International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 3 | Mar-2025 www.irjet.net p-ISSN: 2395-0072

Mischievous Urls detection based on multi-feature using Soft Voting Classifier

Rini Gupta [1], Prof. Nagendra Kumar[2], Prof. Shobha Rajak[3]

[1]*Reseacrh Scholar, Department of CSE, Shri Ram Institute of Science and Technology, Jabalpur, M.P.*
[2]*Professor, Department of CSE, Shri Ram Institute of Science and Technology, Jabalpur, M.P.*
[3]*Professor, Department of CSE, Shri Ram Institute of Science and Technology, Jabalpur, M.P.*

Abstract – URLs allow Internet users to move from one website to another. Fully represent access to content stored on servers somewhere in the world. URLs are available by simply clicking on a link or image or typing in our browsers. A favourite method used by attackers and children of text is to deceive the social media because regular users still click on any link or visit any URL they find. Blocking other URLs is a fundamental and essential way to provide a basic level of security. With the advent of internet technology, network security is under various threats. In particular, cybercriminals can spread the same dangerous industries (URLs) to attack as criminals by sensitive and spam information. Searching for the wrong URL is vital in preventing this attack. Cybercriminals use malicious URLs as distribution channels to distribute malicious software on the web. Attackers use browser vulnerabilities to install malicious software so that they can access the victim's computer remotely. A malware program aims to gain access to the network, filter sensitive information, and secretly monitor targeted computer systems. In this project, we compare models and find the best guessing model for classifying spam and ham URLs in a better way. Our proposed prediction model seeks to improve the accuracy of the forecast by using various factors that take into account the interaction effect of different parameters.

Keywords: Uniform resource locators, Phishing, Diversity, Machine learning, Feature Engineering, Soft Voting Classifier, Accuracy.

1. INTRODUCTION

Phishing attacks are cybercrime using social engineering to deceive users into stealing their information, such as personal identity, financial information, etc. Masquerading as legitimate sources, attackers can reach victims by sending fraudulent messages using emails (such as Gmail, Outlook, etc.) or social media platforms (like Twitter, Facebook, etc.). Users become vulnerable if they input their information or download attachment files [1]. In recent years, there has been an increase in social media platform attacks since it is easy for attackers to reach many users from anywhere in the world by posting a single message [2]. According to [2], the Anti-Phishing Working Group (APWG) reports the number of phishing attacks increased by 250000 in one month in Jan 2021. In addition, the number of business compromises increased 56% from the last quarter of 2020 to the first quarter of 2021. Fig. 1.1 shows that the most targeted industries in 2021 are financial institutions, social media, and web emails [2].

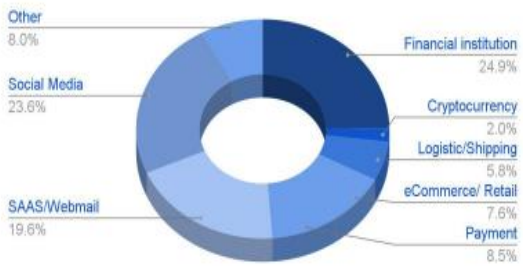

Figure 1.1: APWG report 2023 [2].

According to figure 1.1, the main goal of attackers is to steal victims' financial or personal information by targeting financial markets and social media platforms, respectively. Attackers can also send malware that can lead to other network attacks, such as malware attacks, ransomware attacks, etc. Most organizations now rely on human knowledge to detect these attacks [4]. However, due to the similarity between legitimate and fake messages, phishing attacks are difficult to detect even for experts. Therefore, cybersecurity experts are paying more attention to email links such as uniform resource locators (urls) or email ids to identify phishing emails. However, attackers are improving their attack techniques by using new techniques to create phishing attacks that are difficult to detect. For example, they create phishing urls such as https://www.facebook.com/, https://www.faceb00k.com/, https://www.facebook and web pages that look like harmless urls. Therefore, it is important to determine ways to distinguish between phishing urls and harmless urls. Therefore, researchers have proposed various solutions against phishing in recent years, such as blacklists [5], traditional machine learning [6], and deep learning (dl) [7], [8], [9].

Below we provide a brief review of each solution.

• A blacklist is a list of website urls that are most likely to be phishing sites. Any url or ip on this list will be blocked. However, there are drawbacks to this approach. The system needs to have a phishing attack url to block it; if the url is not on the list, it will not be detected.

• Traditional machine learning models are used to detect phishing attacks. However, traditional machine learning models require manual extraction [11]. Therefore, extracting a set of features is labor-intensive and time-consuming [12]. These functions are based on existing urls. Therefore, when attackers create new phishing urls, the analysis and extraction will increase, which will cause the field length to be larger [10]. Despite efforts to examine various features and dimensions, it is impossible to avoid attacks from new phishing urls [10].

• The advantage of using deep learning to detect phishing urls is that the model can extract features from text and images without human intervention. However, due to the development of phishing attacks and new deep learning techniques, there are some problems with phishing sites. For example, train a model to identify long urls. However, it does not detect small urls [13]. In addition, deep learning has some disadvantages such as requiring a large amount of data to train, test, and validate the model [14], [15]. The cost of the deep learning model is also high due to its complexity [12]. Phishing attacks can be detected using different types of information, such as url-based [9], content-based [16], [17], and hybrid-based [18]. Url-based methods extract url data without searching other information, such as web pages, directories, etc. However, extracting only url-based features results in missing important features of phishing web pages, such as page names and page codes. It is also difficult to identify small urls using only url-based. Content-based systems extract information from the web, such as images, JavaScript, text, and hypertext markup language (html) code. Upload to capture. Content integration combines url-based and content-based features.

1.1 Phishing Attacks

As discussed in the previous section, phishing attacks are on the rise. Part of the increase is due to the ease of creating these attacks. They can be done anywhere and do not have to be in the same location as the victim. These consequences make phishing attacks one of the most dangerous attacks for individuals and organizations. It works by sending fake urls to victims using email and social media. These urls send victims to fake websites that trick them into sharing personal information, such as credit card numbers and login credentials.

There are four parts to a URL. First, the URL protocol tells us how the data is transferred. Second, the host, which contains the top-level domain (tld) and secondary domain (sld) [19]. Tlds help distinguish the purpose of the domain name; for example, edu represents the domain used for education. The sld usually has a website name. The third is the path containing the address of the requested page. The last one can be a question and there will be many no's. It can also be an anchor that takes the user to a specific section of the web page. Attackers can use different techniques to create fake urls. First, attackers can create phishing urls through new query (sql) injection or cross-site compilation (xss) attacks [20]. Secondly, an attacker can use an organization name with a different tld (e.g. Sld) such as www.ua.edu instead of www.ua.com [21]. Thus, uninformed users can fall into this trap and become victims. Therefore, organizations focus on developing a system that detects phishing attacks by analyzing domain names, sld and tld together. Thirdly, urls use different conventions such as "shttp". Fifthly, an attacker can create a URL using random words, which can make the URL very long. Sixth, the attacker creates the URL and wraps it behind a smaller URL. Tiny URL is a service that shortens urls by creating new urls with different patterns [22], for example, https://tinyurl.com/brbm97cx URL. Normal detection models fail to detect tiny urls because they have a different structure than the original URL. Finally, the attacker uses popular blog hosting platforms such as google sites [23] to create a website and places the phishing URL as a link to the fake blog. Thus, the attacker hides the phishing URL on a legitimate website. Therefore, phishing detection technology will not detect these attacks because it can identify urls generated by google sites (legitimate urls, not phishing urls) and protect your users from threats. When we use the internet to facilitate our work, at the same time, many attackers try to steal information from our system. There are many ways to combat bad urls. Blocklists are included in antivirus programs, blockchain/tracking systems, and spam filters. The blocklist method is simple and gives the best accuracy if the list is updated in a timely manner, but this method will not detect the newly created URL problem.

Nowadays, machine learning is used in many ways, and network security is one of them. In today's malicious URL detection, machine learning plays an important role in identifying malicious urls. The URL represents the actual application site, which indicates www. The same service received has two parts:

a) Username, i.e. the domain name or ip address where the application is located.

B) Procedure specifies which procedure to use.

Machine learning uses a portion of the URL data for statistical learning to learn a predictive function to classify urls as malicious or dangerous. This results in the creation of new urls instead of blocking the path. A special point for training learning models is the availability of teaching materials. In the case of malicious URL return, this will be with a large set of urls. Machine learning can be divided into supervised, unsupervised learning, and semi-supervised learning, where training data is collected, unlabeled, and a limited portion of the training data is shown. Labels are associated with information about whether the URL is malicious or harmless. After collecting the data required for training, we need to extract the instructions that will fully describe the url and allow mathematical transformation by the machine learning model. Here, we first extract the length,

number, and binary features of the existing urls in the repository, then add them as rows to the repository and analyze these features. To understand and read information, first do the information and see the information. Machine learning such as Adaboost and random forest are used to separate dangerous urls from dangerous urls. Group voting is used to vote for the classification that provides the most accuracy.

Machine learning algorithms were used to achieve the purpose of this article. Machine learning algorithms are not enough to process big data and bring bad urls in a better and more accurate way. The role of machine learning is to use embedded historical data to predict the resulting results. This article uses Adaboost and random forest algorithms. The voting group is used to vote for the classification that provides the best accuracy for the best classification of spam and normal urls.

II Related Work

Divya Kapil et al., [24] experiments 4 algorithms using the weka tool and also compares the different detection techniques. ISC URL 2016 dataset was used for the experiment, which shows the results using performance metrics TPR, FPR, Precision, Recall and F-measure. Random numbers of samples were taken. Sample dataset contains 47 attributes. Some features were rejected so that overfitting problems can be avoided. The dataset is in 'Malware', 'Spam','Benign', 'Defacement' and 'Phishing' form. Malware URLs are a critical issue for the researchers and machine learning techniques are very helpful in various areas. Malicious URLs detection using machine learning is a better idea than conventional techniques. Dataset is divided into 80% for training and 20% for testing. J48, Random forest, Bayes-Net and lazy classifiers were used and multi-class classification was performed, where classes were labeled as Defacement, Phishing, Spam, Malware and benign. It is observed that Random Forest achieves the highest TPR about 96% followed by Lazy classifier with 95% TPR.

Jino S Ganesh et al., [25] identified Phishing URLs under weak supervision, which requires a small amount of labeled data to start the learning process. They implemented a new hybrid model which combined NLP (natural language processing-based features) and word vector. Two parallel modules are used that extract functional representations of URLs. The first is the character level CNN module. The other is an attention-based hierarchical RNN module that is proposed to find phishing URL detection. The Detecting process is based on:

- Finding the data, retrieving and summarizing the data.
- Making the prediction based on the analysis data.
- Calculating the probabilities of the specific results.
- Adapting to certain development autonomously.
- Optimizing the process based on the recognized pattern.

Four different machine learning algorithms are used, like logistic regression, decision tree, random forest, and multilayer perceptron neural networks. Random forest achieves the highest accuracy about 98.6%.

R. Naresh et al., [26] identified malicious uniform resource locators employing a combination of URL lexical options, payload size, and python supply options. Feature extraction techniques such as Host-based features, lexical based features and popularity based features were used to detect the malicious URLs. Feature extraction technique is used to detect the malicious URLs.

Detecting process: Web Crawling, Feature extraction and processing, Training of classifiers and Running classification to detect malicious URLs. URLs were collected from the Alexa ranking website. Numbers of URL (400,000) out of which 80,000 were malicious and others clean, this makes our data set. The research targets mainly on domain-name and URL's attributes.

Classifiers used are SVM and Logistic Regression. They used a Support Vector Machine with a polynomial kernel and logistic regression to attain maximum accuracy. Logistic regression achieves accuracy about 98%.

Rajesh Kumar et al., [27] used Black and white list technology and machine learning algorithms and formed a multilayer filtering model for detection of malicious URLs. The model was trained for each machine learning algorithm i.e., naive Bayesian classification and decision tree classifier threshold and this threshold is used to refer to guide two classifiers for filtering URL. Naive Bayesian classifier, Decision Tree classifier and SVM classifiers were combined in one multilayer model to improve the malicious URL detection system in terms of accuracy. The Multilayer filter model performs better than all the three classifier models achieving the accuracy about 79.55%.

Gopinath Palaniappan et al., [28] explored an active DNS analysis approach for classifying a domain name as benign or malicious by including the web-based features of the domain name in addition to the usually used lexical-based and DNS-based features. They extracted features of a domain name under DNS-based, web-based, blacklisting and lexical-based categories, and trained a logistic regression classifier and tested the classifier to classify unlabeled dataset of domain names and got an accuracy of about 60% using a small dataset of about 10000 domain names. The usage of web-based features of domain names in addition to using blacklists, DNS data, and lexical features to identify malicious domains has been shown.

III. PROPOSED WORK

The proposed model flow is shown below in figure 3.1.

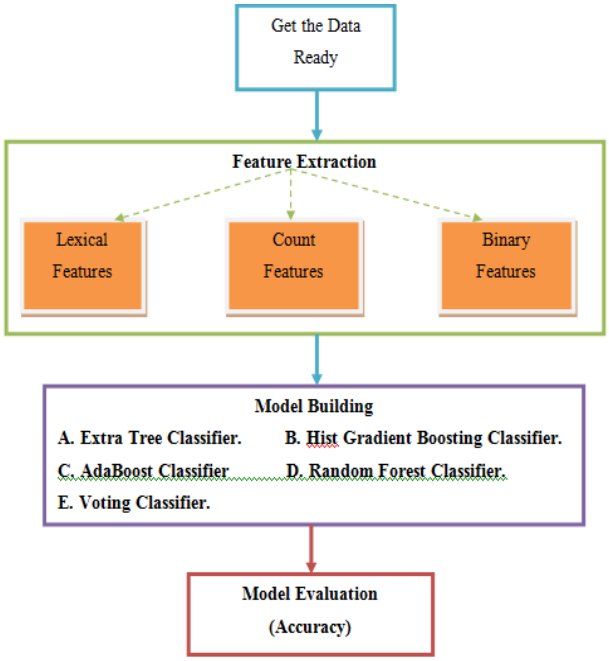

Figure 3.1: Model Flow-work.

3.1 STEPS IN BUILDING THE MODEL ARE:

(i) Data Encoding: This is the initial step in building the model, the feature extraction carried in such a way that the outputs are binary values (0 and 1).

(ii) Feature Extraction: Length, count and binary features have been extracted and added to our dataset using lambda function.

(iii) URL parser: This is a freely available tool/parser which is used in this project to split the given URL into separate parts like hostname length, path length, TLD length, etc,.

(iv) Tokenizing URL: "In Python tokenization basically means separating text into small lines, words or even creating non-English language words. Various token functions are built into the nltk module itself and can be used in programs. Here URLs are subdivided into smaller fields for data analysis".

(v) Data Scaling: No need for data scaling as there were no such columns or data was present.

(vi) Training and Test Set: The fifth step is to split the data into train data and test data using the 'train_test_split' function available at the 'sklearn' library. The classification rate is selected at 80:20, which means that 80% of the data is selected for training and the remaining 20% is selected for the evaluation of new comments and class decisions. A high percentage of training data makes the model train better.

(vii) Balancing Data: The dataset is highly imbalanced with 76.80% of benign URLs and 23.20% of malicious URLs. The dataset has to be balanced before training the model. Hence as the next step, balancing of training data is done using the oversampling technique. If the data is imbalanced, the model will be biased towards the majority variable.

(viii) Feature Selection: The fifth step is to select the key features of the model. This step plays a crucial role as it is very important to find the most relevant features related to the machine learning model.

(ix) Model Building: The last step is to build a binary classification model to detect and classify the benign and malicious URLs. AdaBoost classifier, Random Forest classifier are used, compared and voted using the voting classifier which clearly proves that Random Forest Classifier gives better performance metrics.

(x) Cross-Validation: A 5-fold cross-validation strategy is implemented to validate the Machine Learning model to check the validity of the results.

(xi) Detection: Function has been created and different URLs were provided as user input to detect and classify whether the given URL belongs to a benign class of malicious class. If the given URL is identified as malicious, then an alert message box is displayed, and a "Safe URL" message is printed.

The classifiers like AdaBoost and Random Forest algorithms were used in model building which were later given into the voting classifier to check the best model for our dataset.

A) Adaptive Boosting Algorithm (AdaBoost):

Adaptive Boosting Algorithm, popularly known as AdaBoost Algorithm is on of the Machine

Learning methods used as an Ensemble Method. The most common algorithm used with

AdaBoost is single-level decision trees which means for Decision trees with only 1 division.

These trees are also called Decision Stumps. This algorithm creates a model and provides equal weights for all data points. It then gives us high weights on points that are poorly organized.

Now all the points with the highest weight are given extra value in the next model. It will retain the training models until further notice without a minor error.

B. Voting Classifier:

A voting classifier is a Machine Learning ensemble method which is trained on multiple models and gives an output (class) which is based on the probability of that class being

predicted is high. The prediction from each classifier are sent to voting classifier which combines the predictions and gives final predictions based on majority of votes. In this way, we can eliminate use of multiple models to predict multiple outcomes by passing the predictions from multiple classes to the voting classifier which predicts based on majority of votes. There are two types of votes which are supported by voting classifier.

1. Hard Voting: In hard voting, the final predicted outcome is the outcome which has majority of votes, i.e., the outcome which has highest probability of being predicted by individual classifiers. Suppose there are three classifiers which predicted the outcome class as (2, 1, 2) respectively. So, here the majority of classifiers predicted 2 as outcome. Hence 2 will be the final prediction outcome.

2. **Soft Voting:** In soft voting, the final prediction outcome is based on the average of the probabilities which are gives to that class by each of the classifiers. Suppose, for the inputs provided to the classifiers, the prediction probability for class 1 = (0.20, 0.37) and class 2 = (0.11, 0.53). So the average for class 1 is 0.285 and class 2 is 0.32.So, it is evident that class 2 has highest average probability. Hence, the prediction outcome of the voting classifier through soft voting is class 2.

IV. RESULTS WORK

The results of the models are given below in figure 4.1. From the figure it is concluded that all the classifiers have better accuracy. Also the accuracy gets reduced after applying the data balancing techniques.

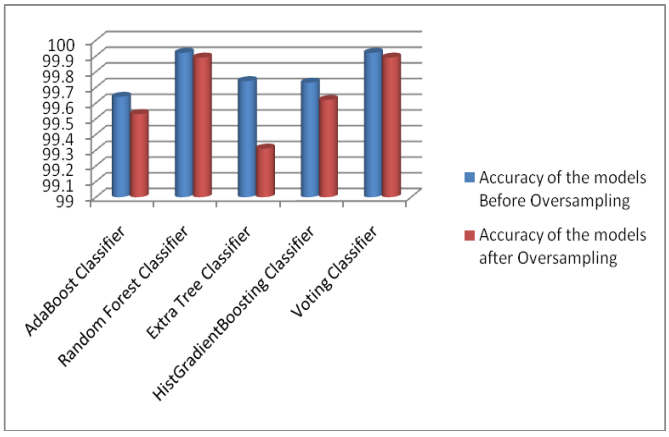

Figure 4.1: Accuracy Comparison Chart.

The training accuracy of the model is 100% and the test accuracy is also 99%. Hence we can ensure that there is no overfitting in the model. Also, we have applied a 5-fold cross-validation strategy to cross check the validity of the model. Both malicious and benign URLs were given as user input to the model to validate the model. The model is observed to give the best accuracy and predicts better when tested.

4.1 CONFUSION MATRIX

A confusion matrix is a multi-dimensional square matrix which is used to test the performance of the given classification model. The size of matrix is the number of target classes. The confusion matrix compares the actual target values to the values that are predicted by the model. The confusion matrix for the classifiers has following information.

- **True positive(s) (TP):** These are the links in which we have predicted benign and are actually benign.
- **True negative(s) (TN):** The links which we have predicted spam/malicious, and they are actually spam/malicious.
- **False Positive(s) (FP):** The links that we have predicted malicious, but they are actually benign. (Also known as "Type I error").
- **False negative(s) (FN):** The links which we have predicted benign, but they are actually malicious. (Also known as "Type II error")

V. Conclusion

Many Malicious URL detection model does the binary classification using machine learning classifiers namely Random Forest and AdaBoost classifiers. The voting classifier is used to check the model that is giving highest accuracy amongst all the models .The results shows that the Random Forest algorithm performs well compared to AdaBoost classifier. The Random Forest classifier gives accuracy about 99.8% whereas AdaBoost classifier gives accuracy about 99.5% and hence voting classifier predicts that the Random Forest algorithm performs well.

The function has been created which identifies the URL as a spam or ham URL. The URLs are accepted as user input and detected whether they are benign or malicious ones. If the URL is found malicious then an alert message box will be displayed showing that "Avoid clicking on such URLs" else "Safe URL" message will be printed. For the URLs that are found malicious, some precautionary measures can be carried out, such as blacklisting of URLs and red-listing the URLs so that they don't appear anymore in the future.

VI. REFERENCE

[1] Y. Zhang, Y. Xiao, K. Ghaboosi, J. Zhang, and H. Deng, ''A survey of cyber crimes,'' Secur. Commun. Netw. vol. 5, no. 4, pp. 422–437, 2012.
[2] APWG Developers. (2021). Phishing Activity Trends Report. [Online]. Available: https://apwg.org/trendsreports.
[3] M. Lei, Y. Xiao, S. V. Vrbsky, and C.-C. Li, "Virtual password using random linear functions for on-line services, ATM machines, and pervasive computing," Comput. Commun. vol. 31, no. 18, pp. 4367–4375, Dec. 2008.

[4] P. Burda, L. Allodi, and N. Zannone, "Don't forget the human: A crowd sourced approach to automate response and containment against spear phishing attacks," in Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS PW), Sep. 2020, pp. 471–476.
[5] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–5.
[6] W. Zhang, Y.-X. Ding, Y. Tang, and B. Zhao, "Malicious web page detection based on on-line learning algorithm," in Proc. Int. Conf. Mach. Learn. Cybern. vol. 4, Jul. 2011, pp. 1914–1919.
[7] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. González, "Classifying phishing URLs using recurrent neural networks," in Proc. APWG Symp. Electron. Crime Res. (eCrime), 2017, pp. 1–8.
[8] 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, pp. 166–178, 2018.
[9] Y. Fang, C. Zhang, C. Huang, L. Liu, and Y. Yang, "Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism," IEEE Access, vol. 7, pp. 56329–56340, 2019.
[10] J. Feng, L. Zou, O. Ye, and J. Han, "Web2Vec: Phishing webpage detection method based on multidimensional features driven by deep learning," IEEE Access, vol. 8, pp. 221214–221224, 2020.
[11] H. Cheng, J. Liu, T. Xu, B. Ren, J. Mao, and W. Zhang, "Machine learning based low-rate DDoS attack detection for SDN enabled IoT networks," Int. J. Sens. Netw., vol. 34, no. 1, pp. 56–69, 2020.
[12] S. Christin, É. Hervet, and N. Lecomte, "Applications for deep learning in ecology," Methods Ecol. Evol., vol. 10, no. 10, pp. 1632–1644, Oct. 2019.
[13] A. Aggarwal, A. Rajadesingan, and P. Kumaraguru, "PhishAri: Automatic realtime phishing detection on Twitter," in Proc. eCrime Res. Summit, Oct. 2012, pp. 1–12.
[14] H. Ma, Y. Zuo, and T. Li, "Vessel navigation behavior analysis and multiple-trajectory prediction model based on AIS data," J. Adv. Transp., vol. 2022, pp. 1–10, Jan. 2022.
[15] J. Fang, B. Li, and M. GAO, "Collaborative filtering recommendation algorithm based on deep neural network fusion," Int. J. Sens. Netw., vol. 34, no. 2, pp. 71–80, 2020.
[16] E. S. Gualberto, R. T. De Sousa, T. P. De Brito Vieira, J. P. C. L. Da Costa, and C. G. Duque, "The answer is in the text: Multi-stage methods for phishing detection based on feature engineering," IEEE Access, vol. 8, pp. 223529–223547, 2020.
[17] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," IEEE Trans. Smart Grid, vol. 10, no. 1, pp. 841–851, Jan. 2019.
[18] E. Zhu, Y. Chen, C. Ye, X. Li, and F. Liu, "OFS-NN: An effective phishing websites detection model based on optimal feature selection and neural network," IEEE Access, vol. 7, pp. 73271–73284, 2019.
[19] T. Mahjabin, Y. Xiao, T. Li, and C. L. P. Chen, "Load distributed and benign-bot mitigation methods for IoT DNS flood attacks," IEEE Internet Things J., vol. 7, no. 2, pp. 986–1000, Feb. 2020.
[20] W. Yang, W. Zuo, and B. Cui, "Detecting malicious URLs via a keyword based convolutional gated-recurrent-unit neural network," IEEE Access, vol. 7, pp. 29891–29900, 2019.
[21] M. Somesha, A. R. Pais, R. S. Rao, and V. S. Rathour, "Efficient deep learning techniques for the detection of phishing websites," Sadhan ¯ a¯, vol. 45, no. 1, pp. 1–18, Dec. 2020.
[22] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.-P. Niyigena, "An effective phishing detection model based on character level convolutional neural network from URL," Electronics, vol. 9, no. 9, p. 1514, Sep. 2020.
[23] Google Developers. (2020). Google Site. [Online]. Available: https:// sites.google.com/.
[24] D.B.A.N.J. Kapil, "Machine Learning-Based Malicious URL Detection," 2, vol. 8, no. 4S, pp. 22-26, 2020.
[25] S. L. R. R. L. A. V. M. G. L. M. R. J. Jany Shabu, "Machine Learning-Based Malicious Website Detection," Journal of Computational and Theoretical Nanoscience, vol. 17, no. 8, pp. 3468-3472, 2020.
[26] R.A.G.S. Naresh, "Malicious URL detection system using combined SVM and logistic regression model," International Journal of Advanced Research in Engineering and Technology, vol. 11, no. 4, pp. 63-73, 2020.
[27] A. L. L. W. P. S. S. Joshi, "Using Lexical Features for Malicious URL Detection – A Machine Learning Approach," 2019.
[28] G.S.S.S.B.S.S.B.B.W. Palaniappan, "Malicious Domain Detection Using Machine Learning on Domain Name Features, Host-Based Features and Web-Based Features," Procedia Computer Science, vol. 171, no. 2019, pp. 654-661, 2020.