Detecting Malicious URL

Fatimah Alkhudair
College of Computer
Qassim University
Saudi Arabia
391214651@qu.edu.sa

Mada Alassaf
College of Computer
Qassim University
Saudi Arabia
391215494@qu.edu.sa

Rehan Ullah Khan
IT Department, IAG Group
College of Computer
Qassim University
Saudi Arabia
re.khan@qu.edu.sa

Shouq Alfarraj
College of Computer
Qassim University
Saudi Arabia
391200404@qu.edu.sa

***Abstract*— With the ubiquitous use of Internet technology and the rapid development, many of essential life activities (banking, medicine, social networks, etc.) have shifted into Web-based services. As Web-based services became increasingly ubiquitous, they attract cybercriminals through malicious Uniform Resource Locator (URLs) and other ways to perform illegal actions. In recent years, malicious URLs have become an important security issue and an increasingly serious threat to the Web security, it is crucial to detect such threats. In this paper, therefore we applied four machine learning algorithms to detect malicious URLs. The experimental results show that the best performance is achieved by Random Forest algorithm with an accuracy of 96% and 95% during two test phases.**

Keywords—CyberSecurity, Malicious URL, Machine Learning Algorithms, Classification, Random Forest.

I. INTRODUCTION

The Internet is becoming more significant and an essential part of our daily life. Many essential services are being done by the Internet and depend on its functionality and security (e.g. business, learning, banking, social networking, medicine, and many others), which are Web-based applications. As the Web is becoming more and more important, cybercriminals (who can illegally use it to exploit vulnerabilities for illegal actions) are becoming more sophisticated. The inherent weakness of the Internet structure gives cybercriminals possible chances to conduct numerous attacks against Web applications which compromise its security [1].

Web applications can be accessed via URLs, which requires either typing it manually in the address bar of the browser or by clicking on it. URLs can be an entry point for cybercriminals, who can conduct a web attack by injecting a malicious code into the URL [2]. Malicious URLs is becoming an increasingly serious issue in current years, the 2018 Internet Security Threat Report (ISTR) state that the amount of malicious URLs in 2017 increased by 2.8% with 7.8% (1 in 13) of all URLs classified as malicious [3]. Thus, it is essential to detect such threats, and this detection can be done by using machine learning techniques to extract and analyze the effective features of URLs, that will identify malicious URLs more accurately, increase the security of the network, and decrease the propagation of malicious data.

Therefore, our study focuses on detection of malicious URLs using machine learning. Four classification algorithms namely Random forest, K Nearest-Neighbor, J48 and BayesNet are applied to public dataset which consist of 20 features and contains 1781 records. In our experiment, we examine the performance of each algorithm two times; first time we use all features of the dataset, second time only six features from the dataset are used. The experiment result shows that the Random Forest classifier have the highest accuracy.

The remaining of this paper is organized as follows. Section II presents related work on malicious URL detection. Section III show the proposed classification approach. The experimental setup is illustrated in section IV. Section V presents the results and discusses them. Finally, the conclusion of this paper is shown in Section VI.

II. RELATED WORK

Cui et al. [2] proposed a detection approach which uses machine learning techniques for automatic classification of URL as a malicious website or benign. The authors used statistical analysis algorithms, based on joining feature extraction and gradient learning by using a sigmoidal threshold level to select effective features. Their results achieved an accuracy of 98.7% on practical application.

Altaher [4] suggested a hybrid approach to detect the website as Phishing, Legitimate or Suspicious websites. The approach is the result of merging two machine learning algorithms, which are the Support Vector Machine (SVM) algorithm, and K Nearest-Neighbors (KNN) algorithm in two phases. The experiment on the proposed approach gets an accuracy of 90.04 %.

Tao et al. [5] proposed an automatic method to detect the malicious website based on HTTP session information, which is covering the HTTP session header based features and the domain based features. The authors used supervised machine learning to construct a classifier to detect the malicious website. The approach classify 92.2% of malicious websites.

Wang [6] developed a hybrid analysis method, which consist of both static and dynamic detection methods. They classified the static features into three groups URL, HTML document and JavaScript. In addition, they classified the web pages into three categories benign, malicious and unknown. The performance of different types of features were compared by applying Receiver Operating Characteristic (ROC) curve, the accuracy of URL features was 93% which was better than the other features (HTML and JavaScript).

Sahingoz et al. [7] proposed a system for detecting phishing URLs, by applying various machine learning algorithms. As well as Natural Language Processing (NLP), word vector and hybrid features. The NLP features performed better than word vector features, while the performance of the system increased when both features of NLP and word vector are used together with rate of 2.24% and 13.14% according to NLP based features and word vector respectively.

Sirageldin et al. [8] used machine learning algorithms to detect malicious web pages depending on two feature groups which are URL lexical and page content. The combination of features provided best result; false positive rate was reduced, and the best performance was achieved by Artificial Neural Network (ANN) with an accuracy of 96%.

978-1-7281-2680-7/20/$31.00 ©2020 IEEE

Liu et al. [9] provided an experimental study on the detection of malicious URLs, using six machine learning algorithms. Authors focused on the character frequency and structural features of URL, in order to obtain the more crucial features of malicious URLs. Their results demonstrate that the extracted URL features perform best and feasible when it is combined with the Random Forest classification algorithm.

Zhao et al. [10] used machine learning models for multi-classification of malicious URLs, two methods were used and compared: 1) Gated Recurrent Neural Network (GRU) method. 2) Random Forest (RF) classifier (with well-selected features). URLs of six types (XSS injection, SQL injection, directory travels, legitimate, sensitive file attack and other attacks) are used to evaluate the two approaches. The results showed that the model GRU achieved an accuracy of 98% and is 2.1% greater than RF model.

Patgiri et al. [11] proposed a system that uses machine learning algorithms to detect malicious URLs. They divided the collected dataset into training and test data to three different ratios and calculated the accuracy of classifiers. The comparison results demonstrate that the split ratio 80:20 is more accurate and average accuracy of Random Forest (RF) is higher than Support Vector Machine (SVM) and observe that RF has low standard deviation than SVM in accuracy.

III. Proposed Classification Approach

The dataset is pre-processed, and then passed to feature selection. The dataset is then divided into two parts, the training and testing data. We use 10 folds cross-validation to split the data into 10 parts 9 parts used for training, and 1 part for testing and test it 10 times. Several algorithms are used which are Random Forest, KNN, J48, and BayesNet, then, this data will be modeled and evaluated. Fig. 1 represents the URL classification process.

IV. Experimental Setup

A. Dataset

The dataset used is obtained from [12], [13], for detecting malicious and benign websites by using multiple network and application layers features. The dataset contains 1781 records. Table I presents a description of the 20 features of the dataset.

B. Data Preprocessing

Preprocessing the data is an essential and significant step to improve the machine learning results. A valid and clean data should be used as an input for the algorithms. In this research, we used a tableau, which is a software developed for data visualization [14]. The processing of the data was done by deleting with the noisy data, which causes problems in the performance of the algorithms. In addition, the results are improved by deleting the content length column for containing a lot of null data.

C. Cross-Validation

Cross validation is a technique used to evaluate the machine learning model by dividing the data into two parts, training set which used to learn or train the model, and test set which used to validate the model. In standard cross-validation, both the training and testing sets should cross-over in all the data, that means every data point has a right to be validated.

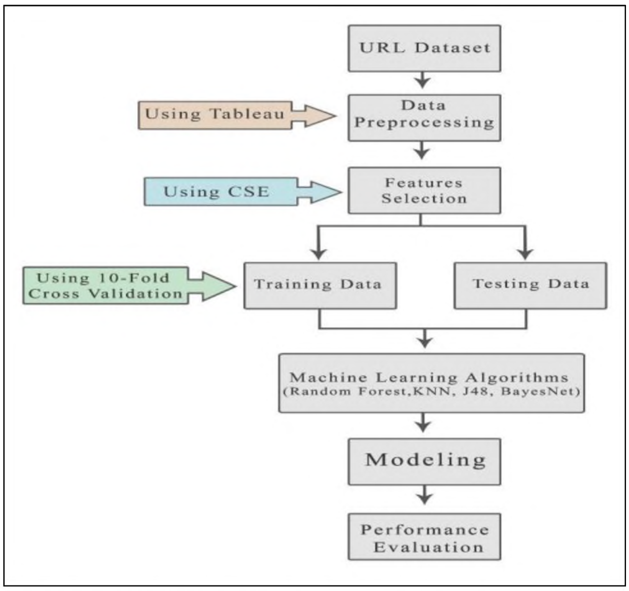

Fig. 1. URL Classification process

TABLE I. DESCRIPTION OF THE FEATURES USED IN THE DATASET

Feature	Description
URL	Unknown identification for URLs used in this study.
URL_LENGTH	Count of characters in the URL.
NUMBER_SPECIAL_ CHARACTERS	Count of special characters in the URL, for example "/", "#", "&".
CHARSET	Standard of character encoding.
SERVER	Type of the server used.
CONTENT_LENGTH	The content volume of the HTTP header.
WHOIS_COUNTRY	The origin country for the server.
WHOIS_STATEPRO	City, state of the server.
WHOIS_REGDATE	The date of begin server registration.
WHOIS_UPDATED_DATE	Date of the last update for the server.
TCP_CONVERSATION_ EXCHANGE	Count of packets exchanged between the server and honeypot client using TCP protocol.
DIST_REMOTE_TCP_PORT	Count of the known ports detected and different from TCP.
REMOTE_IPS	Count of IPs linked to the honeypot.
APP_BYTES	Count of bytes transferred.
SOURCE_APP_PACKETS	The packages sent to the server.
REMOTE_APP_PACKETS	The packages received from the server.
SOURCE_APP_BYTES	Count of bytes transferred to the server.
REMOTE_APP_BYTES	Count of bytes transferred from the server.
APP_PACKETS	The generated IP packets through communication between the honeypot and the server.
DNS_QUERY_TIMES	The generated amount of DNS packets through the communication between the honeypot and server.

In the K-fold cross validation, the main sample is randomly divided into k equal size subsample or folds. Then, from the subsample k, one subsample will be applied as testing data and the remaining will be training data. The cross-validation technique will be repeated several times, and each of the subsample k is used once as data validation [15].

D. Classification Techniques

In the training phase, the data is processed by using four machine learning algorithms. The following are the algorithms that are applied to evaluate the dataset.

- Random Forest

Random Forest (RF) is an ensemble learning algorithm, which can be used for supervised classification and regression. This algorithm operates by merging several decision trees at the training phase and the final decision is choosing by the majority output of decision trees. The Random Forest algorithm reduces randomness to the model when it increases the trees. Random Forest produces a more accurate and constant model [16], [11].

- J48 Decision Tree

J48 decision tree is a developed version of Iterative Dichotomiser 3 algorithm (ID3) [17]. This algorithm contains additional improvements, which are handling with insufficient and missing training values, handling with a continuous attribute value with the discrete values, pruning the decision trees to prevent the over-fitting, and deriving the rules. The J48 decision tree gives the user the chance to make the decision by excluding irrelevant values. In addition, J48 can assist the user to select a suitable target for every new record [17], [18].

- BayesNet

A Bayesian network or Bayes network (BN) belongs to the family of the probabilistic graphical model that represents a set of variables and their conditional dependencies via a Directed Acyclic Graph (DAG). DAG represents a directed acyclic graph and set of a probability distribution, respectively. Both nodes and directed edges are two variables, where random variables are represented by nodes, and direct correlations between variables are represented by edges. A Conditional Probability Table (CPT) is usually used to specify the local distribution. The frequently usage of BNs is for the classification problem. In the classification problem, attribute variables that are collected to predicate class's value represent the set of training examples which are used to construct a classifier [19].

- K Nearest-Neighbor

The K Nearest-Neighbor algorithm (KNN) can be used for both classification and regression. For the classification, this algorithm classifies the objects based on closest training examples. KNN is a kind of instance-based learning also recognized as lazy learning because all the calculations are postponed to the classification process. This algorithm depends on keeping all the training set through the learning and specifies to every query a class by the majority grade for its k-nearest neighbors in the training set. The easiest form of KNN algorithm is when K=1. KNN is considered as one of the simplest classification methods when there is a small or no knowledge about the data distribution [20].

E. Features Selection

Feature selection is the process of electing the important features from the dataset and removing the features that are insignificant with respect to the task that is to be performed. The feature selection is beneficial in reducing the data dimensionality to be handled by classifier, improving prediction accuracy and decreasing execution time [21]. The main goal of this paper is to measure the performance of classifiers with selection of features and without selection. The dataset we used in this paper contains 20 attributes. The attributes are listed in section IV.

The feature selection process is implemented using CfsSubsetEval (CSE). CSE method scales the importance of attributes on the basis of predictive capability of attributes along with its redundancy degree. The preferred subsets have low intercorrelation, yet highly correlated to the target class [22]. We applied CSE attribute evaluator with BestFirst search method. The best 6 attributes resulted from this process that we have taken into account, are the following:

1. *NUMBER_SPECIAL_CHARACTERS*
2. *WHOIS_COUNTRY*
3. *WHOIS_STATEPRO*
4. *DIST_REMOTE_TCP_PORT*
5. *REMOTE_APP_PACKETS*
6. *SOURCE_APP_BYTES*

F. Performance Evaluation

To evaluate the performance of the classification algorithms used, we used performance measures which are the accuracy, recall, and precision. Accuracy is the ratio of results that are classified correctly. Recall is the ratio of positives results which are correctly classified as positive. Precision is the ratio of results classified as positive that are really positive. Equations 1,2, and 3 present the formulas of accuracy, recall, and precision in performance evaluation [23]:

$$Accuracy = TP + TN / TP + TN + FN + FN \quad (1)$$

$$Recall = TP / TP + FN \quad (2)$$

$$Precision = TP / TP + FP \quad (3)$$

We consider the positive class (P) as malicious and the negative class (N) as benign. Where as;

a) TP: True positives, the total number of malicious URLs that are correctly classified.

b) *TN:* True negatives, the total number of benign URLs that are correctly classified.

c) FP: False positives, the total number of benign URLs that are classified as malicious.

d) FN: False negatives, the total number of malicious URLs that are classified as benign.

V. Results Discussion

First, we find the result of the dataset with all features after preprocessing. Second, we find the result of the dataset with selecting features as identified in the previous section. The experiment results show the performance evaluation of the classification algorithm that uses 10-fold cross validation which are Random Forest, J48, BayesNet and KNN.

a) Performance of classifier with all features:

We used the dataset with all features to show the performance of each classifier, Table II lists the values (accuracy, recall, precision and execution time) of each classifier. Among these results, the Random Forest has the highest accuracy, recall and precision compared with other algorithms with values 96%, 95.8% and 96% of accuracy, recall and precision respectively. Also, the KNN has almost similar performance compared with Random Forest.

a) Performance of classifier with selected features:

Instead of using all features, here we use the dataset with selected features, to show the performance of each classifier, Table III lists the values (accuracy, recall, precision and execution time) of each classifier. In this table the Random Forest has the highest accuracy, recall and precision compared with other algorithms with values of accuracy 95%, recall 95.4% and precision 95.3%. The significant of selection features is to reduce the execution time and improve the performance of classifiers as we can see in Table III. Classifiers' accuracy recall and precision have been slightly increased especially in BayesNet, while they have been steady for other algorithms. As well as, execution time decreased for all classifiers.

TABLE II. CLASSIFIER PERFORMANCE OF THE DATASET WITH ALL FEATURES

Classifier	**Accuracy %**	**Recall %**	**Precision %**	**Time Taken s**
Random Forest	96%	95.8%	96%	0.24 seconds
BayesNet	91%	91%	93.6%	0.07 seconds
J48	94%	94.1%	93.9%	0.13 seconds
KNN	95%	95.5%	95.6%	0 seconds

TABLE III. CLASSIFIER PERFORMANCE OF THE DATASET WITH SELECTED FEATURES

Classifier	**Accuracy %**	**Recall %**	**Precision %**	**Time Taken s**
Random Forest	95%	95.4%	95.3%	0.2 seconds
BayesNet	95%	94.5%	94.5%	0 seconds
J48	94%	93.6%	93.3%	0.01 seconds
KNN	94%	93.6%	93.7%	0 seconds

Fig. 2 illustrates the performance of the selected four classifiers. The results of accuracy from Table II and Table III are represented as accuracy1 (blue label) and accuracy2 (red label) respectively. As shown in this figure, the Random Forest has the better accuracy in accuracy1, while BayesNet represents the least. In accuracy2, the Random Forest and BayesNet, have similar accuracy value 95%, they are 1% higher than the other two algorithms. The accuracy obtained in both results specifies that the Random Forest classifier has the best accuracy, recall and precision compared with other algorithms in both cases (Table II & Table III). In the other hand, BayesNet has a better result on selected features, the accuracy increases 4% in Table III than Table II.

Moreover, both the J48 and KNN has same accuracy with selected features, the accuracy they have is 94% as shown in Table III. By contrast, the KNN is 1% higher than J48 as shown in Table II. Also, the KNN is faster than J48 in both cases, with 0 second time taken.

Finally, in both cases, the performance of Random Forest is better than the other algorithms, while K-NN has better performance in [24]. From Table II the accuracy of Random Forest is 3% higher than the accuracy of K-NN reported in [25]. In addition, the recall and precision of Random Forest in our setup are 95.8% and 96% respectively. While in [24] they got recall of 85.05% and precision of 85.25%, of K-NN algorithm.

VI. CONCLUSION AND FUTURE WORK

Nowadays, Web applications have a high importance in our life. On the other hand, cybercriminals are becoming more sophisticated. The basis of the most criminal activities is a malicious Web site. Detection method of the malicious Web sites is one of the various methods that can be used to protect users against those Web sites. This paper focused on using machine learning techniques to detect malicious URLs. In particular, four algorithms were applied to the selected public dataset: Random Forest, KNN, J48 and BayesNet.

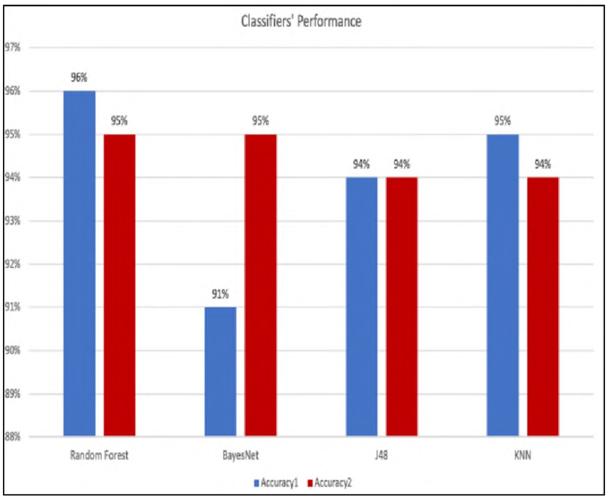

Fig. 2. Performance of the selected four classifiers

Through a controlled experiment, the results show that the execution time of classifiers reduced when features are selected as well as a Random Forest classifier's performance were better than other classifiers in both two phases. This study can be extended by applying other features of the Web pages not only URLs features. Furthermore, other attacks of Web applications can be identified as well as increase the size of the dataset. Finally, the results can be enhanced by using more features or by applying various classification techniques.

REFERENCES

[1] D. Stevanovic, N. Vlajic, and A. An, "Unsupervised Clustering of Web Sessions to Detect Malicious and Non-malicious Website Users," Procedia Comput. Sci., vol. 5, pp. 123 – 131, 2011.

[2] B. Cui, S. He, X. Yao, and P. Shi, "Malicious URL detection with feature extraction based on machine learning," Int. J. High Perform. Comput. Netw., vol. 12, no. 2, p. 166, 2018.

[3] G. Cleary, M. Corpin and O. Cox, et al., "ISTR Internet Security Threat Report Volume 23," Symantec Corporation, CA, USA, 2018.

[4] A. Altaher, "Phishing Websites Classification using Hybrid SVM and

KNN Approach," Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 6, pp. 90 – 94, 2017.

[5] W. Tao, Y. Shunzheng, and X. Bailin, "A Novel Framework for Learning to Detect Malicious Web Pages," in 2010 International Forum on Information Technology and Applications, pp. 353 – 357, 2010.

[6] R. Wang, Y. Zhu, J. Tan, and B. Zhou, "Detection of malicious web pages based on hybrid analysis," J. Inf. Secur. Appl., vol. 35, pp. 68–74, 2017.

[7] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," Expert Syst. Appl., vol. 117, pp. 345–357, 2019.

[8] A. Sirageldin, B. Baharudin, and L. Jung, "Malicious web page detection: A machine learning approach," in Advances in Computer Science and its Applications, Springer, Berlin, Heidelberg, pp. 217–224, 2014.

[9] C. Liu, L. Wang, B. Lang, and Y. Zhou, "Finding effective classifier for malicious URL detection," in Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences - ICMSS 2018, New York: ACM Press, pp. 240 – 244, 2014.

[10] J. Zhao, N. Wang, Q. Ma, and Z. Cheng, "Classifying Malicious URLs Using Gated Recurrent Neural Networks," in Innovative Mobile and Internet Services in Ubiquitous Computing, Springer, pp. 385–394, 2018.

[11] R. Patgiri, H. Katari, R. Kumar, and D. Sharma, "Empirical Study on Malicious URL Detection Using Machine Learning," Distrib. Comput. Internet Technol., Springer, 2019, pp. 380–388.

[12] C. Urcuqui, "Malicious and Benign Websites," 2018. [Online]. Available: https://www.kaggle.com/xwolf12/malicious-and-benign-websites.

[13] C. Urcuqui, A. Navarro, J. Osorio, and M. Garcıa, "Machine Learning Classifiers to Detect Malicious Websites," in CEUR Workshop Proceedings, vol. 1950, pp. 14 – 17, 2017.

[14] D. Murray, Tableau your data! : fast and easy visual analysis with Tableau Software. Weinheim: Wiley, p. 8, 2013.

[15] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-Validation," in Encyclopedia of Database Systems, Boston, MA: Springer US, pp. 532 – 538, 2009.

[16] F. Vanhoenshoven, G. Napoles, R. Falcon, K. Vanhoof, and M. Koppen, "Detecting malicious URLs using machine learning techniques," in 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1 – 8, 2016.

[17] W. Dai and W. Ji, "A MapReduce Implementation of C4.5 Decision Tree Algorithm," Int. J. Database Theory Appl., Vol. 7, No. 1, pp. 49 – 60, 2014.

[18] H. Nguyen and D. Nguyen, "Machine Learning Based Phishing Web Sites Detection," in Recent Advances in Electrical Engineering and Related Sciences, Springer, Cham, pp. 123 – 131, 2016.

[19] J. Su and H. Zhang, "Full Bayesian network classifiers," in Proceedings of the 23rd international conference on Machine learning - ICML '06, pp. 897 – 904, 2006.

[20] S. Imandoust and M. Bolandraftar, "Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background," Int. J. Eng. Res. Appl., Vol. 3, No. 5, pp. 605 – 610, 2013.

[21] P. Kalapatapu, S. Goli, P. Arthum, and A. Malapati, "A Study on Feature Selection and Classification Techniques of Indian Music," Procedia Comput. Sci., Vol. 98, pp. 125 – 131, 2016.

[22] S. Gnanambal, M. Thangaraj, V. Meenatchi, and V. Gayathri, "Classification Algorithms with Attribute Selection: an evaluation study using WEKA," Int. J. Adv. Netw. Appl., Vol. 9, No. 6, pp. 3640 – 3644, 2018.

[23] M. Bramer, Principles of Data Mining, 1st ed. London: Springer, pp. 174 - 177, 2007.

[24] G. Sandag, J. Leopold, and V. Ong, "Klasifikasi Malicious Websites Menggunakan Algoritma K-NN Berdasarkan Application Layers dan Network Characteristics," CogITo Smart J., vol. 4, no. 1, pp. 37 – 44, 2018.