Revue d'Intelligence Artificielle
Vol. 37, No. 2, April, 2023, pp. 525-530
Journal homepage: http://iieta.org/journals/ria

ConvNet Based Malicious URL Identification for Safer Use

Vinod Sapkal*, Praveen Gupta

Department of Computer Science and Engineering, CSMU, Navi Mumbai, Panvel 410221, India

Corresponding Author Email: vinod180129@csmu.ac.in

(This article is part of the Special Issue **Technology Innovations and AI Technology in Healthcare**)

https://doi.org/10.18280/ria.370230

Received: 2 January 2023
Accepted: 10 February 2023

Keywords:
malicious URL, machine learning, CNN, spam, Naïve Bayes

ABSTRACT

A malicious URL or website is a type of threat that can affect the users' cybersecurity. It can host unsolicited content and lure users into clicking on links and downloading malware. It can also lead to the theft of private information and monetary losses. People must take the necessary steps to prevent these types of threats from happening promptly. Unfortunately, denylists are not capable of identifying new malicious content. Instead, they are mainly used to identify existing threats. Due to the increasing number of studies being conducted on the use of machine learning techniques, the general capabilities of these tools have been improved. The rise of the internet has made it an essential component of our lives. It allows us to exchange information and knowledge in a timelier manner. Unfortunately, identity fraud and identity theft are two of the most common forms of cybercrime. In both cases, the attackers' goal is to collect the users' personal data so they can commit fraud or deceit for financial gain. Phishing, drive-by exploits, and spam are some types of content commonly featured in malicious URLs. They are also designed to trick users into clicking on links and downloading malware. The vast majority of these scams are carried out through email, and they result in losses of billions of dollars. Systems that are capable of quickly identifying and preventing these types of crimes need to be developed, as well as have the ability to spot new malicious content. Blacklist methods have traditionally been used to detect these types of crimes. On the other hand, blacklists cannot identify newly produced harmful content. Due to the increasing number of studies being conducted on machine learning techniques to improve the detection of harmful web pages, the focus on this field has increased. This article presents an algorithm that can analyze and predict the likelihood of a link being good or bad. It is compared with other standard methods to analyze the performance of this method.

1. INTRODUCTION

The term "cyber security" refers to an organisation as well as a collection of assets, procedures, and systems utilised to protect cyberspace and cyberspace-enabled systems from miscorrelated occurrences due to default possession rights. The term "cyber safety" refers to the accumulation of tools, legislation, security measures, education, risk management strategies guarantee, and technology that can be utilised to protect cyber organisations and the online environment. Many aspects of our day-to-day lives, such as communication, co-ordination, exchange, financial services, registration numbers, packages, and many others, are increasingly moving from the physical world to the virtual world as a result of the dramatic increase in the number of people using the internet. This trend is largely attributable to the rapid growth of the internet. Because of this, evil people and assailants have also moved to this overseas location, where they may more easily remain anonymous while carrying out their threats and crimes. Therefore, era needs to be implemented and prepared properly with the assistance about the use of Cyber safety in order to guarantee the confidentiality of cyber statistics and ensure their safety.

Hackers frequently make use of phishing and spam [1, 2] to deceive users into clicking on malicious URLs. If this is successful, the victims' systems will be infected with Trojans, or sensitive information about the victims will be disclosed. Being able to identify dangerous URLs is very important for users, as it can help them protect their computers from being compromised by malicious websites. Blacklist-based methods have traditionally been used in research on harmful URL detection. These methods identify malicious URLs. This approach offers benefits that cannot be found elsewhere. It is easy to comprehend, possesses a fast rate of speed, and has a low false-positive rate. Today, the domain generation algorithm known as DGA is able to generate thousands of unique domain names every day. This makes it impossible for traditional blacklist-based methods to identify these types of domain names.

Researchers have been employing a method of machine learning to determine whether or not a URL is dangerous. However, these methods frequently need the features to be extracted by hand, and malicious actors can craft these traits so that they cannot be traced back to them. In light of the currently complicated network environment, one of the primary focuses of study is the development of a malicious URL detection model that is more successful.

A model for identifying fraudulent URLs that is based on a convolutional neural network is proposed in this paper. It uses word embedding, which is based on the method of character embedding, to automatically extract features and learn the expression of the URL. In the meantime, we are conducting a number of comparison experiments to test the model's robustness.

The development of a model that amassed a significant number of legitimate and phishing URLs. The research uses the collected information to evaluate and compare a number of different classification algorithms. These classification algorithms include KNN, Logistic Regression, and the Naive Bayes (NB) methodology.

The remaining parts of this work are structured as described below. In Section 2, we examine the research that has already been done on different strategies for detecting malicious URLs. The following section (Section 3) discusses the malicious URL detection methodology and its primary components. In Section 4, we put both the embedding approaches and the malicious URL detection model through their paces by doing tests on each. Section 5 is where we will present the conclusion.

2. RELATED WORK

Conventional detection approaches that are based on blacklists and detection methods that are based upon machine learning are the two primary categories into which the currently available methods [3-5] of identifying malicious URLs may be loosely classified. The detection method that is based on a blacklist is introduced in the literature [6, 7]. Although this method is straightforward and effective, it cannot identify newly produced malicious URLs, which means it has significant limits. According to the research presented in the study [8], attackers can circumvent the conventional detection method that is based on a blacklist by generating a wide variety of fraudulent domain names using a random seed.

Researchers have used machine learning technologies to identify malicious URLs, as documented in the aforementioned literatures [9-11]. The classification of a URL as either malicious or benign is accomplished through machine learning, which involves learning a forecasting model derived from statistical features. In order to extract the features, this technique attempts to analyse URLs along with the information included on applicable websites or web pages. Static characteristics and selected variables are typically the two categories used to classify the features extracted using this method. The contents of web pages, including information about hosts and HTML and JavaScript code, can be used in the literature to identify and retrieve various network traffic-related properties [12]. A support vector machine detects and retrieves these properties [13].

The authors [14] recommended the use of a dynamic whitelist that is capable of self-replicating updates. Their method consists of the following two stages: (i) the matching of IP addresses, and (ii) the extraction of features from individual URL text parts. The experiments' results demonstrated that internet users' protection from misleading URLs was quite effective. In order to manage URLs, blacklists are constructed. In most cases, a blacklist is utilised as an essential component of spam detection systems, anti-virus software, and other security software systems. The most important benefit of employing a blacklist is that it stops hackers from utilising the same URL or IP address repeatedly. On the other hand, a blacklist might not be able to stop an attack that is launched for the first time using a new URL or IP address. It is possible that the blacklist strategy will not have a success rate that is higher than twenty percent [15, 16]. A blacklist service is offered by a number of different companies, including Google Safe Browsing API and PhishNet, among others. Unfortunately, the blacklist needs to be updated on a regular basis and consumes a significant amount of system resources [17].

The developers [18] made use of a straightforward algorithm in order to determine and forecast whether or not URLs are real or phishing sites. First, a URL is run through a database that contains blacklisted URLs. If the URL is already on the blacklist, then the content at that location is assumed to be malicious. In the event that it cannot be found, the attributes of the URL are extracted for the purpose of conducting further research. Finally, the URL is run through a straightforward classifier to determine whether or not it contains harmful content. B. Detection methods that are based on machine learning Approaches based on machine learning (ML) have been used successfully to identify potentially misleading URL links. Machine learning approaches this issue as a binary classification. To construct effective models for online detection, learning algorithms need to be trained using sufficient samples of true and false URLs.

The authors presented a text-based detection technique in the study [19], which collected keywords from URL links and searched for these terms using the Search engine such as google. This strategy was first introduced by the authors. This approach was cited as an example of a text-based detection method. If the text of the URL is found inside the search results, then the URL will be regarded to be authentic; otherwise, it's going to be considered to be a fake URL. The authors [20] utilised an adaptable self-structuring computational model for the purpose of classifying true URLs from bogus ones.

Natural language processing (NLP) is the methodology that was utilised by the writers [21]. They have a feature vector that has 209 words and 17 features that are based on NLP. An events denoising convolutional neural network (EDCNN) system was proposed by the authors in the publication [22] the objective of the EDCNN tool was to identify fraudulent URL sequences in proxy logs. It was used to prevent the harmful effects of these sequences on innocent websites. According to their evaluation, the tool significantly lowers the number of false alarms and helps prevent malware attacks. Browser fingerprinting is a type of security measure that prevents users from accessing exploit code after visiting a compromised website. The authors [23] used a nonlinear regression technique to determine whether a website is genuine or not in order to validate their findings. During the training phase, they employed a method that was a combination of the harmony search (HS) and the support vector machine (SVM) techniques. Natural language processing was utilised by the authors [24] in order to identify fraudulent emails. The method that was suggested involved applying semantic analysis to the text contained within emails while making use of a specified blacklist. In this paper, we propose a one-dimensional convolutional neural networks (CNN-1D) model that can identify malicious URLs. We use a benchmarked dataset in conjunction with two different evaluation measures—accuracy and area under the curve (AUC)—to assess the effectiveness of our model. The author [25] suggested the system proposes using ant colony optimization techniques to identify the most important traits that are related to phishing scams. It then compiles this information into a Bayesian classifier. Akhtar and Feng [26] proposed CNN-LSTM method was able to achieve an R2 of 99.19% in the dataset, and a correlation coefficient of 100% for the method was obtained using the provided data. The symmetry of the correlation between the CNN-LSTM and the provided dataset shows that it has the highest accuracy when it comes to detecting malware. The other classifiers had an accuracy of 99% for DT, 95% for SVM, and 99% for CNN-LSTM. The accuracy of the CNN-LSTM model is 99%. Ghaleb et al. [27] Presented a two-stage ensemble learning model was proposed that combines the random forest algorithm for preclassification and the MLP for final decision making. The new model eliminates the majority voting scheme in favor of the trained MLP. The proposed CTI-based model performed better than the traditional URL-based model in terms of accuracy and false-positive rates. Afzal et al. [28] proposed an URLdeepDetect an unsupervised and supervised mechanism that performs well in identifying and classifying web addresses. It uses LSTM and k-means clustering to achieve an accuracy of 98.3% and 99.7%, respectively.

3. METHODS FOR MALICIOUS URL DETECTION

Convolutional neural networks provide the basis of the model that our article proposes for the detection of harmful URLs. Figure 1 illustrates the model's assembled state. The model comprises three modules: a module for vector embedding, a module for dynamic convolution, and an extraction module for blocks. In the following, we will provide an in-depth discussion on each module as well as the detecting procedure.

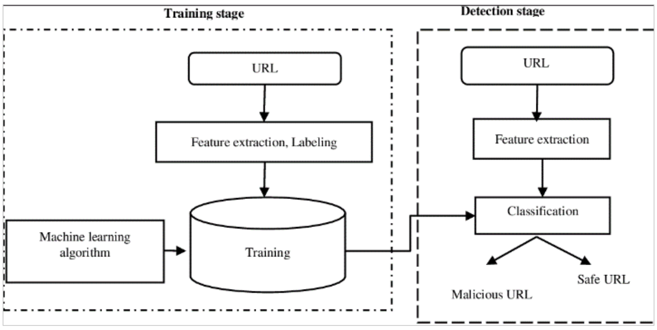

Figure 1. Architecture for detection model

3.1 Convolutional neural network (CNN) based prediction model

Convolutional neural networks perform functions that are very similar to those performed by neural networks such as the Perceptron. In this study, we tackle the problem of identifying fraudulent URLs by constructing a CNN-1D model created with Keras and a TensorFlow- GPU backend technique. Both of these methods are built on top of TensorFlow. The core concept behind CNN is similar to that of a linear neural network, which takes raw data as its input. Both of these types of neural networks are used to predict outcomes (1D vector). The first convolutional layer has 64 filters with a kernel size of 3, the second convolutional layer has 64 filters with a kernel size of 5, and then there is a concatenate layer in between the two convolutional layers. Following the completion of the concatenate layer comes the ReLU activation step. An embedding layer comes next, and then a dropout layer comes after that. The dropout layer is the one that gets the subdomains, domains, and domain suffix that were just sent in. One neuron is present in the extremely dense layer that lies on top of everything else. This makes it possible for the model as a whole to generate binary classifications (i.e., a Malicious or legitimate URL). The CNN-1D model that was used in this study can be seen shown in the figure that is located up top where it can be found.

If we wish to use deep learning, we will need to have the textual data represented by URLs transformed into the numerical data it represents. The brief explanation provides an overview of the various steps involved in preparing and representing a URL. It is important to note that the data contained in the URL has a defined syntax. Therefore, it is logical to perform data analysis using techniques such as natural language processing.

A URL is tokenized using space and punctuation, and then a dictionary composed of the first M words is created to represent the most frequently used requests.

These punctuations are contained in the vocabulary because attackers primarily use them to structure unusual requests, and the lexicon was created for that purpose. Both the value of information and the high - level semantic of URLs are preserved in this manner. In specifically, a one-hot vector is substituted for every word and punctuation mark, and several one-hot vectors are used to represent each component of a URL. The one-hot vectors represent the unique syntax of a given URL. This model is trained in deep learning to understand this unique characteristic. It then presents it in a feature presentation.

Figure 2 depicts the organisational structure of both the preprocessing and feature representation processes. Each word and punctuation mark in a URL is changed into a one-hot vector, and the feature representation will consist of each individual vector from a single URL being fed into the input layer.

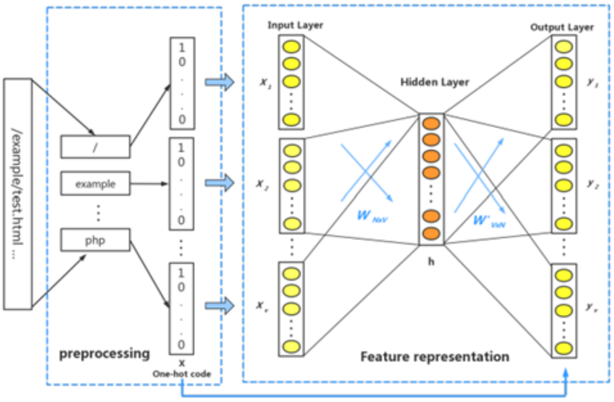

Figure 2. Pre-processing and feature representation process

The proposed network is made up of several layers, including a drop-out layer following each of these layers as well as a convolutional layer, three fully - connected, and one more connected layer. In addition to these layers, it possesses a normalisation layer, four Comp-block layers, and a final layer following each of these layers. Figure 3 depicts the structure that results when normalizing layers and drop out layers are eliminated from the structure. It is essential to stress that the feature maps that come after the convolution layer are a combination of the old feature representation that came before the convolution layer and the new feature maps that came after the convolution layer. We chose not to employ convolution layers but instead just comp-blocks in the successive layers even though comp-blocks keep the original information while convolution layers drop some useful information from the input. Despite the fact that convolution layers get such an extracted features simply from the input and drop some valuable information from the input, we decided not to use convolution layers because we wanted to keep the original data.

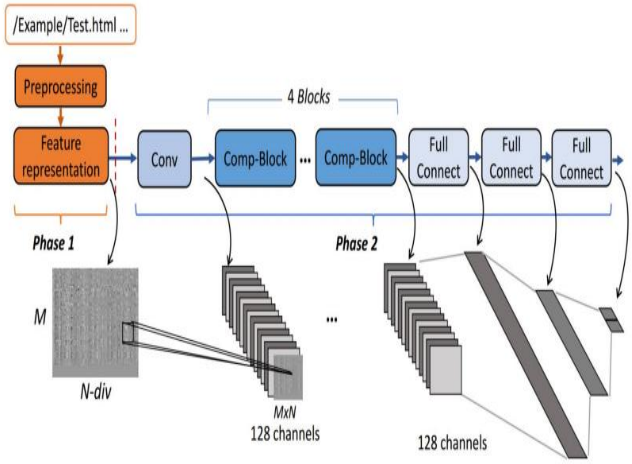

Figure 3. Architecture of CNN

4. EXPERIMENTAL RESULTS

Experiments are run with a dataset of malicious URLs that is extensively used for online attack detection, and these are used so that the suggested system may be evaluated. In particular, we conduct all of our studies in an environment consisting of a computer with an Intel Core i5-8900k processor, 8GB DDR4 memory, and Windows 10. Following that, there will be a brief discussion followed by an introduction to the datasets that used and the experimental outcomes.

4.1 Dataset

For the experiment we have used the dataset obatined from kaggle, consisting of 11055 instances and 31 attriutes.

4.2 Evaluation parameters

Precision – The percentage of positive neural model predictions is computed by considering the total number of predicted positive instances. This is done by taking into account the whole dataset.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Recall – The percentage of positive instances in the dataset is computed by taking into account the number of actually positive instances. This is done to find out how much extra correct ones the model showed.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

F1-Score – The harmonic mean of recall and precision is a measure of how well a model performs in the F1 score. It takes into account the contribution of both, so a higher score is better. For instance, if a model can accurately predict the positive outcomes of a given event, it can outperform a model that cannot accurately predict the negative outcomes.

$$F1 - Score = \frac{2\ *\ Precision\ *\ recall}{Precision\ * Recall} \tag{3}$$

Accuracy - It is the score that is generated when the class is generalised.

The model's ability to generalize appropriately.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

Confusion Matrix - A Confusion matrix represents a classification model's performance that shows how it performed against the predicted targets. It compares the model's actual values with those of the machine learning model.

5. DISCUSSION AND CONCLUSION

Figure 4 shows the confusion matrix of good and bad URLs. Figure 5 shows the evaluation parameters of various algorithm where CNN outperforms with the accuracy of 99.99%. Figure 6 shows the CNN model accuracy graph. Finally, Figure 7 shows the model loss, which indicates the as no. of epoch increase loss tends to almost zero, indicating efficient execution of algorithm. Figure 8 and Figure 9 represents the count of URL as benign or malicious and as good or bad respectively.

The number of positive instances in the dataset is computed by taking into account the number of actually positive instances. This is done to find out how much extra correct ones the model showed. The rise of computer and system technologies has made it easy for people to exchange information online. Due to the convenience of these technologies, people are more likely to exchange information about their daily lives. This includes their passwords and other personal information. Most network applications are aware of their users' behavior. Due to the rapid growth of web pages and applications, they have become the primary targets for attackers. There has been a significant increase in the number of malicious websites.

The users who are most vulnerable to these types of harmful web pages are those who are not aware of what's happening on the Internet. Attackers can easily take advantage of this by embedding or uploading malicious code on the page. According to Google, over 10% of all web pages have malicious code. Due to the increasing number of web pages and applications, the authorities and the users must be aware of the warning signs of these threats. This is because determining if a particular web page is being used for malicious activities can help prevent the exploitation of these threats. In the past few years, malicious websites have increased significantly. As a result, they have become a severe threat to the security of network applications.

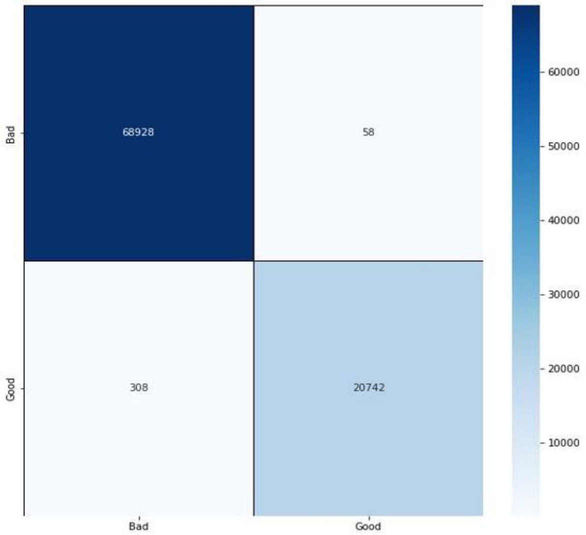

Figure 4. Confusion matrix

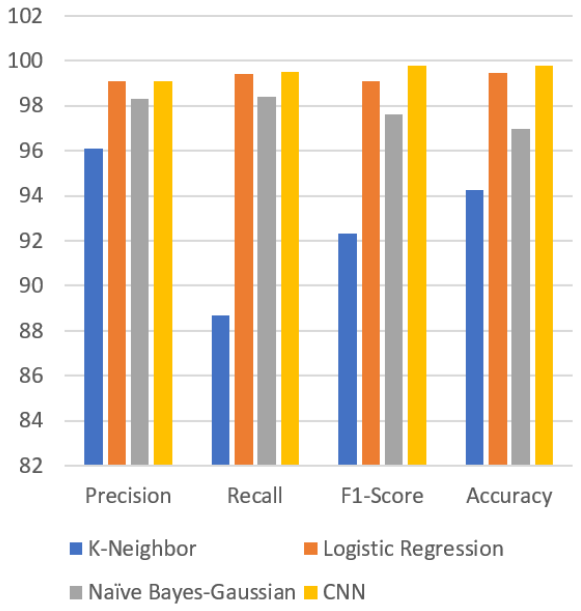

Figure 5. Evaluation parameters

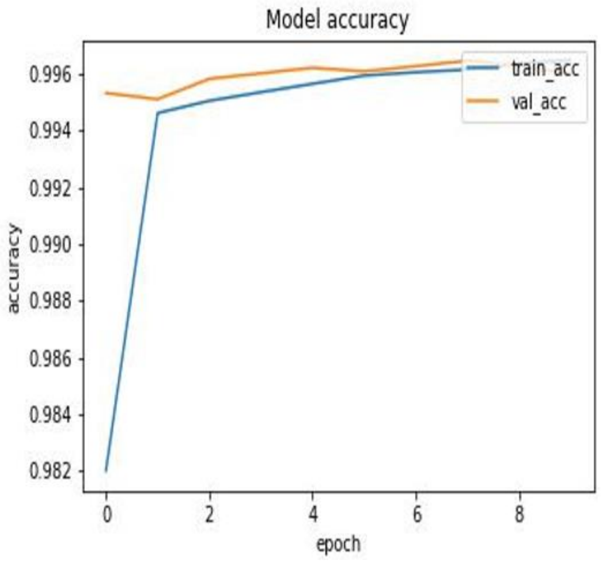

Figure 6. CNN model accuracy

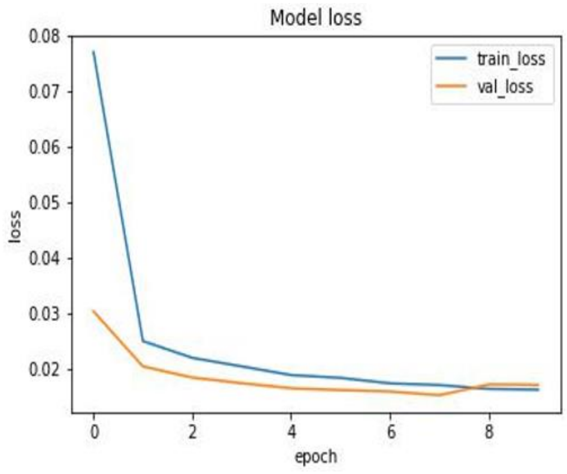

Figure 7. Model loss

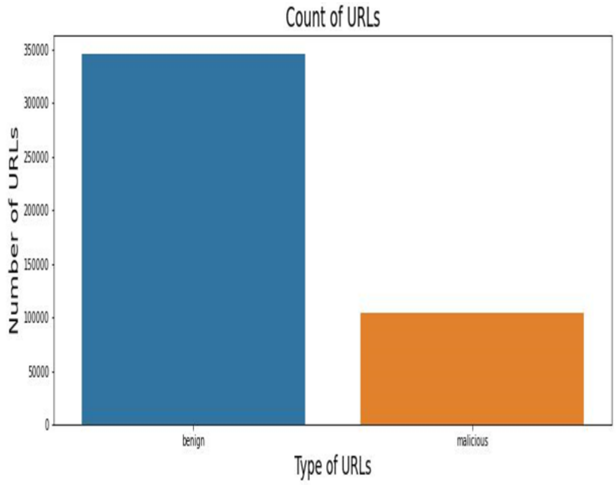

Figure 8. Count of URL as benign or malicious

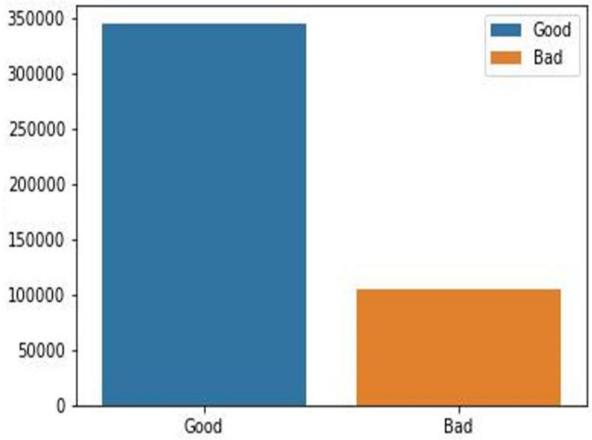

Figure 9. Number of instances classified as good or bad URL

REFERENCES

[1] Lemay, D.J., Basnet, R.B., Doleck, T. (2020). Examining the relationship between threat and coping appraisal in phishing detection among college students. Journal of Internet Services and Information Security, 10(1): 38-49. https://dx.doi.org/10.22667/JISIS.2020.02.29.038

[2] Kim, H. (2020) 5G core network security issues and

attack classification from network protocol perspective. Journal of Internet Services and Information Security, 10(2): 1-15. https://doi.org/10.22667/JISIS.2020.05.31.001

[3] Aram, K., SoK, J.O. (2019). A systematic review of insider threat detection. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), 10(4): 46-67.

[4] Basnet, R.B., Shash, R. (2019). Towards detecting and classifying network intrusion traffic using deep learning frameworks. Journal of Internet Services and Information Security, 9(4): 1-17. http://dx.doi.org/10.22667/JISIS.2019.11.30.001

[5] Valenza, F., Cheminod, M. (2020). An optimized firewall anomaly resolution. Journal of Internet Services and Information Security, 10: 22-37.

[6] Patil, D.R., Patil, J.B. (2015). Survey on malicious web pages detection techniques. International Journal of U- and E-Service, Science and Technology, 8(5): 195-206. https://dx.doi.org/10.14257/ijunesst.2015.8.5.18

[7] Yu, B., Pan, J., Hu, J., Nascimento, A., De Cock, M. (2018). Character level based detection of DGA domain names. Proceedings of the International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, pp. 1-8.

[8] Choudhary, C., Sivaguru, R., Pereira, M., Yu, B., Nascimento, A., De Cock, M. (2018). Algorithmically generated domain detection and malware family classification. Proceedings of the International Symposium on Security in Computing and Communication, Singapore, pp. 640-655. http://dx.doi.org/10.1007/978-981-13-5826-5_50

[9] Garera, S., Provos, N., Chew, M. (2007). A framework for detection and measurement of phishing attacks. Proceedings of the ACM Workshop on Recurring Malcode, ACM, New York, NY, USA, http://dx.doi.org/10.1145/1314389.1314391.

[10] Gupta, D.K.M.M. (2008). Behind phishing: an examination of phisher modi operandi. Proceedings of the Usenix Workshop on Large-scale Exploits & Emergent Threats, DBLP, San Francisco, CA, USA, pp. 1-8.

[11] Ma, J., Saul, L.K., Savage, S. (2009). Beyond blacklists: learning to detect malicious web sites from suspicious URL. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Paris, France. http://dx.doi.org/10.1145/1557019.1557153

[12] Choi, H., Zhu, B.B., Lee, H. (2011). Detecting malicious web links and identifying their attack types. Proceedings of the 2nd USENIX conference on Web application development, Boston, MA, USA, pp. 125-136.

[13] Bartos, K., Sofka, M., Franc, V. (2016). Optimized invariant representation of network traffic for detecting unseen malware variants. Proceedings of the USENIX Security Symposium, Austin, TX, USA, pp. 807-822.

[14] Jain, A.K., Gupta, B.B. (2016). A novel approach to protect against phishing attacks at client side using auto-updated white-list. EURASIP Journal on Information Security, 2016(1): 9.

[15] Khonji, M., Iraqi, Y., Jones, A. (2013). Phishing detection: A literature survey. IEEE Communications Surveys Tutorials, 15(4): 2091-2121. https://doi.org/10.1109/SURV.2013.032213.00009

[16] Sheng, S., Holbrook, M., Kumaraguru, P., Cranor, L.F., Downs, J. (2010). Who falls for phish?: A demographic analysis of phishing susceptibility and effectiveness of interventions. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI ’10. New York, NY, USA, pp. 373-382.

[17] Sheng, S., Wardman, B., Warner, G., Cranor, L., Hong, J., Zhang, C. (2009). An empirical analysis of phishing blacklists. International Conference on Email and Anti-Spam. https://doi.org/10.1184/R1%2F6469805.V1

[18] Abdi, F.D., Wenjuan, L. (2017). Malicious URL detection using convolutional neural network. Journal International Journal of Computer Science, Engineering and Information Technology, 7(6): 1-8. https://doi.org/10.1155/2021/5518528

[19] Zhang, Y., Hong, J.I., Cranor, L.F. (2007). Cantina: A content-based approach to detecting phishing web sites. In Proceedings of the 16th International Conference on World Wide Web, WWW ’07, ACM, New York, NY, USA, pp. 639-648. http://dx.doi.org/10.1145/1242572.1242659

[20] Mohammad, R.M., Thabtah, F., McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. Neural Computing and Applications, 25(2): 443-458. http://dx.doi.org/10.1007/s00521-013-1490-z

[21] Buber, E., Dırı, B., Sahingoz, O.K. (2017). Detecting phishing attacks from URL by using NLP techniques. In 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, pp. 337-342. https://doi.org/10.1109/UBMK.2017.8093406

[22] Shibahara, T., Yamanishi, K., Takata, Y., Chiba, D., Akiyama, M., Yagi, T., Ohsita, Y., Murata, M. (2017). Malicious URL sequence detection using event denoising convolutional neural network. In 2017 IEEE International Conference on Communications (ICC). Paris, France, pp. 1-7. https://doi.org/10.1109/ICC.2017.7996831

[23] Babagoli, M., Aghababa, M.P., Solouk, V. (2018). Heuristic nonlinear regression strategy for detecting phishing websites. Soft Computing, 23(12). https://link.springer.com/article/10.1007/s00500-018-3084-2

[24] Peng, T., Harris, I., Sawa, Y. (2018). Detecting phishing attacks using natural language processing and machine learning. 2018 IEEE 12th International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, pp. 300-301. https://doi.org/10.1109/ICSC.2018.00056

[25] Sapkal, V., More, N. (2021). An improved classification model for identifying the phishing attacks. Webology, 18.

[26] Akhtar, M.S., Feng, T. (2022). Detection of malware by deep learning as CNN-LSTM machine learning techniques in real time. Symmetry (Basel)., 14(11). https://doi.org/10.3390/sym14112308

[27] Ghaleb, F.A., Alsaedi, M., Saeed, F., Ahmad, J., Alasli, M. (2022). Model using ensemble learning. Sensors, pp. 1-20.

[28] Afzal, S., Asim, M., Javed, A.R., Beg, M.O., Baker, T. (2021). URLdeepDetect: A deep learning approach for detecting malicious URLs Using semantic vector models. Journal of Network and Systems Management, 29(3): 1-27. https://doi.org/10.1007/s10922-021-09587-8