Detection of Malicious Websites Using Machine Learning

Dr.Kasiselvanathan.M,

1. *Assistant Professor Sr.G, Department of Electronics and Communication Engineering, Sri Ramakrishna Engineering College, Coimbatore-641022, kasiselvanathan.m@srec.ac.in*

Sathiyapriya.A.M

2. *, Department of Electronics and Communication Engineering, Sri Ramakrishna Engineering College, Coimbatore-641022, sathiyapriya.1702207@srec.ac.in*

Suruthimathi.C,

3. *Department of Electronics and Communication Engineering, Sri Ramakrishna Engineering College, Coimbatore-641022, suruthimathi.1702231@srec.ac.in*

Suryamitra.P

4. *Department of Electronics and Communication Engineering, Sri Ramakrishna Engineering College, Coimbatore-641022, suryamitra.1702232@srec.ac.in*

Abstract

In Modern Internet world, security of private information is nightmare to every person due to the development of Web Apps, Mob Apps, etc., There are several ways for scammers to steal information from internet users, including web surfing, spam mail, and social media apps. The internet has become a forum for a wide variety of illicit activities, ranging from spam advertisements to financial fraud, thanks to technological advancements. Malicious websites contribute significantly to the growth of online illegal activity and stifle the advancement of Web services. As a result, there has been a significant push to create a systematic solution to prevent users from accessing such websites. Phishing is a malicious practice that involves inducing people to disclose personal details such as passwords and credit card numbers. It is one of the most common and least-protected security threats. Phishing attacks are dangerous threats that use human contact to persuade people to reveal sensitive information or take inappropriate acts. To recognize URLs, the proposed system only uses six features. The number of hyphens, dots, numeric characters, discrete variables that refer to the presence of an IP address in the URL, and the similarity index are all features of the URL. We suggest a learning-based method for categorizing Web sites into three categories: benign (safe), spam, and malicious. Our scheme achieves this by using machine learning algorithms such as the SVM Algorithm, MLP classifier, and Random Forest Algorithm.

Keywords:*URL, Phishing attacks, Machine learning, SVM Algorithm, MLP classifier, Random forest Algorithm, Security, Accuracy*

I. INTRODUCTION

The internet has evolved into a forum for a wide variety of illicit activities, from spam advertisements to financial fraud, thanks to technological advancements. Malware programmes are embedded in URLs to carry out some of these operations. Users must be mindful that there is a negative side to the system. One of the dangers is that people may be vulnerable to online fraud by phishing while they are online. Heuristic techniques are used to detect phishing URLs, phishing emails, and phishing websites due to the fact that the key data entry points are typically a masqueraded URL (or link) It serves as a strong motivator for this study's proposal of a broad-based solution for detecting fake URLs. URL analysis is the most promising technique because it has fewer limitations than other approaches, especially those that rely solely on lexical analysis because lexical features are extracted directly from URLs. As a result of these studies, combining machine learning with URL lexical features will result in phishing detection systems that are both accurate and lightweight.

II. LITERATURE SURVEY

For the identification of malicious websites, a one-of-a-kind literature survey was carried out. For the best results, researchers looked at a variety of literature surveys from various journals and conference papers.

A customised whitelist method for phishing webpage detection(Authors: Belabed, E. Ameur) The number of phishing attacks against web services has steadily increased, posing a threat to banking and financial institutions' ability to provide secure online services, for example. This paper describes a method for automatically detecting phishing attacks. A customised whitelisting approach is combined with machine learning techniques in our approach. The whitelist is a filter that prevents phish web pages from imitating harmless user actions. The phishing pages that are not blocked by the whitelist pass are further filtered using a Support Vector Machine classifier that has been specifically developed and optimised for classifying these risks. The proposed method outperforms existing state-of-the-art approaches, according to our findings.

An Efficacious Method for Detecting Phishing Webpage Through Target Domain Identification (Authors: R.Gowthama, Dr.IlangoKrishnamurthib, K.SampathSree Kumara)
Phishing is a deceptive method of obtaining confidential information from unsuspecting users by impersonating a reliable entity in an electronic transaction. To deceive the victims, a variety of methods are used, including spoofed e-mails, DNS spoofing, and chat rooms with links to phishing websites. Despite the numerous anti-phishing solutions available, phishers continue to entice victims. We present a novel approach in this paper that not only overcomes many of the challenges associated with detecting phishing websites, but also recognises the phishing target that is being imitated. We've developed an anti-phishing technique that groups domains based on hyper links that have a direct or indirect connection to a suspicious webpage. To arrive at a target domain collection, domains collected from directly related webpages are compared to domains collected from indirectly related webpages. On this package, we zero-in the target domain by using the Target Identification(TID) algorithm. The suspicious domain and the target domain are then looked up using a third-party DNS service, and the authenticity of the suspicious page is determined by a comparison.

The papers listed above discuss different methods for locating malicious websites. Email metadata is used in most automated phishing email detection methods. Naive Bayes, KNN Classifier, and Linear Regression are some of the machine learning methods used. In comparison to those articles, the proposed model performs better and is more accurate.

III. METHODOLOGY

In this section, we'll go through the architecture of our system as well as the malicious URL detection implementation info. On a wide scale, we use Machine Learning to assist in the detection of malicious websites.

Machine Learning is a branch of Artificial Intelligence that is focused on the concept of giving machines access to data and allowing them to learn and experiment on their own. It is concerned with extracting patterns from massive data sets. Machine Learning allows a machine to learn from data, enhance output based on past experiences, and predict outcomes without having to be specifically programmed. The learning process starts with observations or data, such as examples, direct experience, or instruction, so that we can search for trends in data and make informed decisions in the future based on the examples we have. The primary goal is for computers to learn on their own, without the need for human interference, and to adapt their behaviour accordingly. Machine learning is needed because it is capable of performing tasks that are too complex for a human to perform directly. A type of machine learning method called supervised learning is used. which feeds labelled data to the machine learning system in order to train it, and then predicts the outcome based on that. In supervised learning, the aim is to map input data to output data. Supervised learning is dependent on instruction, and it is similar to when a student learns under the guidance of an instructor. There are two types of algorithms that can be used in supervised learning:

1. Regression
2. Classification

Python Version 3 is the programme we used for this project. It's a programming language that's free to use. Python was designed to be both simple and effective to read. Python is a scripting language that runs on the interpreter. Compilation is not needed to run interpreted languages. On almost any machine, a programme known as an interpreter executes Python code. This means a programmer can make changes to the code and see the results almost immediately. Since Python does not run machine code directly, it is slower than a compiled language like C. Python is an excellent first programming language. Since it is a high-level language, a programmer may concentrate on what needs to be done rather than how it should be done. Python takes less time to write programmes than certain other languages. Many functions are included with Python when it is loaded in its regular library. There are several other libraries available on the Internet that extend the capabilities of the Python programming language. These libraries render it a versatile language that can accomplish a wide range of tasks. On December 3, 2008, Python 3.0 (also known as "Python 3000" or "Py3K") was announced. It was created to address fundamental design deficiencies in the language; however, the necessary improvements could not be made while maintaining complete backwards compatibility with the 2.x series, necessitating the creation of a new major version number.

The guiding principle of Python 3 was to reduce feature duplication by removing old ways of doing things. Some of the library modules in Python includes:

1. NUMPY
2. PANDAS
3. MATPLOTLIB
4. SKLEARN

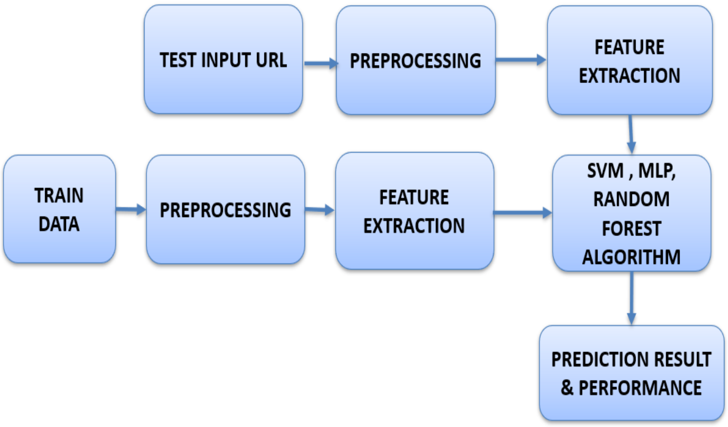


Fig. 1. Block Diagram

The figure 1 shows the block diagram of the proposed model. Each module description is given as follows:

A. *Testing Data*

Data that has been explicitly defined for use in experiments, usually of a computer programme, is referred to as test data. Some data may be used in a confirmatory manner, for example, to ensure that a given collection of inputs to a function produces the expected result.

B. *Pre-processing*

Cleaning, instance collection, normalisation, transformation, feature extraction, and selection are all examples of data preprocessing techniques. The process of finding, fixing, or deleting inaccurate information from data is known as data cleaning. The process of data normalisation involves converting a set of independent variables or data features into [0, 1] or [-1, +1] values. The process of transforming data from one format to another that people expect is known as data transformation. The final training set is the result of data preprocessing.

C. *Feature Extraction*

The process of converting input data into a collection of features that can accurately represent the input data is known as feature extraction. From the URL string, we extracted 63 URL attributes.

D. *Training Data*

The training data would be a critical component of any implementation. This is the information that the model iterates over and improves to enhance its accuracy.

E. *SVM Algorithm*

SVM (Support Vector Machine) is a common Supervised Learning algorithm for Classification and Regression. However, it is most commonly used in Machine Learning for Classification problems. The SVM algorithm's aim is to find the best line or decision boundary that can divide n-dimensional space into classes so that new data points can be conveniently placed in the correct category in the future. A hyperplane denotes the strongest judgement boundary. SVM selects the hyperplane-helping extreme points/vectors. Support vectors are the extreme cases, and the Support Vector Machine algorithm is named after them. In the battle against phishing, detection is extremely critical. A browser-based technique is one of them. The phishing URL classification scheme, which is focused solely on the examination of the suspicious URL, will help the end user avoid unwelcome events. A new method for detecting phishing URLs based on SVM is proposed in this report. The use of the Hamming distance as one of our system's input characteristics increased the recognition rate by 21.8 percent in tests using the SVM process. The boundary of the data must be classified between the planes. Using the values -1 and 1 from the boundary 0, we can predict whether the given URL is phishing or not. The aim of SVM is to find the best separating hyperplane for the training data that maximises the margin between different groups.

F. *MLP Classifier*

MLP Classifier stands for Multi-layer Perceptron Classifier, which is connected to a Neural Network by its name. A feed forward artificial neural network with multiple layers is known as a multilayer perceptron. The term MLP is ambiguous; it can refer to any feedforward ANN, or it can refer specifically to networks made up of multiple layers of perceptrons. Multilayer perceptrons, particularly those with a single hidden layer, are often referred to as "vanilla" neural networks. There are at least three layers of nodes in an MLP: an input layer, a hidden layer, and an output layer. Each node, with the exception of the input nodes, is a neuron with a nonlinear activation function. For preparation, MLP employs backpropagation, a supervised learning technique. MLP differs from a linear perceptron in that it has several layers and activation that is non-linear. It can tell the difference between data that isn't linearly separable and data that is linearly separable.

G. RandomForest Algorithm

Random Forest is a well-known supervised machine learning algorithm. In machine learning, it can be used for both classification and regression. It is based on ensemble learning, which is the process of combining multiple classifiers to solve a complex problem and improve the model's accuracy. Random Forest, as the name implies, is a classifier that combines a number of decision trees on different subsets of a dataset and averages their results to increase the dataset's predictive accuracy. The more trees in the forest, the more accurate it is and the problem of overfitting is avoided. Banking, medicine, land use, and marketing are the four key sectors where Random forest is most commonly used. Random Forest is capable of classifying and predicting data. It can handle massive datasets with a lot of variables. It improves the model's accuracy and eliminates the problem of overfitting.

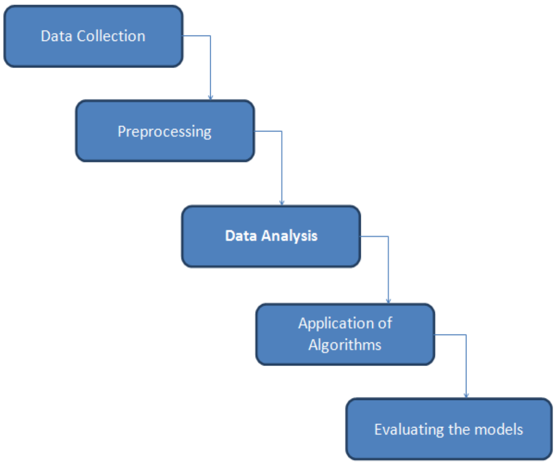


Fig. 2. Flow Diagram

The figure 2 shows the flow diagram of the project. Various steps are given as follows:

1. Data Collection: The data consists of legal as well as phishing websites. Each website with various set of features are collected.
2. Pre-processing: The data gets transformed to bring it to a state that machine can be easily interpreted by the algorithm.
3. Data Analysis: Data collected and preprocessed are analyzed and the features are extracted.
4. Application of Algorithms: SVM Algorithm, MLP classifier and Random Forest Algorithm are applied.

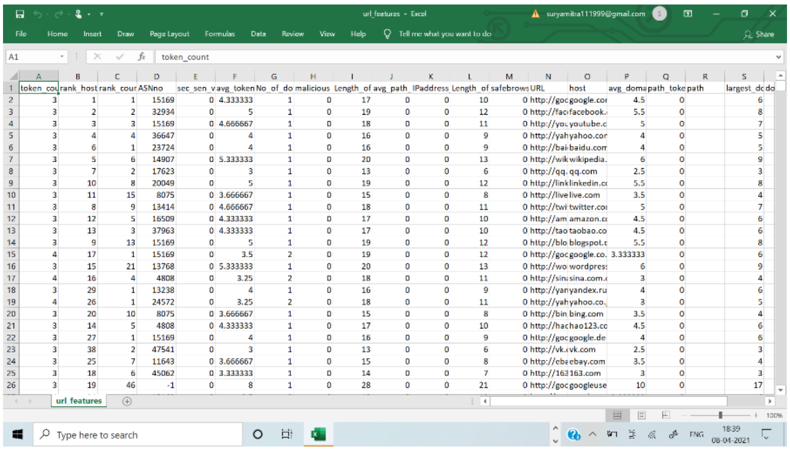
Evaluating the models: After the application of Algorithms, the results are predicted.

IV. EXPERIMENTAL RESULTS

In this chapter the experimental results are shown. The model is created in Python version 3 software platform.

URLs of the websites are separated into 3 classes:

- Benign: Safe websites with normal services
- Spam: Website performs the act of attempting to flood the user with advertising or sites such as fake surveys and online dating etc.
- Malware: Website created by attackers to disrupt computer operation, gather sensitive information, or gain access to private computer systems.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1	token_cou	rank_host	rank_cour	ASNno	sec_sen_v	avg_token	No_of_do	malicious	Length_of	avg_path_	IPaddress	Length_of	safebrows	URL	host	avg_doma	path_toke	path	largest_dc	do
2	3	1	1	15169	0	4.333333	1	0	17	0	0	10	0	http://goc	google.cor	4.5	0		6	
3	3	2	2	32934	0	5	1	0	19	0	0	12	0	http://fac	facebook.	5.5	0		8	
4	3	3	3	15169	0	4.666667	1	0	18	0	0	11	0	http://you	youtube.c	5	0		7	
5	3	4	4	36647	0	4	1	0	16	0	0	9	0	http://yah	yahoo.cor	4	0		5	
6	3	6	1	23724	0	4	1	0	16	0	0	9	0	http://bai	baidu.com	4	0		5	
7	3	5	6	14907	0	5.333333	1	0	20	0	0	13	0	http://wik	wikipedia.	6	0		9	
8	3	7	2	17623	0	3	1	0	13	0	0	6	0	http://qq.	qq.com	2.5	0		3	
9	3	10	8	20049	0	5	1	0	19	0	0	12	0	http://link	linkedin.cc	5.5	0		8	
10	3	11	15	8075	0	3.666667	1	0	15	0	0	8	0	http://live	live.com	3.5	0		4	
11	3	8	9	13414	0	4.666667	1	0	18	0	0	11	0	http://twi	twitter.cor	5	0		7	
12	3	12	5	16509	0	4.333333	1	0	17	0	0	10	0	http://am	amazon.cc	4.5	0		6	
13	3	13	3	37963	0	4.333333	1	0	17	0	0	10	0	http://tao	taobao.co	4.5	0		6	
14	3	9	13	15169	0	5	1	0	19	0	0	12	0	http://blo	blogspot.c	5.5	0		8	
15	4	17	1	15169	0	3.5	2	0	19	0	0	12	0	http://goc	google.co.	3.333333	0		6	
16	3	15	21	13768	0	5.333333	1	0	20	0	0	13	0	http://wo	wordpress	6	0		9	
17	4	16	4	4808	0	3.25	2	0	18	0	0	11	0	http://sin	sina.com.c	3	0		4	
18	3	29	1	13238	0	4	1	0	16	0	0	9	0	http://yan	yandex.ru	4	0		6	
19	4	26	1	24572	0	3.25	2	0	18	0	0	11	0	http://yah	yahoo.co.j	3	0		5	
20	3	20	10	8075	0	3.666667	1	0	15	0	0	8	0	http://bin	bing.com	3.5	0		4	
21	3	14	5	4808	0	4.333333	1	0	17	0	0	10	0	http://hac	hao123.cc	4.5	0		6	
22	3	27	1	15169	0	4	1	0	16	0	0	9	0	http://goc	google.de	4	0		6	
23	3	38	2	47541	0	3	1	0	13	0	0	6	0	http://vk.c	vk.com	2.5	0		3	
24	3	25	7	11643	0	3.666667	1	0	15	0	0	8	0	http://eba	ebay.com	3.5	0		4	
25	3	18	6	45062	0	3.333333	1	0	14	0	0	7	0	http://163	163.com	3	0		3	
26	3	19	46	-1	0	8	1	0	28	0	0	21	0	http://goc	googleuse	10	0		17	

Fig. 3. Datasets Collected

The set of URLs is shown in Figure 3. A confusion matrix is a table that shows how well a classification model performs on a collection of test data for which the true values are known. Although the confusion matrix is easy to comprehend, the words used to describe it can be perplexing.

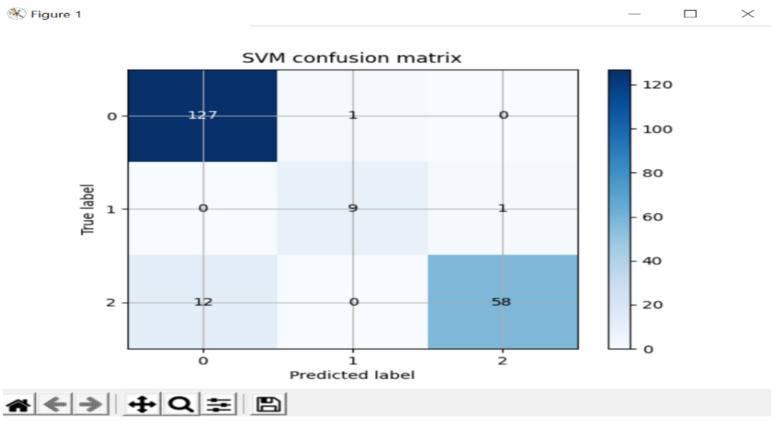


Fig. 4. Confusion Matrix for SVM

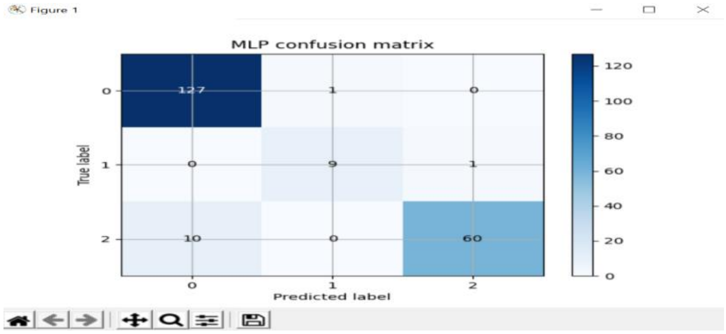


Fig. 5. Confusion Matrix for MLP

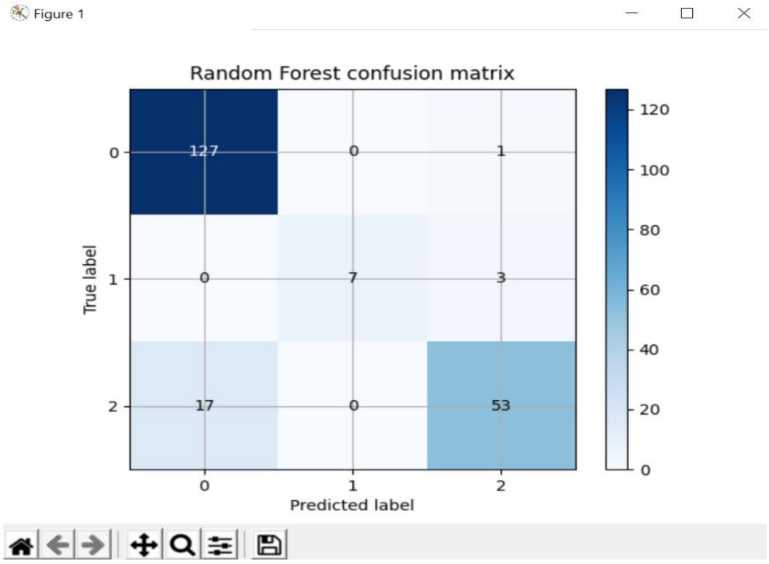


Fig. 6. Confusion Matrix for Random Forest

Figures 4, 5, 6 depicts the confusion matrix for SVM Algorithm, MLP Classifier and Random Forest Algorithm respectively.

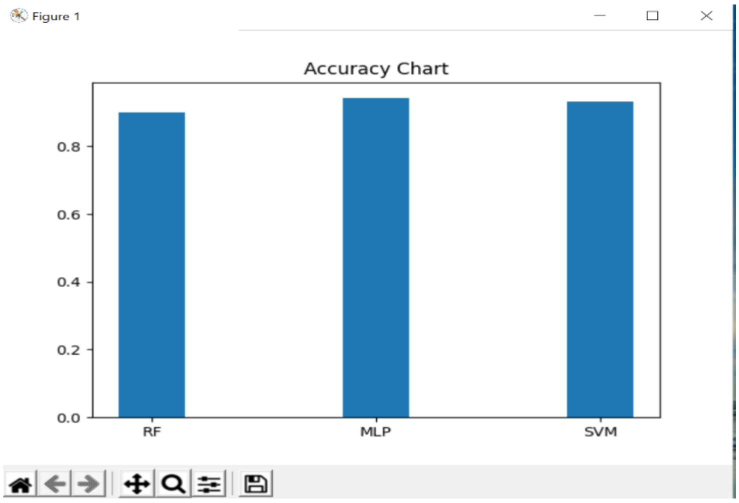


Fig. 7. Accuracy Chart

Based on the comparisions between the three Algorithms, the Accuracy chart is plotted and it is shown in Figure 7.

- The Accuracy of SVM Algorithm is 93.2%
- The Accuracy of MLP Classifier is 94.2%
- The Accuracy of Random Forest Algorithm is 89.9%

After the comparisions, the Best Accuaracy and performance is achieved by MLP Classifier.

V. CONCLUSION

In this project, we demonstrated how to use Machine Learning to identify malicious websites. The results of three different Algorithms are compared. Phishing is a major issue that leads to identity theft. Phishing attacks, despite their simplicity, are extremely powerful and have caused billions of dollars in harm in recent years. In certain cases, the phisher does not cause the economic harm directly, but instead resells the illegally acquired information on the secondary market. As a result, phishing attacks are still prevalent, and solutions to the issues are needed. We investigate the structure of URLs, lexical features in URL characters, and the phishing goal brand name in this report, and suggest an MLP-based phishing URL detection solution. The results of the experiments show that the solution is good at catching phishing URLs and can be used as a browser plug-in to filter phishing sites. In the future, we will build an AI-based protection framework to detect malicious URLs.

References

[1] Vinayakumar R, Mamoun Alazab, Soman Kp, Prabaharan Poornachandran, Ameer Al-Nemrat, and Sitalakshmi Venkatraman, “Deep Learning Approach for Intelligent Intrusion Detection System”, IEEE Access, April 2019.

[2] Akshay Sushena Manjeri, Kaushik R, Ajay MNV, Priyanka C. Nair, “A Machine Learning Approach for Detecting Malicious Websites using URL Features”, IEEE Conference on Electronics Communication and Aerospace Technology, 2019.

[3] Frank Vanhoenshoven, Gonzalo Napoles, Rafael Falcon, Koen Vanhoof and Mario Koppen, “Detecting Malicious URLs using Machine Learning Techniques”, IEEE, 2019.

[4] Xiaodan Yan, Yang XuB, Member, Baojiang Cui, Shuhan Zhang, Taibiao Guo, and Chaoliang Li, “Learning URL Embedding for Malicious Website Detection”, IEEE Transactions on Industrial Informatics, 2019.

[5] Guolin Tan, Peng Zhang, Qingyun Liu, Xinran Liu, Chunge Zhu, Fenghu Dou, “Adaptive Malicious URL Detection: Learning in the Presence of Concept Drifts”, IEEE International Conference On Trust, Security And Privacy In Computing And Communications, 2018.

[6] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang,``An empirical analysis of phishing blacklists,'' in *Proc. 6th Conf. Email Anti-Spam (CEAS),* Mountain View, CA, USA, Jul. 2009, pp. 1_20.VOLUME 7, 2019 73283

[7] J. Kang and D. Lee, ``Advanced white list approach for preventing access to phishing sites,'' in Proc. Int. Conf. Converg. Inf. Technol. (ICCIT), Gyeongju, South Korea, Nov. 2007, pp. 491_496.

[8] M. Shari_ and S. Siadati, ``A phishing sites blacklist generator,'' in *Proc.*6th ACS/IEEE Int. Conf. Comput. Syst. Appl. (AICCSA), Doha, Qatar,Mar./Apr. 2008, pp. 840_843.

[9] X. Han, N. Kheir, and D. Balzarotti, ``PhishEye: Live monitoring of sandboxed phishing kits,'' in Proc. 23rd ACM Conf. Comput. Commun. *Secur. (CCS)*, Vienna, Austria, Oct. 2016, pp. 1402_1413.