Malicious URL Detection by using Ensemble Learning

*

1st Shivangi Pachauri
Department of Mathematics
Chandigarh University, Gharuan
Punjab, India
pachaurishivangi8@gmail.com

2nd Neha Dhariwal
Dept. of Computer science Engineering
Chandigarh University, Gharuan
Punjab,India
nehadhariwal12.nd@gmail.com

3nd Gagandeep Marken
Dept. of Computer Science Engineering
Chandigarh University, Gharuan
Punjab, India
Gaganmarken1990@live.com

***Abstract*—The most fundamental requirement these days is the internet. Additionally, we must loop through a number of URLs in order to retrieve any type of information (Uniform Resource Locators)[1]. Therefore, it is crucial to confirm that those URLs are safe and won't harm the computer.Advances in cloud and Internet technology have contributed to a notable rise in electronic trade in recent years, where customers conduct transactions and purchases online. This expansion harms an enterprise's resources by allowing unauthorised access to sensitive user data[3]. any manner,Malicious URL detection is a difficult yet interesting issue. Scammers mostly create URLs by implementing incredibly complex adjustments, and researchers must identify them while keeping in mind how the produced URLs behave[1]. There are several techniques for phishing detection in the anti-malware space, while URL-based schemes are more secure and more practical for two reasons: zero-hour detection capabilities and the elimination of the need to visit rogue websites. Hybrid ensemble-based machine learning technology is the foundation of this work.**

subsectionKeywords Cyberattacks, Machine Learning, Supervised Learning, Ensemble Learning, Malicious URL

I. Introduction

The Internet has evolved into a household requirement rather than only a luxury or a need. You may reach locations and stay in touch with loved ones without physically travelling thanks to the internet[2].In the modern world, where things move quickly, technology has become a need for everyone. With the rapid advancement of technology, our experiences have become more comfortable.

You may easily discover information about anything by searching for it on Google, Facebook, Twitter, or by entering the topic's name into the search bar or browser window. When using social search, you frequently don't even need to type your whole query before finding what you're looking for. A web address becomes a URL each time you click on a link on a website or enter it into your browser[6]. The Uniform Resource Locator is known as URL[1]. Consider it as a street address, with each segment of the URL standing in for a distinct portion of the address and offering a separate piece of information. What is the value of a URL, then? URLs are becoming less valuable than what is done with them. Developing a brand that functions across many social media channels and has the ability to engage prospective clients is as, if not more, crucial. To make it as strong as it looks in, a lot of money and effort may be required. Rich material, internal and external links, interactive elements, and a contemporary style that functions well on a variety of devices and platforms are all necessary for websites[7].

Ensemble Learning Analysis.

II. Background

The number of hazards related to network information security is rising quickly nowadays. Today's hackers mostly target end-to-end technology and take advantage of weaknesses in people. Social engineering, phishing, pharming, and other methods are among them[9]. As a result, finding malicious URLs has drawn a lot of attention lately. Numerous scientific research using machine learning and deep learning approaches provide several methods for identifying dangerous URLs. This paper suggests a machine learning approach for identifying harmful URLs by analysing the characteristics and behaviour of the suggested URLs[8]. Furthermore, the use of ensemble learning is employed to improve the detection of dangerous URLs by identifying their unusual behaviour. To put it briefly, the machine learning algorithms and new set of URL attributes and behaviours make up the detection system that is being offered by. The suggested URL properties and behaviours significantly enhance our capacity to identify fraudulent URLs, according to test findings. This implies that our suggested technique might be seen as a simplified and approachable malicious URL detection tool. Hackers of today generally target end-to-end technology and exploit human vulnerabilities[4]. Among them include pharming, social engineering, phishing, and other techniques. Therefore, there has been a lot of attention lately in identifying rogue URLs.

III. URL

URLs (Uniform Resource Locators) are used to refer to resources on the Internet[5]. It is nothing but the address

of a particular unique resource on the web. In theory, every valid URL points to a unique resource. Such resources include HTML pages, CSS documents, images, and so on. There are actually some exceptions, but the most common are URLs pointing to non-existent or moved resources.The resource represented by the URL and the URL itself are managed by the web server, so the web server owner should carefully manage this resource and the URLs associated with it[11].

IV. Malicious URL

A malicious URL is a false link created intentionally to spread assaults, frauds, and scams. You can download viruses, trojan horses, ransomware, and other malware that can harm your computer or even your network by clicking on an infected URL[15]. We can also be tricked into divulging critical information on phoney websites via malicious URLs[7]. This is why specialists refer to what a lot of people refer to as "malicious URLs" instead of "viral links," "infected links," or just "armed links."

V. Methods of detecting Malicious URL

Through social engineering operations, attackers and script kiddies have a favourite vector. This is due to the fact that the average user still follows every link or goes to every URL. One fundamental and important method of implementing some sort of primary level security is to blacklist specific URLs. But it's also possible to utilise machine learning to determine if a URL is dangerous. Therefore, there are essentially two methods by which we may identify malicious URLs:

1) Traditional Blacklisting Method: Every time a new URL is viewed, a database query is run. A malicious URL will be marked as blacklisted, and an alert will be sent out. If not, URLs are regarded as secure. This method's primary flaw is that it makes it exceedingly challenging to identify new malicious URLs that are not included in a list. We have to update the database every time a new URL is created. Every day, a large number of phoney URLs are created; if the URL is not on the blacklist, it is quite likely to be regarded as secure.

2) Traditional ML Algorithms: Investigations were conducted into many harmful URL schemes that used machine learning techniques. Among these machine learning algorithms are Decision Trees, Naive Bayes, SVM, and Logistic Regression, among others[8]. The precision of these algorithms using various Experimental findings are used to illustrate parameter settings.

A. Hybrid Ensemble Model

Hybrid ensemble learning makes use of the advantages of each component by combining several base model or algorithm types in an ensemble. It leverages the variety of various learning algorithms to increase overall prediction performance and resilience[2]. A machine learning concept known as "ensemble learning" involves using the aggregate strength of machine learning models to a learning task, such as a regression or classification problem. This method groups together a number of homogenous machine learning models that are assumed to be poor learners. When the classes predicted by the weak learners become the final class predictions of the ensemble model, the Max Voting Classifier approach is mostly utilised. All URLs in this document have been categorised into four main groups, which are explained below:

1) Benign URL: The right URL that displays the search results a person wants to see on the internet is called a benign URL[11]. It alone fulfils the function for which the URL was created and has no negative effects on the machine.

2) Phishing URL: Phishing attacks can take many different forms, but they usually begin with an email. Attacks using malicious URLs take phishing to a new level.

3) Malware URL: A link made with the intention of aiding in a scam, attack, or fraud is known as a malware URL[9]. By clicking on malicious URLs, you risk infecting your computer or possibly your computer network with viruses, Trojan horses, ransomware, and other malware.

4) Defacement URL: An assault known as "web defacement" occurs when a malevolent party breaches into a website.

and adds a message of his own in lieu of the information on the page. remarks may suggest that the website has been hacked by a certain hacker gang, contain offensive or political remarks, or contain other improper information that embarrasses the site owner.

VI. Accomplishment

A.

Softwares and Tools use:
The project was completed using the following tools.

1) Shodan API: For gadgets with an Internet connection, Shodan is a search engine. Every device that is directly linked to the internet is monitored by Shodan. When a device is linked to the Internet directly[12], Shodan asks it for different publically accessible data. This API was utilised in this project for retrieving all URLs' host-based information.

2) Kaggle: One well-known website for data science and machine learning competitions is Kaggle. It gives academics, machine learning engineers, and data scientists a place to work together on projects[13], take part in different data-driven challenges, and demonstrate their abilities. Kaggle provides a configurable, no-setup environment for Jupyter Notebooks. Kaggle is ideal for machine reading, data analysis, and teaching and allows anybody to develop and execute improper Python code via a browser.

B. Data Analysis

The process of examining, purifying, converting, and analysing data in order to find relevant information, make inferences, and aid in decision-making is known as data analysis.

6, 51, 191 URLs make up the dataset used for training and testing, of which 4,28,103 are safe or benign URLs, 96,457 are defacement URLs, 94,111 are phishing URLs, and 32,520 are malware URLs. It includes two columns: one for the URL and another for the type, which denotes the maliciousness class[15].

VII. CODEBASE

The following modules and libraries were utilised to carry out the project:

1) Pandas: It is a well-known Python module that is used for both data manipulation and data analysis. It as a well-liked open-source Python package for data analysis and manipulation is called pandas. In addition to offering data analysis tools for activities like cleaning, converting, and studying information, it offers high-performance, user-friendly data structures.

2) numpy: "Numerical Python" is the name of the Python package. It is a library made up of several array processing algorithms and multidimensional array objects. To work effectively on various data structures, it offers support for matrices, arrays, and other mathematical operations. For activities involving numerical computations in disciplines like data science, science, engineering, and mathematics, NumPy is an essential library.

3) seaborn: This Python package is mostly utilised for creating statistical visuals. It is based on Matplotlib, which offers a high-level interface for making statistical visualisations that are both aesthetically pleasing and educational. It works especially well for visualising complex datasets and simplifying the interpretation of changing connections.

4) re: It's a Python package that's mostly used for regular expression manipulation.It is an integrated module that supports regular expressions, which are effective tools for text editing and pattern matching.The "re" package allows us to do a number of tasks, including pattern-based replacements, information extraction from strings, and text search for patterns.

5) Matplotlib: A complete Python visualisation toolkit for static, animated, and interactive graphics is called Matplotlib.

6) urllib: Python's URL handling module is called Urllib. It's employed for fetching Uniform Resource Locators, or URLs.

7) tensorflow: Experts and novices alike can create machine learning models for the web, cloud, mobile, and desktop with ease thanks to TensorFlow.

8) sklearn: Several potent methods for statistical modelling and machine learning, such as dimensionality reduction, regression, clustering, and classification, are included in the sklearn package.

9) tld: With the help of Artur Barseghyan's tld Python package, we can quickly extract the top level domain (TLD) from a given URL. In the Internet's hierarchical Domain Name System (DNS), it is the highest level. The portion of a domain name that comes after the last dot, such as.com,.org, or.net, is known as the top-level domain.

VIII. PROCESS FLOW DIAGRAM

It is a visual representation of Process that contain series of activities.To represent the processes or actions involved in finishing a task, addressing an issue, or explaining a workflow, it is made up of various shapes, arrows, and labels.
The sequence of actions can be shown as follows.

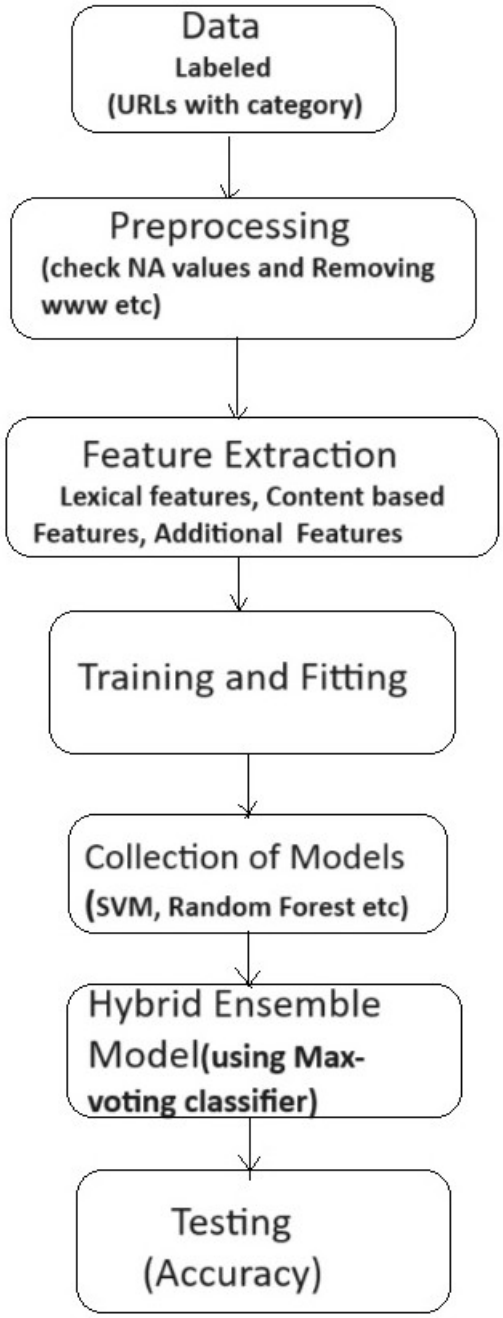

Fig. 1. Process-flow Block Diagram

IX. Importing the dataset from CSV file

The URL itself and its categorization (namely benign, malware, defacement, phishing) are the two attributes shared by each record.

A. Extracting info of the complete data

Extract the Information From the Data using Python code.It is the process to define relevent information from the Dataset whatever you use for the Model.

B. Checking if any NaN value exist

We check if there any Null Value or missing value. Here NaN refers the Not a Number. In python we might use isna() function for find the missing values.

C. obtaining the number of URLs in each category and graphing them

In this, analysing a dataset to find the number of online links (URLs) that fall into certain groupings or categories, and then displaying the results in a graph or chart.

D. leaving out the (www.) in the URL, which is actually a subdomain on its own

A subdomain is a section of a primary domain name that is usually utilised for website organisation and navigation. It is situated between the primary domain name and the leftmost dot[12]. "www" is the subdomain in the URL "www.example.com". Subdomains are frequently used to separate sections or areas of a website, each of which may have unique services or content. We are effectively visiting the main domain directly if we omit the "www." from a URL. The website remains the same even if we type "example.com" (without the "www") into our browser; however, any customised subdomains (such "blog.example.com" or "shop.example.com") that may have been created are not accessed. This is standard procedure, and a lot of websites are set up to function both with and without the "www." prefix. Although "www" is a popular subdomain for web servers, it's important to remember that there are other options as well. Subdomains can be added to websites for a variety of uses[17].

E. Assigning numerical value for each category type for the purpose of model training

"Encoding" or "labelling" categorical data is the process of giving numerical values to categorical variables. Since the majority of machine learning methods require numerical inputs, this is frequently done while preparing data for machine learning models.

X. Feature Extraction

A URL's retrieved features serve as the foundation for identifying whether or not the URL is malicious. The grade of our training set and the attributes that go into our model determine how well our learning endeavour performs.

A. Lexical Features

Frequently, malicious URLs may be identified using lexical factors like domain length, URL length, and the quantity of special characters. Some phishing URLs are created by merely altering benign URLs to look real [4].

The features listed below were taken out for lexical analysis:

• Length of URL; • Length of host name; • If IP host is used for URL; • If port number is assigned to URL

• The quantity of subdirectories; • The number of digits; • The number of parameters; • If the URL comprises the term "admin"; • If the URL is encoded (contains

1) Method definitions for various lexical feature extraction:

2) Inserting the above attributes into the data field. :

B. Content-based Features

Content-based features are those that are classified based on the HTML content of the URL. Among other things, it contains the number of times certain HTML tags are processed, white spaces, and integrated hyperlinks. The content analysis elements that were extracted included: page entropy; number of script tags; HTML length; script to body ratio; punctuation; hidden tags; number of iframes; number of hyperlinks; number of whitespaces.

1) Definitions of methods for extracting different content-based features : In natural language processing (NLP), features that are based on the content itself as opposed to its structure or grammatical aspects are known as content-based features.

2) Inserting the data into the data frame of each URL: You may use Python's pandas package to put data into a DataFrame.

C. Additional Features

We have enhanced the model's accuracy by adding a few more characteristics.

1) Checks to see whether URL contains a shortening service: Regular expressions may be used to search for common patterns in URLs that are suggestive of URL shorteners, which can be used to determine whether a given URL comprises a shortening service. Remember that there are a variety of URL shortening providers available, and that their patterns might change, so this approach might not always work.

2) values are added to the data frame. : Now we add the values to the Dataframe by using various function in Python code. For example, df.head() and df.tail() used for viewing the data.df[Column] used for access the Column from dataframe, df.describe() used for summerize the data.

3) The Shortening Service count plotted : Use a bar chart or any other suitable visualisation to plot the number of URLs that incorporate a shortening service. Each URL is first subjected to the contains shortening service function, and the outcome is then stored in a new column named "Shortened URL." Next, we determine how many URLs are shortened and not truncated.

XI. Visualisation of Data

Data visualisation allows us to view the appearance of our data as well as the correlations between its features. It's the simplest method of determining if the qualities and output match[18].

XII. Train and test split

1) dividing characteristics for input and output:

XIII. Training Models

The following machine learning methods were applied to train the model: Gaussian Naive Bayes, Random Forest, K-Nearest Neighbour, Support Vector Machine, and Decision Tree approaches.

1) Classification Report of Random Forest Classifier: An ensemble learning technique for problems like regression and classification is called Random Forest. During training, a large number of decision trees are built, and at the end, the class that represents the mean prediction (regression) or class mode (classification) of each individual tree is produced. A classification report offers a thorough assessment of the performance of a classification model, taking into account parameters such as support for each class, accuracy, recall, and F1-score.

2) Classification Report of Support Vector Machine Classifier: A potent supervised machine learning approach for both regression and classification applications is called Support Vector Machine (SVM). It operates by locating the best hyperplane in the feature space to divide classes.

3) Classification Report of K-Nearest Neighbour: A flexible and straightforward supervised machine learning approach, the k-Nearest Neighbours (k-NN) algorithm is utilised for both regression and classification applications. Based on the "k" closest data points in the feature space, it generates predictions.

4) Classification Report of Decision Tree Classifier: A supervised machine learning technique used for both regression and classification applications is the Decision Tree Classifier. Using the characteristics in the dataset, it builds a decision-tree model.

5) Classification Report of Gaussian Naive Bayes Classifier: A probabilistic classification technique based on the Bayes theorem is called Gaussian Naive Bayes (GNB). It works especially well for classification jobs if the features are thought to have a Gaussian (normal) distribution and are continuous.

XIV. Hybrid Ensemble Learning

A total of fifteen learning classifiers—three from each of the five techniques mentioned above—will be employed. Next, the ensemble model's final class prediction will be determined by using the Max Voting Classifier approach, with the class predominantly predicted by the weaker learners.

1) Defining and fitting the ensemble voting classifier: To increase overall performance, an ensemble voting classifier aggregates the predictions of several independent classifiers (e.g., Decision Tree, Random Forest, SVM, etc.). It is applicable to jobs involving both regression and classification.

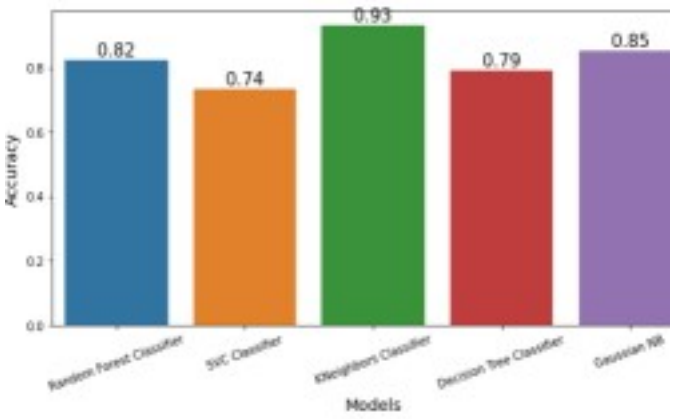

Fig. 2.

2) Plotting the confusion matrix for the final ensemble model : A table known as a confusion matrix is frequently used to illustrate how well a classification model performs[15] when applied to a set of data for which the real values are known. It enables you to see how well a classification algorithm is performing.

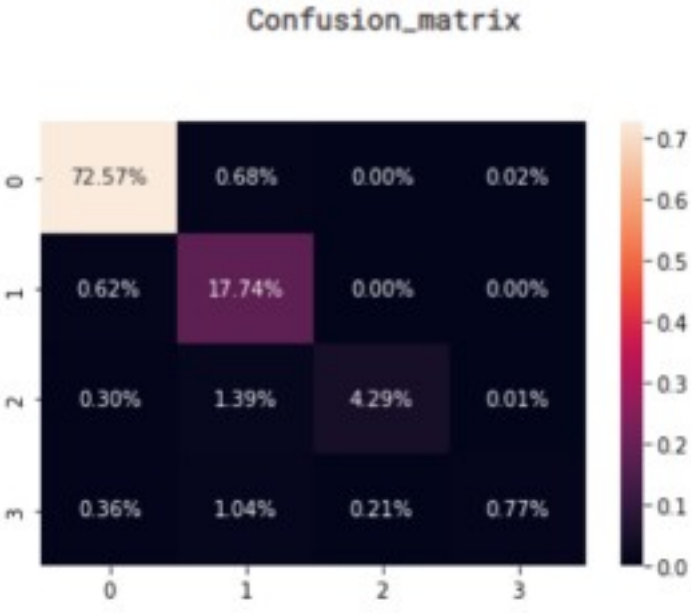

Fig. 3.

The results we obtained for the various models may be summarised as follows, showing that the "Hybrid Ensemble Model" performed better than the separate classic ML models.

CONCLUSION

Our objective for this research was to develop a model that could effectively identify malicious URLs. In order to achieve the aforementioned goal, we used a sizable labelled dataset with over 6 lakh URLs to train and test a number of conventional machine learning techniques, such as Random Forest Classifier, Decision Tree Classifier, Support Vector Classifier, K-Nearest Neighbour, and Gaussian Naive Bayes. With an accuracy of 93.13To enhance the detection system's precision and effectiveness, we integrated the many machine learning classifiers mentioned earlier into a single model through the use of ensemble learning. For the same objective, we employed an ensemble voting classifier, and the result was a remarkable 95.37Suggestions for future development include: • Expanding the scope of testing to encompass a greater variety of malicious URLs; adding more advanced JavaScript function extractors; and using more network characteristics. More significantly, independent of the surfing device being used, employing a trained SVM can offer a real-time service for checking URLs for malware.

REFERENCES

[1] . Khonji, Mahmoud, Youssef Iraqi, and Andrew Jones. "Phishing detection: a literature survey." Communications Surveys and Tutorials, IEEE 15.4 (2013): 2091-2121.

[2] . Anti Phishing Working Group. (2015. March.) APWG Phishing Activity Trend Report 2nd Quarter 2010. [Online]. Available: http://docs.apwg.org/reports/apwg trends report q2 2014.

[3] . Anti Phishing Working Group. (2015. March.) APWG Phishing Activity Trend Report 2nd Quarter 2014. [Online]. Available: http://docs.apwg.org/reports/apwg report q2 2010.

[4] . Huang, Huajun, Junshan Tan, and Lingxi Liu. "Countermeasure techniques for deceptive phishing attack." New Trends in Information and Service Science, 2009. NISS'09. International Conference on. IEEE, 2009.

[5] . Ma, Justin, et al. "Beyond blacklists: learning to detect malicious web sites from suspicious URLs." Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009.

[6] . Dhanalakshmi Ranganayakulu, Chellappan C., DetectingMalicious URLs in E-mail – An Implementation, AASRIProcedia, Vol. 4, 2013, Pages 125-131, ISSN 2212-6716,https://doi.org/10.1016/j.aasri.2013.10.020

[7] . Wikipedia. (2015. March) Uniform Resource Loactor. Avaliable: http://en.wikipedia.org/wiki/Uniform resource locator

[8] . Kausar, Firdous, et al. "Hybrid Client Side Phishing Websites Detection Approach." International Journal of Advanced Computer Science and Applications (IJACSA) 5.7 (2014).

[9] unil, A. Naga Venkata, and Anjali Sardana. "A pagerank based detection technique for phishing web sites." Computers Informatics (ISCI), 2012 IEEE Symposium on. IEEE, 2012.

[10] . Mohammad, Rami M., Fadi Thabtah, and Lee McCluskey. "Intelligent rule-based phishing websites classification." Information Security, IET 8.3 (2014): 153-160.

[11] . Singh, C., and 'Meenu., "Phishing Website Detection Based on Machine Learning: A Survey", IEEE 6th International Conference on Advanced Computing and Communication Systems, Gorakhpur, India, 2020, 978- 1-7281-5197-7.

[12] . Frank Vanhoenshoven, Gonzalo Napoles, Rafael Falcon, Koen Vanhoofand Mario Koppen, Detecting Malicious URLs using Machine LearningTechniques, 978-1-5090-4240-1/16 2016

[13] . A, A. A., and K, P. "Towards the Detection of Phishing Attacks", IEEE 4th International Conference on Trends in Electronics and Informatics, Coimbatore,India, July 27-2020, 978-1-7281-5518-0

[14] . Arun Kulkarni and Leonard L. Brown, " Phishing Websites Detection using Machine Learning ", International Journal of Advanced Computer Science and Applications , Tyler, TX, 2019.

[15] . El Aassal, A., Baki, S., Das, A., and Verma, R. M.,"An In-Depth Benchmarking and Evaluationof Phishing Detection Research for Security Needs", IEEE Access,Houston, U.S., 5-Feb-2020, 2969780.

[16] . Korkmaz, M., Sahingoz, O. K., and Diri, B., "Detection of Phishing Websites by Using Machine Learning- Based URL Analysis",IEEE IIT - Kharagpur, Istanbul, Turkey, 1-Jul-2020, 49239.

[17] Hung Le, Quang Pham, Doyen Sahoo, and Steven C.H. Hoi, "URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection", Conference'17, Washington, DC, USA, arXiv:1802.03162, July 2017.

[18] Tan CL, Chiew KL, Wong K, "PhishWHO: phishing webpage detection via identity keywords extraction and target domain name finder", Decision Support Systems, vol. 88, pp 18–27, 2016.