, the newest attacks, which are named zero-day attacks, cannot be catches with these type protection systems.

B. Rule-Based Phishing Detection Systems

In these systems, features are obtained based on relational rule mining. The rules are estimated to emphasize features that are more common in phishing URLs [9]. In studies using this type of system, it is aimed to use effective features more actively in the classification. In these systems, a set of rules are determined. Thus, the system gives a higher accuracy rate when trained with these rules.

In this context, like CANTINA [10] study, the Term Frequency - Inverse Document Frequency (TF-IDF) and rules were used to detect phishing attacks. In addition, in similar studies, models were created by using some features and rules.

C. Visual Similarity-Based Phishing Detection Systems

These systems are based on visual similarity comparison of the web pages. Phishing and non-phishing sites are classified by taking a server-side view of them. These two data are compared with image processing techniques. Fake websites are often designed very close to the original ones. But visually, there are minor differences between them. It is easier to notice these differences, which users cannot easily notice, with image processing techniques. According to the similarity obtained, it is decided whether the website is phishing or not. In the literature, as in the study [11], there are studies, which detect the differences based on basic similarities.

D. Machine Learning Based Phishing Detection Systems

The detection of the phishing website in Machine Learning Based Phishing Detection Systems is based on the classification of the specified features by using some artificial intelligence techniques. Features are created by collecting in different categories such as URL, domain name, website features or website content etc. Due to the dynamic structure, especially for the detection of the anomaly in the web sites, it has high popularity on the security of the users.

In the literature, there are some works on this type of detection mechanism. The previously mentioned CANTINA project [10] was also done by using the machine learning method. According to Tf-Idf and heuristic approaches, they detected a 90% accuracy rate. Researchers developed a phishing protection system called PhishWHO applied with three steps in [12] to determine if the website is legitimate. According to their 3-Tier Identity Matching System, they detected a 96.10% accuracy rate. In [13], phishing websites are defined by classifying them with URL attributes such as length, number of special characters, directory, domain name, and file name. The title and priority order of the incoming e-mail is discussed in [14]. In [15], URL-based features are used together with features related to transport layer security (Length, slash number, point number and location). They detected the accuracy rate 93% by using the rules obtained by apriori algorithm. In [16], a nonlinear regression strategy is used to determine whether a website is phishing. The system was operated by using the harmony search and Support Vector Machine (SVM) methods. They used 11055 webpages and 20 features. Features were selected by using the decision tree method instead of the wrapper. They detected the accuracy rate 92.80% by using nonlinear regression based on HS led. In another study [17], a phishing detection system with 209 word-vector features and 17 NLP based features was proposed. The Random Forest, SMO, and Naïve Bayes algorithms were compared, and the best result was obtained with the Random Forest algorithm in the hybrid approach with an accuracy rate of 89.9%.

In the system proposed in [18], the number of NLP vectors was increased, and three different machine learning algorithms were compared according to their accuracy values. The Random Forest, SMO, and Naïve Bayes algorithms were compared, and the best result was obtained with the Random Forest algorithm in the hybrid approach with an accuracy rate of 97.2%. Researchers implemented a phishing detection system in [19] by using adaptive self-configuring neural networks for classification. In the study, 17 different features are used, which are also used third-party services. Therefore, it was stated that much more time is needed in real-life implementation. In [20], to distinguish phishing websites from legitimate ones, a machine learning method was used with 19 features from the URL and Source code, which do not depend on any third party. The results showed that, by using this system, a 99.09% accuracy rate was calculated.

In [21], the neural network-based classification method is proposed for the detection of phishing websites using the Monte Carlo algorithm and risk reduction principle. [29] focused on the effect of the training functions on neural network to increase the efficiency of the proposals. In [22], four different categories are specified: e-mail headers, URLs in the content, HTML content, and main text. The classification was made in machine learning by using 50 features in these categories. The results demonstrated a 98.6% accuracy rate. In [23], machine learning algorithms are compared with the features extracted from URL, source code, and third-party services. Principal component analysis Random Forest performed the accuracy of 99.55% with also detecting zero-day phishing attacks. In an NLP-based study [24], the text of e-mails was analyzed and classified. In [25], classification was made using TF-IDF, hand-crafted features, and both, along with 35 features. In the study, the phishing attack detection rates were compared by using 6 different algorithms. The best result was obtained in the Random Forest algorithm, with an accuracy rate of 99.55%.

In this paper, it is observed that higher accuracy rate can be achieved by using different features with examining previous studies. Different from previous studies, a new study based on features selected and coded from more features was made. 58 features were determined by making URL analysis. Using the machine learning method, the accuracy rates and model training times of different algorithms were compared.

III. Proposed System

In this paper, we aimed to implement a phishing detection system by analyzing the URL of the webpage. URL is a complex string that expresses syntactically and semantically expressions for a resource available over the Internet. When examined in detail, the structure of the URL is shown in Figure 3. In its most basic form, it is as follows: <protocol>://<hostname><uri>, in the complex form it is detailed as follows.

Fig. 3. Form of URL

Fields such as domain, subdomain, Top Level Domain (TLD), protocol, directory, file name, path and query allow creating different URL addresses. These related fields in the phishing URLs are generally different from the legitimate

ones on websites. Therefore, URLs have an important place in detecting phishing attacks especially for classifying the web page quickly.

It was observed in the literature review that effective features obtained from the URL increase the accuracy of the classification. Additionally, third-party service usage, site layout, CSS, content, meta information, etc. features can also improve accuracy. However, these features will cause an increase in the classification time of the new websites which needed to be classified. The proposed model, trained only with the features obtained from the URL, is expected to classify in a shorter time than other models. Considering this information, only URL analysis is planned in the study. Thus, the classification results of the obtained features in different algorithms in machine learning are compared. In addition, the results from another study [25] with the same dataset are compared with those of the current study.

A. Datasets

Phistank.com [26] is a site where phishing URLs are detected and can be accessed via API call. It is an organization whose data is used by companies such as YahooMail, McAfee, APWG, Mozilla, Opera, Kaspersky and Avira. In the literature review, it has been observed that the phishing data used in the machine learning method are generally taken from Phistank.com. It made the necessary classification about the previous webpage addresses. It also gave the data about the positive/negative (phishing/not phishing) classification. However, it does not store the content of the webpages; therefore, it is a good source for the URL based analysis.

In this paper, datasets in [25], which are open-source and available, are used. For making a comparative study, we preferred open datasets. Three datasets were used in this paper, which the researchers named their system as CatchPhish. The first of these datasets: Legitimate sites from Alexa database and phishing sites from PhishTank. Second: Legitimate sites from common-crawl and phishing sites from PhishTank. Third: Legitimate sites from both common-crawl and Alexa databases whereas phishing sites from PhishTank. The numbers of URLs in these datasets are given in Table I.

TABLE I. DATASETS

	Dataset-1	Dataset-2	Dataset-3
Phishing	40,668	40,668	40,668
Not Phishing	43,189	42,220	85,409
Total	83,857	82,888	126,077

B. Feature Extraction

The effectiveness of the trained system is directly related to the used features and machine learning algorithms. Therefore, to detect the critical features, we did an extensive literature review. In addition to the studies that only analyze the URL, studies that use features in different categories such as e-mail content analysis and website analysis were also examined.

The features of URL have been examined separately in the hostname, domain, and path sections. In our study, we detected 58 different features on this content. These features were obtained with scripts written using the Python programming language. After sorting the features with the Random Forest Classifier, it was decided to use the best 48 of them. Thus, a higher accuracy rate is aimed to achieve. The features used in the study are listed in Table II.

TABLE II. FEATURES OF URL

#		Name	#		Name	#		Name
1	Number of	Words	17	Number of	Underscore	33	Length of	Path
2		Url Paths	18		Dots in Host	34		Subdomain
3		Digits	19		Dots in Path	35		Url
4		Ampersand	20		Hyphen in Host	36		Domain Name
5		Sensitive Words	21	Has	Url Without Www	37		Longest Word
6		“?”	22		Query	38		Parameters
7		Special Chars	23		Character Repetition	39		Average Word
8		Punctuation	24		Https Protocol	40		Shortest Word
9		Dots in Sub Domain	25		Digits in Domain name	41		Longest Word in Hostname
10		Tld in Paths	26		Ip Address	42		Host
11		Subdomain	27		Subdomain	43	Ratio of	Url/Path
12		Digits in Host	28		“Www” Or “Com”	44		Vowel/Consonant
13		Dots	29		“@”	45		Digit/Letter
14		Words in Host Name	30		Hyphen in Url	46		Longest/Shortest Word Length
15		Hyphen in Path	31		Suffix	47	-	STD of Words Length
16		“=”	32		Redirected	48	-	Port Number

The “/”, ”:”, “\”, “.”, “?” , “=”, “&”, “-“ characters have been used to obtain the tokens in the URL. Not every word mentioned in these tokens has been found separately. The token has been considered as a whole. The most common tokens were identified and used in feature 12. For feature 25, it was checked whether the top 10 TLDs of Alexa rank 1 million data [27] are in the URL. The system is designed for URL classification in a short time by using 48 features, without the need for third party services, and without content analysis.

C. System Implementation

In the machine learning based system, 8 different algorithms were run in the experiment. These are: Logistic Regression (LR), K-Nearest Neighborhood (KNN), Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), XGBoost, Random Forest (RF) and Artificial Neural Network (ANN). The models created with these algorithms trained by using the Sklearn library in the Python programming language.

Logistic Regression is an algorithm that can generate effective predictions when the dependent variable is collected in two classes. Although it provides an advantage in this

respect, factors such as incompatibility between features, repetition of features, outlier values affect the prediction negatively. *KNN* is a fast and efficient algorithm that produces estimates based on the distance of k neighbors. However, calculating this distance in large data means using a lot of memory. At the same time, the correct k value is extremely important for the result. *SVM* is an easy-to-implement algorithm that can work with a large number of independent variables. It can produce effective solutions by using the core trick in nonlinear problems. However, it is not very suitable for large databases. And if there is a lot of noise, it cannot perform well.

Decision Tree is an algorithm that works by dividing the dataset into sub-sections so that the tree is formed. In the formed tree, each node is assigned to a feature and each leaf to a class. This algorithm, which is easy to explain and has little hyperparameter, cannot be effective in multi-class or small data sets. It can also be easily overfitted. *Naive Bayes* is a classification algorithm based on conditional probability. It works according to Bayes' theorem. It is preferred due to its easy implementation and less training time. However, it is not preferred for reasons such as lower estimation with less data, assuming that the features are independent of each other. *XGBoost* is an algorithm based on gradient boosted decision trees that put speed and performance in the foreground. It aims to reduce errors in the previous tree by producing a new tree each time. However, this process can take a long time. It can also be easily overfitted.

Random Forest is an algorithm that works with the Ensemble Learning technique by creating a large number of trees in the dataset. It divides into subtrees. It is a powerful algorithm that does not require feature scaling, is resistant to separation (overfit), and less affected by noise. However, it has been training for a long time that needs memory and processor power. *Artificial Neural Network (ANN)* is an algorithm that works with at least three layers with its structure like biological neural networks. While training the model, it uses fewer statistics; it can detect the connections between the features and generalize well. Although it may be prone to overfitting, it may require high memory and processor power (depending on the size of the model). Experimental results were obtained by performing 10-Fold Cross Validation so that statistical results were validated. The experiments were run on the computer with 6 GB Ram capacity and 2.0 Ghz 2 core Intel i7 processor.

IV. Experimental Results

In this section, the experimental results of the algorithms whose advantages and disadvantages are explained above are deactivated in three datasets. In the experiment, the results are shown in two categories as accuracy rates and training time. Also, in this section, the results of the current study are compared with the results of the experiment in [25].

A. Test Results on Datasets

For the three data sets used, results were obtained in 8 different algorithms. In the tables below, these results are shared separately. Table III contains the accuracy rate and training period for Dataset-1. According to these data, the highest test classification result was obtained in the model using RF classifier with an accuracy rate of 94.59%. At the same time, the NB algorithm has the shortest training time.

TABLE III. Test Results of Classifiers on Dataset-1

Classifier	Accuracy (%)	Time(sec.)
XGBOOST	92.95	622.6
RF	**94.59**	784.3
LR	91.31	20.5
KNN	91.49	413.7
SVM	87.03	7537.4
DT	92.59	18.8
ANN	94.35	57.6
NB	88.35	3.9

Table IV contains the accuracy rate and training time taken for Dataset-2. According to these data, the highest test classification result was obtained in the model using RF classifier with an accuracy rate of 90.50%. At the same time, the NB algorithm has the shortest training time.

TABLE IV. Test Results of Classifiers on Dataset-2

Classifier	Accuracy (%)	Time(sec.)
XGBOOST	83.69	395.6
RF	**90.5**	439.1
LR	75.65	51.7
KNN	81.47	154.5
SVM	70.2	9833.6
DT	81.67	23
ANN	88.22	40.3
NB	70.05	1.8

Table V contains the accuracy rate and training time taken for Dataset-3. According to these data, the highest test classification result was obtained in the model using RF classifier with an accuracy rate of 91.26%. At the same time, the NB algorithm has the shortest training time.

TABLE V. Test Results of Classifiers on Dataset-3

Classifier	Accuracy (%)	Time(sec.)
XGBOOST	83.27	271.4
RF	**91.26**	133.2
LR	78.26	63.5
KNN	81.11	262.1
SVM	76.76	15956.2
DT	81.66	17.8
ANN	88.88	29.4
NB	67.04	2.1

In general, when the results are analyzed, the models using LR, SVM and NB classifiers have a lower accuracy rate. It can be said that such a result depends on a large number of data, the variables are not very independent from each other, and the URLs in Dataset-2 have longer paths than Dataset-1. When Table III, Table IV and Table V are examined, it can be seen that RF algorithm has the highest accuracy rate.

When the training times of the models are examined, it is seen that the NB, DT, LR and ANN algorithms give better results. SVM draws attention with its high train time. Considering two of the models with the highest accuracy (RF and ANN), RF has more training time than ANN (13.6 times in Dataset-1, 10.9 times in Dataset-2 and 4.5 times in Dataset-3).

B. Comparison of Test Results

In this paper, the work in [25] is named as E1. In experiments conducted in E1, it is reported that the model with the highest accuracy rate was obtained with RF algorithm in all datasets. This study was named as E2. In E2, the highest accuracy rate results were obtained with RF algorithm too. In Table VI, experimental results with RF algorithm in E1 and E2 are shared for all three datasets.

TABLE VI. COMPARISON OF TEST RESULTS FOR E1 AND E2

Data Sets	D1		D2		D3	
Experiments	E1	E2	E1	E2	E1	E2
TPR	94.41	94.69	88.6	91.11	91.67	93.02
FPR	5.79	5.51	10.69	10.1	12.9	12.47
TNR	94.21	94.49	89.31	89.9	87.1	87.53
FNR	5.59	5.31	11.4	8.89	8.33	6.98
Accuracy	94.32	94.59	88.95	90.5	90.28	91.26
Precision	94.56	94.5	89.86	90.02	94.22	88.18
F-measure	94.49	94.59	89.23	90.56	92.93	90.53

According to the comparison results for all three datasets, the test results obtained in this paper using 48 features have a higher accuracy rate. Although the accuracy rate was very close to other study results in Dataset-1, 1.55% improvement in Dataset-2, and 0.98% improvement in Dataset-3 were achieved.

C. Runtime Efficiency of the System

For using the proposed system in run-time, the detection time of a web page from its URL is critical. To decrease the execution time, we aimed not to use third-party services from the Internet. The reached values for each learning algorithms are depicted in Table VII.Due to the structure of the trained system, the execution time for a single URL address and 100 URL has not a considerable difference.

As seen from Table VII, XGBOOST, RF, LR, DT, ANN algorithms give the better execution time between the tested models. It can be seen from the table that ANN algorithm has the best time.

TABLE VII. TEST RESULTS OF CLASSIFIERS ON DATASET-2

Classifier	Feature Analysis of the URL (msec)	Execution Time (msec) (1 URL)	Execution Time (msec) (100 URL)
XGBOOST	0.083	0.001	0.003
RF		0.004	0.007
LR		0.001	0.0009
KNN		0.048	1.339
SVM		0.005	0.505
DT		0.001	0.004
ANN		**0.0009**	**0.001**
NB		0.001	0.015

Systems which using the RF or ANN algorithm take less time than other algorithms to detect Phishing URLs. It is calculated that the detection time of a URL takes approximately 0.083 seconds when considered in terms of feature extraction. Also, when the test time is added, it can be said that the time does not differ much for all algorithms except KNN and SVM. When modeling the system, it is preferable to use the algorithm with the highest accuracy rate for the analysis of 1 URL. In this case, it will be appropriate to use RF algorithm. Apart from this, the reason for the preference for the algorithm to be used can be the train time along with the high accuracy rate. Accordingly, ANN algorithm may be used.

V. CONCLUSIONS AND FUTURE WORKS

In recent years, due to the evolving technologies on networking not only for traditional web applications but also for mobile and social networking tools, phishing attacks have become one of the important threats in cyberspace. Although most of security attacks target on system vulnerabilities, phishing exploits the vulnerabilities of the human end-users. Therefore, the main defense form for the companies is informing the employees about this type of attack. However, security managers can get some additional protection mechanism which can be executed either as a decision support system for the user or as a prevention mechanism on the servers.

In this paper, we aimed to implement a phishing detection system by using some machine learning algorithms. The proposed systems are tested with some recent datasets in the literature and reached results are compared with the newest works in the literature. The comparison results show that the proposed systems enhance the efficiency of phishing detection and reach very good accuracy rates. As future works, firstly, it is aimed to create a new and huge dataset for URL based Phishing Detection Systems. With the use of this dataset, we plan to enhance our system by using some hybrid algorithms, and also deep learning models as mentioned in [28]

REFERENCES

[1] State of Cybersecurity Implications for 2016. An ISACA and RSA Conference Survey. [Online]. Available:

https://cybersecurity.isaca.org/csx-resources/state-of-cybersecurity-implications-for-2016. [Accessed: 09-Mar-2020].

[2] Republic of Turkey, "National Cyber Security Strategy, 2016," Ministry of Transport Martime Affairs and Communications.

[3] R. Loftus, "What cybersecurity trends should you look out for in 2020?," Daily English Global blogkasperskycom. [Online]. Available: https://www.kaspersky.com/blog/secure-futures-magazine/2020-cybersecurity-predictions/32068/. [Accessed: 09-Mar-2020].

[4] E. Buber, Ö. Demir and O. K. Sahingoz, "Feature selections for the machine learning based detection of phishing websites," 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, 2017, pp. 1-5.

[5] "Retruster," Retruster. [Online]. Available: https://retruster.com/blog/2019-phishing-and-email-fraud-statistics.html. [Accessed: 09-Mar-2020].

[6] "Phishing Activity Trends Reports, 1st-2nd-3rd Half" APWG. [Online]. Available: https://apwg.org/trendsreports/. [Accessed: 09-Mar-2020].

[7] Y. Cao, W. Han, and Y. Le, "Anti-phishing based on automated individual white-list," Proceedings of the 4th ACM workshop on Digital identity management - DIM 08, pp. 51–60, 2008.

[8] M. Sharifi and S. H. Siadati, "A phishing sites blacklist generator," 2008 IEEE/ACS International Conference on Computer Systems and Applications, pp. 840–843, 2008.

[9] M. Khonji, Y. Iraqi, and A. Jones, "Phishing Detection: A Literature Survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091–2121, 2013.

[10] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina,a content based approach to detecting phishing web sites" Proceedings of the 16th international conference on World Wide Web - WWW 07, pp. 639-648, 2007.

[11] L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng, "Detection of phishing webpages based on visual similarity," Special interest tracks and posters of the 14th international conference on World Wide Web - WWW 05, pp. 1060-1061, 2005.

[12] C. L. Tan, K. L. Chiew, K. Wong, and S. N. Sze, "PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder," Decision Support Systems, vol. 88, pp. 18–27, 2016.

[13] A. Le, A. Markopoulou, and M. Faloutsos, "PhishDef: URL names say it all," 2011 Proceedings IEEE INFOCOM, pp. 191-195, 2011.

[14] R. Islam and J. Abawajy, "A multi-tier phishing detection and filtering approach," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 324–335, 2013.

[15] S. C. Jeeva and E. B. Rajsingh, "Intelligent phishing url detection using association rule mining," Human-centric Computing and Information Sciences, vol. 6, no. 1, Oct. 2016.

[16] M. Babagoli, M. P. Aghababa, and V. Solouk, "Heuristic nonlinear regression strategy for detecting phishing websites," Soft Computing, vol. 23, no. 12, pp. 4315–4327, 2018.

[17] E. Buber, B. Diri, and O. K. Sahingoz, "Detecting phishing attacks from URL by using NLP techniques," 2017 International Conference on Computer Science and Engineering (UBMK), pp. 337-342, 2017.

[18] E. Buber, B. Diri, and O. K. Sahingoz, "NLP Based Phishing Attack Detection from URLs," Advances in Intelligent Systems and Computing Intelligent Systems Design and Applications, pp. 608–618, 2018.

[19] R. M. Mohammad, F. Thabtah, and L. Mccluskey, "Predicting phishing websites based on self-structuring neural network," Neural Computing and Applications, vol. 25, no. 2, pp. 443–458, 2013.

[20] A. K. Jain and B. B. Gupta, "Towards detection of phishing websites on client-side using machine learning based approach," Telecommunication Systems, vol. 68, no. 4, pp. 687–700, 2017.

[21] F. Feng, Q. Zhou, Z. Shen, X. Yang, L. Han & J. Wang, "The application of a novel neural network in the detection of phishing websites," Journal of Ambient Intelligence and Humanized Computing, pp 1-15, 2018.

[22] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," Decision Support Systems, vol. 107, pp. 88–102, 2018.

[23] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," Neural Computing and Applications, vol. 31, no. 8, pp. 3851–3873, Jun. 2018.

[24] T. Peng, I. Harris, and Y. Sawa, "Detecting Phishing Attacks Using Natural Language Processing and Machine Learning," 2018 IEEE 12th International Conference on Semantic Computing (ICSC), pp. 300-301, 2018.

[25] R. S. Rao, T. Vaishnavi, and A. R. Pais, "CatchPhish: detection of phishing websites by inspecting URLs," Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 2, pp. 813–825, Oct. 2019.

[26] PhishTank-Friends of PhishTank," PhishTank. [Online]. Available: https://www.phishtank.com/friends.php. [Accessed: 09-Mar-2020].

[27] Amazon, Alexa Statistic, [Online]. Available: http://s3.amazonaws.com/alexa-static/top-1m.csv.zip. [Accessed: 09-Mar-2020].

[28] G. Karatas, O. Demir and O. K. Sahingoz, "Deep Learning in Intrusion Detection Systems," 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), ANKARA, Turkey, 2018, pp. 113-116, doi: 10.1109/IBIGDELFT.2018.8625278.

[29] G. Karatas and O. K. Sahingoz, "Neural network based intrusion detection systems with different training functions," 2018 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, 2018, pp. 1-6, doi: 10.1109/ISDFS.2018.8355327.