UNMASKING CYBER THREATS - LEVERAGING MACHINE LEARNING TO DETECT PHISHING WEBSITES

Ms. Sherine. S[1], Dr.S.Stewart Kirubakaran[2], Dr.I.Kala[3]
[1]Third Year B.Tech Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, India.
[2] Assistant Professor, Computer Science and Engineering,, Karunya Institute of Technology and Sciences, Coimbatore, India.
[3]Associate Professor, Department of Computer Science and Engineering, PSG Institute of Technology and Applied Research, Coimbatore, India.

[1]sherine8112004@gmail.com, [2]stewart@karunya.edu,[3]ootykala@gmail.com

Abstract. Nowadays, smart phones are widely used, which makes them sus- ceptible to phishing. The majority of phishing websites attempt to obtain the victim's data by using the same user interface and universal resource location (URL) as the legitimate websites (user name, password, credit card details, etc.). Protecting users from cyberattacks requires an intelligent strategy. Phish- ing can hurt a company in a number of ways, including loss of financial proper- ty, intellectual loss, reputation damage, and disturbance of business activities. As a result, there is a pressing need for a mobile phishing detection system. This project's primary objective is to predict phishing websites, which are common social engineering techniques that resemble trustworthy URLs and webpages. The important goal of this research is to identify websites that are vulnerable to user privacy. Algorithms for statistical machine learning are used for the detection. The algorithm is made to determine a website's quality based on a few characteristics (such as spam reports, report counts, etc.) and user ac- tivity on the website. In the actual world, it functions by notifying consumers as they browse a specific website.

Keywords:phishing, cyber-attacks, statistical, machine learning, surfing.

1 Introduction

Phishing attacks include the delivery of false messages that appear to be from a trustworthy source. We are using email for this. The goal is to either steal personal data like login credentials of user and credit card information and or affect the user's computer with malicious programs. To defend oneself, everyone should be aware of the frequent hack known as phishing. The first step in phishing is to lure a victim with a fraudulent email or other kind of communication. The message appears to have been sent from a reliable source to the addressee. When someone is tricked, they frequently

feel pressured into providing critical information on a scam website. Malware may occasionally make its way onto the target's PC.

In exchange for money, attackers might be pleased to obtain a victim's credit card number or other personal information. Phishing emails may occasionally be sent in an effort to gather employee login passwords or other information for use in a sophisticated assault on a particular company. Phishing is regularly employed as the opening maneuver in cybercrime attacks like ransomware and very persistent threats (APTs). In order to safeguard oneself against phishing attempts, both consumers and businesses must make an effort. Users should use caution. Little errors in the message frequently show where a communication is being forged. Grammar mistakes or domain name modifications, like in the previous URL example, can be among them. Users should also take a moment to reflect on why they are even receiving such an email.

2 Literature Survey

Zichen Fan (2021)[1], 3rd International Conference on Applied Machine Learning (ICAML). This paper presents a strategy for identifying phishing websites using joint properties in great detail. PhishTank was crawled for the original data. Bayes and SVM techniques are combined while training classifiers. The packet flow detection mechanism makes use of Wireshark. The classifier can detect 1000 webpages per second after training. The comparison analysis makes use of a data set that includes 21615 authentic and phishing websites. 51 features are additionally used to train and evaluate the classifiers.

Shweta Singh, M.P. Singh and Ramprakash Pandey (2020)[2], 5th International Conference on Computing, Communication and Security (ICCCS). In order to stop phishing assaults, a phishing detection system is put into place in this article employing deep learning techniques. Convolutional neural networks (CNNs) are used by the system to analyse URLs in order to identify phishing websites. The new system in this research outperformed the previous model with accuracy of 98.00%. As the CNN automatically extracts features from the URLs through its hidden layers. This system need not require features engineering. Another benefit of the suggested system over the older paradigm is this.

M. Amaadet al.[3] classified phishing websites using a hybrid technique. In this paper, the suggested model was put to the test twice. They each carry out phase 1 classification processes and choose the top three models based on high accuracy and as well as other performance criteria. Phase 2 of the process involved further integrating each individual model with the highest three models to produce a hybrid model that is more accurate than the individual models. They had a 97.5 percent accuracy for the test dataset. The disadvantage of this idea is that it usually takes longer to create a hybrid model.

Toqeer Mahmood, Muhammad Wasif Nisar, Junaid Rashid, Tahira Nazir, and others (2020)[4], First International Conference of Smart Systems and Emerging Technologies (SMARTTECH). This study proposed a machine learning-based technique for efficient phishing detection. Overall, the studies' findings show that the suggested approach, when used in conjunction with the Support vector machine classifier, performs the best in reliably identifying 95.66% of phishing and authentic websites while consuming only 22.5% of the unique capabilities. The proposed method exhibits good results when evaluated against a number of popular phishing datasets from the "University of California Irvine (UCI)" collection.

Hossein et al.[5] created the "Fresh-Phish" open-source framework. They built the query in Python and eliminated features. They generate a sizable labelled dataset, which they then use to compare several machine learning classifiers. Results from this analysis utilising machine learning classifiers are extremely accurate. They examine the length of time required to put the model through training.

Mustafa et al. [6] created the safer approach for phishing website detection. To improve accuracy, they collected website-based URL information and used a subset-based selection method. The approaches for selecting subgroups from CFS subgroups and content groups are contrasted and analysed in this article. For classification, machine learning techniques are also employed.

Ahmad et al. [7] suggested three new elements to increase the efficacy of phishing website detection. This study's author distinguished between legitimate and phishing websites using both well-known traits and brand-new criteria. The author has come to the conclusion that this work can be enhanced by fusing these state-of-the-art features with decision tree based machine learning classifying algorithms.

Mohammad et al. [8] suggested approach that detects phishing websites without the need for human interaction by automatically extracting key attributes. The author of this research has come to the conclusion that using their programme to extract features is significantly faster and more accurate than doing so manually.

3 Workflow of Proposed System

The initial step in the research procedure was selecting the right data set. The dataset for this exercise was gathered using Phish Tank. There are many reasons why this dataset was chosen. It includes:

- Working with the enormous data collection is interesting.
- There are many different types of features in the data collection.

3.1 Splitting

A testing portion and a training portion of the dataset must be separated. The dataset was split into testing and training datasets using the "test and train split" approach, with 75% used for training and 25% for testing. The splitting was done once the dependent and independent variables were established.

3.2 Preprocessing

To create a clean dataset, preprocessing requires either adding missing data or remov- ing it entirely. But since the chosen dataset had already undergone preprocessing, I didn't have to do any more. Only the feature scaling preprocessing step was necessary.

3.3 Feature Scaling

A technique for putting the independent variable in the data into a predetermined range is feature scaling. During the data pre-processing, it is done to deal with varied magnitudes. The two different kinds of feature scaling are normalisation and standardisation. The project employs strategies for feature standardisation scaling. The variables should be scaled similarly to avoid one variable taking the lead. Implementation The usage of machine learning algorithms for phishing website identification may have an impact on the result.

3.4 Feature Extraction

Data about an IP address, the length of a URL, a domain name's subdomains, the presence of a favicon, etc. are collected using Python modules like whois, requests, socket, re, ipaddress, and BeautifulSoup. The outcome is kept as a list value. This is being done because the dataset is in this format and the classifier will be trained using input in this format. Hence, when a URL is entered, the system converts it into a Py- thon list of 30 components, where each component stands for a separate feature.

4 Models used in Proposed System

4.1 Logistic Regression

Logistic regression is a powerful and popular method for classification of super-vised algorithms (LR). It can only estimate a binary value, which typically specifies whether an occurrence will take place or not, and is seen as an expansion of conven-tional regression. The probability that a concept has developed belongs to a particu-lar class can be calculated using LR. The result, which is a probability, falls between 0 and 1. As a result, a barrier must be defined to differentiate between two classes in order to use the LR as a base classification. For instance, an input instance is catego-rised as belonging to "class A" if its probability value is greater than 0.50; otherwise, it is

categorised as belonging to "class B". The LR model can be used to more broad-ly model a category variable with more than two values. This LR has been general-ised, and its name is multinomial logistic regression.

4.2 K-Nearest Neighbors

K-nearest neighbors is a prime example of a supervised learning approach which is applied to both classification and regression (KNN). KNN calculates the distance between all of the training points and testing data in an effort to output the correct group for the data for testing. Next, decide which K point number most closely re-sembles the data for testing. The K-Nearest neighbors algorithm calculates the prob-ability that each of the "K" training data classes corresponds to the test data, and it selects the group with top probability. Regression uses the averages of the "K" cho-sen training points to determine its value.

4.3 Support Vector Machine

One of the most popular supervised learning techniques, Support Vector Machine. Support Vector Machine is used to handle classification and regression problems. All the same, Machine Learning Classification challenges are where it is most fre-quently employed. The SVM method aims to identify the best line or distance meas-ure that can separate n-dimensional spaces into classes in order to quickly classify new sets of data in the future. This optimal decision boundary is known as a hyper-plane. SVM is used to choose the maximal vectors and vertices that belong to the hyperplane. Sup-port vectors, that are employed to represent these extreme situations, are the corner-stone of the SVM technique.

4.4 CatBoost Classifier

Yandex's CatBoost is an unique machine learning algorithm. Machine learning frameworks with simple interfaces include Apple's Core ML and Google's Tensor-Flow. It may interact with many data formats to help companies with a variety of current issues. It also provides the most advanced degree of accuracy in its area. In two ways, it is very effective. It produces state-of-the-art results without necessitat-ing the substantial data retraining that some other machine learning algorithms nor-mally demand, and it offers strong out-of-the-box coverage with the more descrip-tive file types connected to several business concerns. The name "CatBoost" was created by fusing the phrases "Category" and "Boosting".

4.5 Gradient Boosting Classifier

Gradient boosting classifiers are a sort of machine learning methods that merge nu-merous ineffective learning models into one strong prediction model. Classifier aug-mentation typically uses decision trees. Recently, a lot of Kaggle machine learn-ing

events have been won. These victories reflect the growing prominence of gradi-ent boosting algorithms, which are particularly good at identifying difficult datasets. Sev-eral gradient boosting classifier methods are supported by the Python machine learn-ing framework Scikit-Learn, most notably XGBoost.

5 Proposed Work

Installing the Python package manager pip is the first step. By typing pip install your package, you can easily download any package of python that is listed in the Python based Package Index. The Flask web framework was used to create the web applica-tion. It has been downloaded and set up to utilize the Jupyter Notebook forIntegrated Development Environment (IDE) of PyCharm. Activate the ML libraries.

- numpy: for any matrix work, notably for mathematical computations
- pandas: handling of data, analysis and manipulation of data.
- matplotlib: visualization of data
- scikit learn: for machine learning algorithms

Python is used to implement the project that is being presented. This particu-lar application was picked since it offers more flexibility and is very helpful from a programmer's point of view. What distinguishes this project from others is the pack-age's simplicity of use for developing Graphical User Interfaces (GUIs), preparing data, and developing and improving machine learning algorithms. In this project, user data is stored in files. The Flask web framework offers the technologies, tools, and libraries needed to create a web application.

Features:

Fig. 5.1 Feature Tabulation

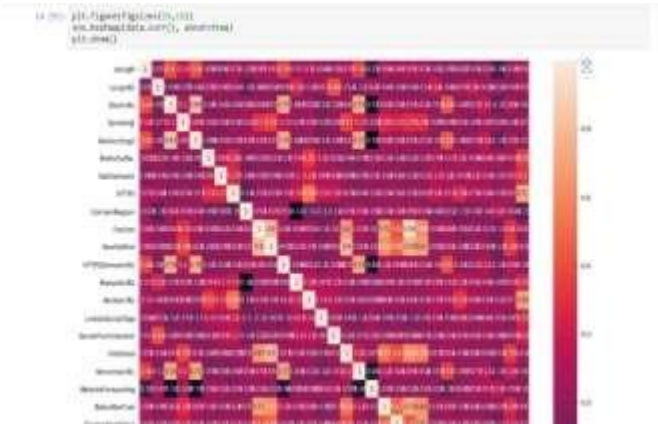

Fig. 5.2 Heatmap defining the correlation between features

Fig. 5.3 Pie Chart representing the phishing count and non-phishing count

Logistic Regression:

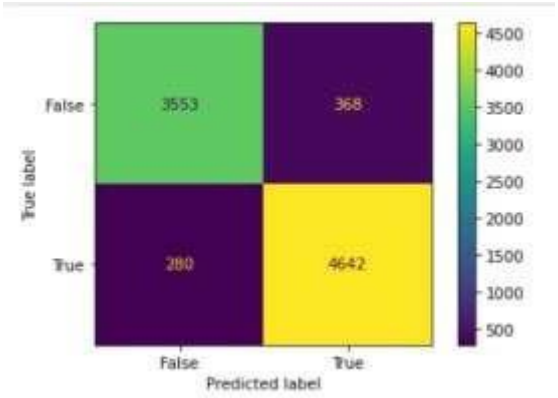

Fig. 5.4 Confusion matrix ofLogistic Regression Classifier

K-Nearest Neighbors:

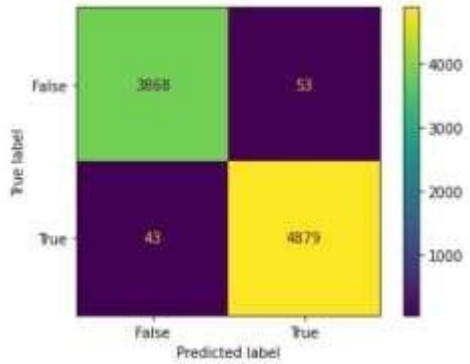

Fig. 5.5 Confusion matrix ofK- Nearest Neighbors Classifier

Support Vector Machine:

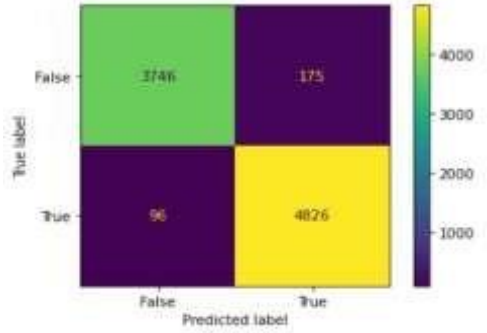

Fig. 5.6 Confusion matrix ofSupport Vector Machine Classifier

CatBoost Classifier:

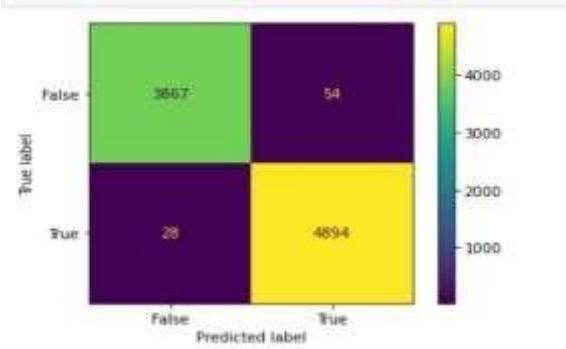

Fig. 5.7 Confusion matrix ofCatBoost Classifier

Gradient Boosting Classifier:

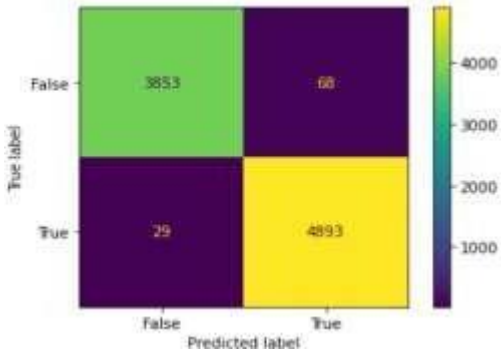

Fig. 5.8 Confusion matrix ofGradient Boosting Classifier

Accuracy:

Fig. 5.10 Comparison of accuracy of different machine learning algorithms

6 Result Analysis

Testing the project

By creating test scenarios for unit testing, the code logic is verified as accurate and that code inputs result in valid outputs. It is critical to check the compiled code flow and each decision branch. It entails testing each component of the program's software individually. It is completed after each unit is complete, but before integration. This invasive structural test requires an understanding of its construction. Unit tests, which also take a close look at a specific arrangement of a system, software, or business process, carry out fundamental testing at the component level.

Integration tests are made to look at connected software parts and determine whether they work together as a single application. The fundamental result of fields or displays is the major focus of event-driven testing. Integration tests show that the overall product is reliable and precise even when the test automation of the individual elements was successful. Integrity testing is performed to identify issues that could occur during component fusion. Software and hardware components are tested as a combined unit to see if they adhere to the set requirements. Black box testing, which includes system testing, doesn't require understanding the concept or internal dynamics of the code.

Result Analysis

To make sure the models can forecast disease more precisely and rapidly, their functioning has been tested for a variety of inputs.

Feature Analysis:

Training and Testing data ratio	Models	Precision	Recall	Accuracy	F1-score
60-40	LG	0.923	0.95	0.927	0.937
	KNN	0.957	0.962	0.954	0.959

	SVM	0.95	0.974	0.957	0.962
	GBC	0.965	0.979	0.969	0.972
	CBC	0.965	0.98	0.969	0.973
70-30	LG	0.924	0.949	0.927	0.936
	KNN	0.957	0.962	0.954	0.959
	SVM	0.954	0.973	0.958	0.963
	GBC	0.963	0.981	0.968	0.972
	CBC	0.968	0.979	0.97	0.973
80-20	LG	0.93	0.953	0.934	0.941
	KNN	0.96	0.962	0.956	0.961
	SVM	0.957	0.98	0.964	0.968
	GBC	0.986	0.989	0.974	0.977
	CBC	0.969	0.982	0.972	0.975

Table 6.1 Comparison of various performance metrics under different model

LG-Logistic Regression
KNN-K Nearest Neighbors
SVM – Support Vector Machine
GB-Gradient Boosting Classifier
CB-CatBoost Classifier

- **Long URL versus Short URL:**

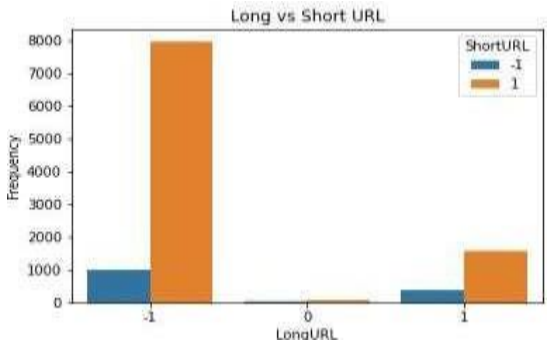

Fig. 6.1 Long URL versus Short URL

- **Frequency count plot for phishing and no-phishing:**

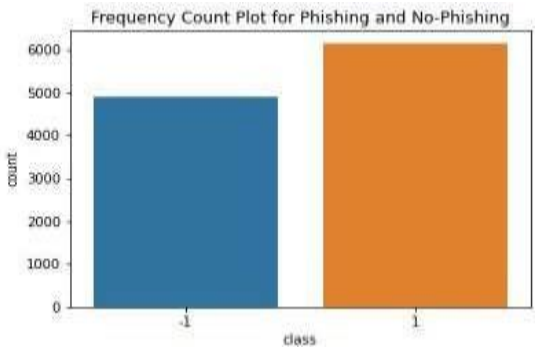

Fig. 6.2 Frequency count plot for phishing and no-phishing

- **Website Traffic:**

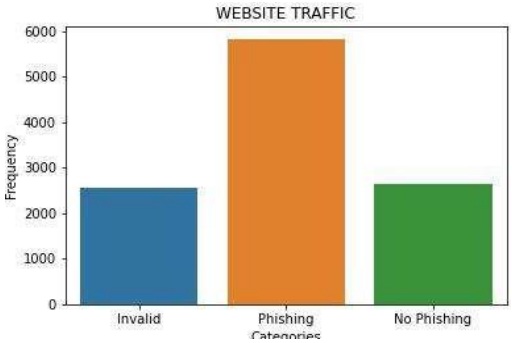

Fig. 6.3 Website Traffic

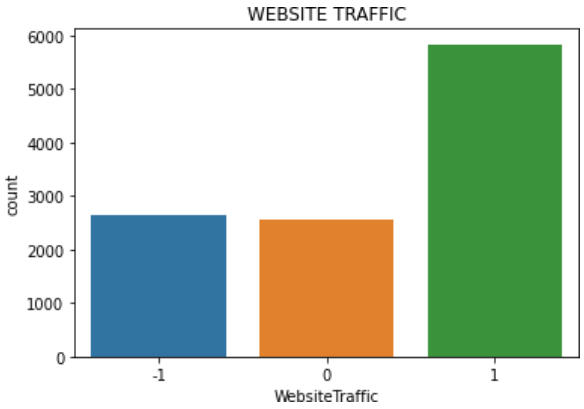

Fig. 6.4 Website traffic vs count

- **Precision of various models under different split ratio:**

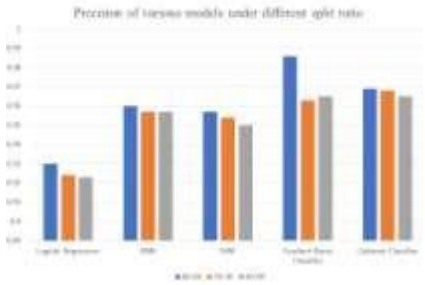

Fig. 6.5 Precision of various models under different split ratio

- **Recall of various models under different split ratio:**

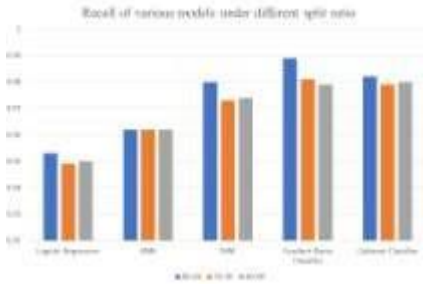

Fig. 6.6 Recall of various models under different split ratio

- **F1 score of various models under different split ratio:**

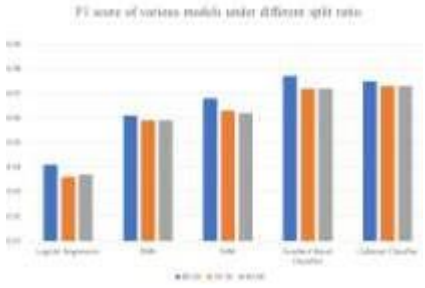

Fig. 6.7 F1 score of various models under different split ratio

- **Accuracy of various models under different split ratio:**

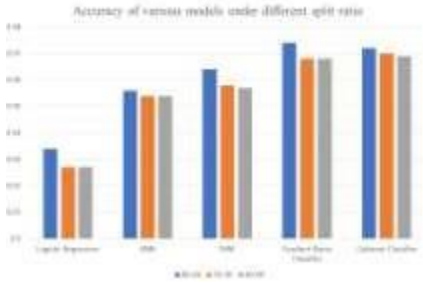

Fig. 6.8 Accuracy of various models under different split ratio

7 Conclusion and Future Work

In this quickly developing electronic world, phishing is becoming a more sophisticated menace. To keep up with the growing global, every country is currently focusing on cashless payment, online shopping, paper tickets, and other technology. If a layman is absolutely unable to understand a security risk, they should never put themself in danger by engaging in financial transactions online. Phishers are focusing their efforts on the installation sector and cloud advantages. The goal of the study is to explore this issue by demonstrating how machine learning may be used to identify phishing websites. It sought to develop a machine having to learn phishing detection system that was accurate, efficient, and effective. The project was developed in a Jupyter Notebook using Python. The proposed method does this by employing Gradient Boosting machine learning classifiers. Moreover, an outstanding accuracy rating was reached. This model can be applied in real time to distinguish between legitimate and phishing URLs.

References

[1] Z. Fan, "Detecting and Classifying Phishing Websites by Machine Learning," 2021 3rd International Conference on Applied Machine Learning (ICAML), 2021, pp. 48-51, doi: 10.1109/ICAML54311.2021.00018.

[2] S. Singh, M. P. Singh and R. Pandey, "Phishing Detection from URLs Using Deep Learning Approach," 2020 5th International Conference on Computing, Communication and Security (ICCCS), 2020, pp. 1-4, doi: 10.1109/ICCCS49678.2020.9277459.)

[3] M. AmaadUl Haq Tahir, Sohail Asghar, Ayesha Zafar, Saira Gillani : A Hybrid Model to Detect Phishing-Sites using Supervised Learning Algorithms :In International Conference on Computational Science and Computational Intelligence IEEE ,2016

[4] J. Rashid, T. Mahmood, M. W. Nisar and T. Nazir, "Phishing Detection Using Machine Learning Technique," 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH), 2020, pp. 43-46, doi: 10.1109/SMART-TECH49988.2020.00026.

[5] Hossein Shirazi, Kyle Haefner, Indrakshi Ray: Fresh-Phish: A Framework for Auto-Detection of Phishing Websites: In (International Conference on Information Reuse and Integration (IRI))IEEE,2017

[6] Mustafa AYDIN, NazifeBAYKAL : Feature Extraction and Classification Phishing Websites Based on URL : IEEE,2015

[7] Ahmad Abunadi, AnazidaZainal ,OluwatobiAkanb: Feature Extraction Process: A Phishing Detection Approach :In IEEE,2013

[8] Mahajan, Rishikesh &Siddavatam, Irfan. (2018). Phishing Website Detection using Machine Learning Algorithms. International Journal of Computer Applications. 181. 45-47. 10.5120/ijca2018918026.

[9] Atharva Deshpande , Omkar Pedamkar , Nachiket Chaudhary , Dr. Swapna Borde, 2021, Detection of Phishing Websites using Machine Learning, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10, Issue 05 (May 2021)

[10] Rami M. Mohammad, FadiThabtah, Lee McCluskey: An Assessment of Features Related to Phishing Websites using an Automated Technique:In The 7th International Conference for Internet Technology and Secured Transactions,IEEE,2012

[11] P. Yang, G. Zhao, and P. Zeng. Phishing website detection based on multidimensional features driven by deep learning. IEEE Access, 7:15196–15209, 2019.

[12] T. Nathezhtha, D. Sangeetha, and V. Vaidehi. Wc-pad: Web crawling based phishing attack detection. In 2019 International Carnahan Conference on Security Technology (ICCST), pages 1–6, 2019.

[13] Y. Huang, Q. Yang, J. Qin, and W. Wen. Phishing url detection via cnn and attention-based hierarchical rnn. In 2019 18th IEEE International Conference On 55 Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pages 112–119, 2019.

[14] Chirag N. Modi and KamatchiAcha. Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review. The Journal of Supercomputing, 73(3):1192–1234, Mar 2017.

[15] Theodoros Iliou, Christos-Nikolaos Anagnostopoulos, Marina Nerantzaki, and George Anastassopoulos. A novel machine learning data preprocessing method for enhancing classification algorithms performance. 09 2015.

[16] S. Patil and S. Dhage. A methodical overview on phishing detection along with an organized way to construct an anti-phishing framework. In 2019 5th International Conference on Advanced Computing Communication Systems (ICACCS), pages 588– 593, 2019.

[17] K. S. C. Yong, K. L. Chiew, and C. L. Tan. A survey of the qr code phishing: the current attacks and countermeasures. In 2019 7th International Conference on Smart Computing Communications (ICSCC), pages 1–5, 2019.

[18] G. J. W. Kathrine, P. M. Praise, A. A. Rose, and E. C. Kalaivani. Variants of phishing attacks and their detection techniques. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), pages 255–259, 2019.

[19] Sebastian Raschka. About feature scaling and normalization and the effect of standardization for machine learning algorithms. 07 2014.

[20] ”modules - python documentation”. https://docs.python.org/3/tutorial/ modules.html, 2020.