A CNN-Based Model for Detecting Malicious URLs

Conference Paper · December 2023

DOI: 10.1109/RIVF60135.2023.10471782

3 authors, including:

Dau Hoang
Posts and Telecommunications Institute of Technology
39 PUBLICATIONS **600** CITATIONS
SEE PROFILE

Ninh Thị Thu Trang
Posts and Telecommunications Institute of Technology
5 PUBLICATIONS **12** CITATIONS
SEE PROFILE

A CNN-Based Model for Detecting Malicious URLs

Xuan Dau Hoang
Faculty of Information Security
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
dauhx@ptit.edu.vn

Dang Le Minh
Faculty of Postgraduate Studies
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
leminhdang9801@gmail.com

Thi Thu Trang Ninh
Faculty of Information Security
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
trangntt2@ptit.edu.vn

Abstract*—Malicious URLs are one of the major security threats to organizations and end-users on the Internet because they are associated with various types of attacks and misuses, such as spreading malwares and launching phishing attacks. Because of serious threats of malicious URLs, there have been several proposed approaches for detecting and preventing malicious URLs, including backlisting and machine learning-based approaches. Following machine learning-based approaches, this paper proposes a model for detecting malicious URLs based on the CNN deep learning technique. The aim of the proposed CNN-based URL detection model is to increase detection rate as well as to reduce false alarm rate. Experiments on the dataset of 651,191 labelled URLs show that our proposed model produces a high F1-score of 98.99%.

Keywords*—Malicious URL detection, Malicious URL detection based on machine learning, CNN-based malicious URL detection.

I. INTRODUCTION

URL (Uniform Resource Locator) is a type of addresses that is used to refer to a resource on the Internet. Basically, the structure of URLs consists of three components, including protocol, domain name and path [1]. The protocol, such as HTTP, HTTPS or FTP is used as the primary access medium. The domain name or the IP address is used to specify the location of the resource. The URL's final part is the path to a specific page or file within the domain [1][2].

URLs are very commonly used when accessing Internet resources. A user can access an Internet resource by typing an URL directly into the address bar of a web browser or by clicking at a hyperlink in a web page, an email or from another application. Due to the popular use of URLs on the Internet, attackers often create malicious URLs to trick users accessing their 'bad' sites. The malicious URLs are usually embedded in the form of hyperlinks in web pages, emails and messages of instant messengers, such as Facebook messenger. In practice, malicious URLs are frequently used by attackers to launch phishing attacks, spread various types of malwares and trick users to visit malicious web pages or 'inappropriate' contents. According to statistics by Hu et al. [3], over 67 million malicious URLs and domain names were found in second half of 2022, an increase of more than 52% over the first half of 2022. There are four typical categories of malicious URLs and domain names, which are malware, phishing, grayware and botnet command-and-control. Among them, malware and phishing URLs are most common URLs used by attackers [3][4][5].

Due to serious threats of malicious domain names and URLs, there have been several proposed approaches for detecting and preventing malicious domain names and URLs. These approaches can be divided into two major categories: (i) blacklisting or heuristics approaches and (ii) machine learning-based approaches [2]. The blacklisting approach is the most common and widely used technique for detecting and preventing malicious URLs. In this approach, a list of known malicious URLs is maintained and used. If a visited URL is found in the list, it is considered malicious and an alarm will be generated. Otherwise, the URL is assumed to be benign. The advantage of the blacklisting approach is fast and it can be used to detect known malicious URLs efficiently. However, it is not able to detect new malicious URLs because they are not in the list. In addition, it is impossible to maintain a large list of all malicious URLs because it is easy to generate new URLs.

The heuristic approach is an extension of the blacklisting approach, in which a database of attack/intrusion signatures is created and used instead of the list of malicious URLs. An attack signature can have several elements, including strings and IP addresses. Attack/intrusion signatures are commonly used in intrusion detection systems (IDS) for detecting various types of attacks. Similar to the blacklisting approach, the heuristic approach is not able to detect new attacks or intrusions because their signatures are not in the database. Moreover, it is also very difficult to maintain the database of attack/intrusion signatures [2].

On the other hand, the machine learning-based approach first tries to analyze benign and malicious URLs' information and/or their corresponding websites or webpages to extract features in order to construct training data. Next, the train data is used to build a classifier or profile using a machine learning algorithm. The machine learning algorithms can be traditional supervised algorithms, such as Naïve Bayes, decision tree and random forest, or deep learning techniques, such as CNN, RNN and LSTM. Then, the classifier or profile is used to classify or detect a new URL into benign or malicious [2][6]. The machine learning-based approach has been an extensive attention of the research community because it has the potential of detecting new malicious domain names and URLs. In addition, the classifier or profile can be built automatically using the training data, therefore reducing the requirement of man-labor for maintaining URL blacklists or signature databases.

In this paper, we propose a model for detecting malicious domain names and URLs based on the CNN deep learning technique. The CNN deep learning technique has been widely used and proven to give good performance in solving many computer science problems, such as natural language processing, image processing and recognition [2][6]. The aim of the proposed CNN-based URL detection model is to increase detection rate as well as to reduce false alarm rate. The rest of this paper is organized as follows: Section II reviews some related works and Section III presents the

proposed detection models. Section IV describes experimental datasets, then the data preprocessing and model training and validation, some experimental results and discussion. Section V is the conclusion of the paper.

II. Related Works

This section reviews some recent proposals that are closely related to our work, including Hoang et al. [7][8], Zhao et al. [9], Cho et al. [10], Crisan et al. [11] and Wang et al. [12]. Hoang et al. [7] proposes a machine learning-based model for detecting botnets using DNS query analysis. The paper uses supervised machine learning techniques, such as Naive Bayes, decision tree and random forest to construct detection models for the classification of normal domain names and malicious domain names generated and used by botnet malwares. The proposed model extracts 18 features for each domain name, including 16 n-gram statistical features, a domain vowel distribution feature and a domain character entropy feature. Experimental results confirm that most machine learning-based models generates high accuracy, in which the random forest-based model produces the highest accuracy of over 90% and lowest false alarm rate. The drawbacks of the proposed model are (i) it can only detect malicious domain names generated by character-based DGA botnet malwares and (ii) its false alarm rate is fairly high.

In a similar direction, Hoang et al. [8] proposes an extension of [7] using random forest machine learning algorithm for detecting malicious domain names generated by character-based DGA botnet malwares. The aim of this proposal is to increase the detection accuracy and to reduce the false alarm rate. The proposed model uses a set of 24 features for each domain name, including 16 n-gram statistical features, 6 statistical features of vowels, consonants and digits, a domain character entropy feature, and an expected value of the domain name. Experiments on the dataset of 100,000 normal domain names and 153,000 DGA domain names show that the proposed model gives high detection accuracy of over 97% and low false alarm rate of about 3%. However, the model's major shortcoming is it can only detect malicious domain names generated by character-based DGA botnet malwares, not general URLs.

Using another approach, Zhao et al. [9] proposes a statistical method for detecting malicious domain names using n-gram techniques. Each domain name in the training set of legitimate domains is first divided into substring sequences using 3, 4, 5, 6 and 7-gram technique. Then, the statistics and weight values of substrings of all training domains are calculated to construct the 'profile'. To validate an input domain name if it is legitimate or malicious, the domain name is also first divided into substring sequences using 3, 4, 5, 6 and 7-gram technique. Then, the statistics of domain name substrings are calculated and then it is used to compute the 'reputation value' of the domain name based on the 'profile'. A domain reputation threshold is generated for each category of malicious domain names using the 'profile'. If the domain name's reputation value is greater than the threshold, it is legitimate. Otherwise, it is malicious. Experimental results show that the proposed approach achieves the detection accuracy of 94.04%. However, the detection performance of the proposed approach heavily depends on the selection of the domain reputation threshold that is currently generated and selected manually. Furthermore, it can only detect malicious domain names, not general URLs.

Similar to Hoang et al. [7][8], Cho et al. [10] proposes a method to detect malicious URLs using machine learning techniques. The paper proposes to use 54 features in 3 groups of lexical group, host-based group and correlated group for detection model training and validation. Two machine learning techniques, including SVM and random forest are used to construct and validate the proposed detection model. Experiments on 470,000 URL dataset shows that the random forest-based model generates the highest detection accuracy of 96.28%. The advantage of the proposed method is high detection accuracy. However, many features in host-based group are not available because a large number of URLs are offline, hence this may affect the detection performance.

Crisan et al. [11] proposes a method to detect malicious URLs based on machine learning techniques and word embeddings. The paper uses a vector of 335 features, including 300 word-embedding features, 11 attack-word features, 8 Linux-command features and other features to represent an URL. Three machine learning techniques, such as cost sensitive neural network (CSNN), multilayer-perceptron (MLP) and extra trees are used to construct and validate the detection model. Experiments show that the CSNN-based model produces the best overall detection results with the accuracy, precision and recall of 90.11%, 99.68% and 89.24%, respectively. The advantage of the proposed method is the high detection precision. Although the proposed method's precision is high, its overall accuracy is not high because its recall is not high either. In addition, a large feature vector of 335 features may create a large and slow model.

Based on deep learning techniques, Wang et al. [12] proposes a model for detecting malicious URLs based on dynamic convolutional neural network (DCNN). DCNN is a modified version of CNN, in which a new folding layer is added to the original multi-layer convolution network. Experimental results on the dataset of 400,000 URLs confirm that the proposed model produces a high detection accuracy of 98.70% and a high F1-score of 98.70%. The advantages of the proposed method are it generates high detection performance and the detection model can be built automatically from the training data.

III. Proposed URL Detection Model

A. The CNN-based Malicious URL Detection Model

Fig. 1 shows the malicious URL detection model using CNN deep learning. The CNN deep learning technique is selected because it has been widely used and proven to give good performance in many applications of computer science, such as natural language processing, image & video processing and recognition [2][6][12][13]. The model has 4 inputs, including the input URL, domain, subdomain and suffix domain. The domain, subdomain and suffix domain are extracted from the URL. While the input URL is passed through an embedding layer, convolution layers, a concatenate layer and a flatten layer, other inputs are passed through an embedding layer and a reshape layer. Then, the results of all 4 inputs are concatenated using a concatenate layer, followed by two groups of dropout-dense layers.

Specifically, the processing of the detection model is as follows:

1) Inputs

- The first input is the URLs that are padded to the same length and then tokenized;

- The next inputs are domains, subdomains and domain suffixes, which are extracted from input URLs.

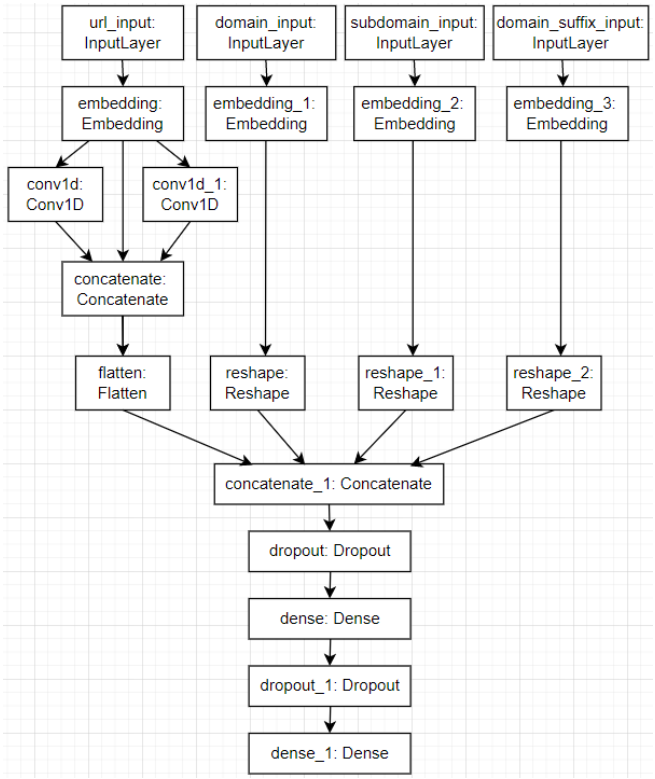

Fig. 1. The CNN-based Malicious URL Detection Model

2) Embedding layers

- The input URL is passed through an embedding layer that converts the URL into a fixed length vector;
- Other inputs of domains, subdomains and domain suffixes are also passed through an embedding layer that converts the them into fixed length vectors.

3) Convolutional layers

- After the embedding layer, the input URL is passed through the first convolutional layer (conv1d) with 64 filters and the kernel size of 3;
- The input URL is also passed through the second convolutional layer (conv1d_1) with 64 filters and the kernel size of 5;
- The purpose of using convolutional layers is to learn local features of input URLs.

4) Flatten/Reshape layer

- After convolutional layers, the results of input URLs are flattened;
- After the embedding layer, the results of domains, subdomains and domain suffixes are reshaped;
- This layer's purpose is to create 1-dimention vector that represents the input URL.

5) Concatenate layers

- The results of URL convolutional layers are concatenated using a concatenate layer to make a single output;
- The outputs of the URL flatten layer and reshape layers of domains, subdomains and domain suffixes are concatenated using a concatenate layer to make a vector that represents the whole input URL.

6) Fully connected layers

- The output of the concatenate layer (concatenate_1) is passed through a dropout layer (dropout) and then a fully connected layer (dense) with 128 neurons;
- The ReLU activation function is used to generate non-negative values and achieves non-linear model.

7) Ouput layer

- The output of the fully connected layer (dense) is passed through another dropout layer (dropout_1) and then the final fully connected layer (dense_1) with 1 neuron;
- The Sigmoid activation function is used to compute the probability of the URL output to be 'malicious' or 'safe'.

B. Performance Measures

In order to evaluate the performance of the proposed detection models, the precision, recall and F1-score are used. These measures are computed as follows:

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (2)$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% \quad (3)$$

where TP is the number of "malicious" URLs that are correctly classified, TN is the number of "safe" URLs that are correctly classified, FP is the number of "safe" URLs that are wrongly classified as "malicious" and FN is the number of "malicious" URLs that are wrongly classified as "safe".

IV. Experiments and Results

A. Experimental Dataset

The experimental data used in this paper includes 651,191 labelled URLs [14], including 428,103 'benign' URLs, 96,457 'defacement' URLs, 94,111 'phishing' URLs and 32,520 'malware' URLs. TABLE I. presents some URLs with their labels in the experimental data and TABLE II. shows components of experimental dataset. The 'benign' URLs are re-labelled as 'safe' and the 'defacement', 'phishing' and 'malware' URLs are grouped and re-labelled as 'malicious' URLs. Fig. 2 shows the ratio of 'safe' URLs (65.7%) and 'malicious' URLs (34.3%) in the experimental data. The preprocessed data is divided into 2 sets: 80% of the data to the TrainSet and the rest of 20% to the TestSet. The TrainSet is used for training to build the detection model and the TestSet is used to validate the model performance.

TABLE I. URL Samples of Experimental Data

URL	Label
br-icloud.com.br	phishing
mp3raid.com/music/krizz_kaliko.html	benign
bopsecrets.org/rexroth/cr/1.htm	benign
http://www.garage-pirenne.be/index.php?option=com_content&view=article&id=70&vsig70_0=15	defacement
http://www.824555.com/app/member/SportOption.php?uid=guest&langx=gb	malware
http://www.kingsmillshotel.com/spring/mothers-day	defacement
retajconsultancy.com	phishing
parsippanysoccerclub.org/	benign

TABLE II. COMPONENTS OF EXPERIMENTAL DATASET

Dataset components	Number of URLs
'benign' URLs	428,103
'defacement' URLs	96,457
'phishing' URLs	94,111
'malware' URLs	32,520
Total	**651,191**

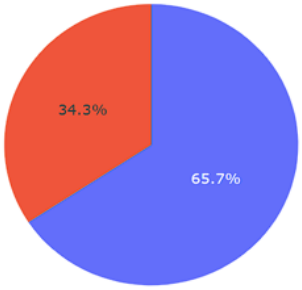

Fig. 2. The ratio of 'safe' URLs (65.7%) and 'malicious' URLs (34.3%) in the experimental data

B. Data Preprocessing and Model Training & Validation

The URL experimental data is preprocessed to create training and testing data for model construction and validation. The preprocessing step includes the following tasks:

- Extract domains, subdomains and domain suffixes from the URLs;
- The domains, subdomains and domain suffixes are encoded to integer values;
- The 'safe' label is converted to 0 and the 'malicious' label is converted to 1;
- Input URLs are padded to the same length and then tokenized to be the inputs of the CNN-based deep learning model.

After the preprocessing, the TrainSet and the TestSet are created using 80% and 20% of the experimental data, respectively. Then, the TrainSet and TestSet are used to build and validate the detection model using the CNN deep learning technique, as presented in Fig. 1. The training process is planned to run 25 rounds on training data. However, the training process is stopped at 13 rounds because there is almost no improvement in the model's performance.

C. Experimental Results

TABLE III. shows the precision, recall and F1-score of the proposed CNN-based malicious URL detection model on 13 execution rounds. The training process stops at 13th round because there is little improvement in performance measures. At 13th round, the precision, recall and F1-score of the proposed model are very high at 99.26%, 98.73% and 98.99%, respectively. TABLE IV. presents the model's detection results for 20 real URLs: there are 17 URLs correctly detected and 3 URLs wrongly detected. The reasons for this are the number of tested URLs is relatively small and the coverage of the training dataset is not sufficient in order to produce high detection accuracy in all URL areas.

TABLE III. PERFORMANCE MEASURES OF PROPOSED CNN-BASED MODEL

Execution Round	Precision (%)	Recall (%)	F1-score (%)
1	95.40	92.63	93.99
2	97.48	95.89	96.68
3	98.02	96.69	97.35
4	98.38	97.27	97.82
5	98.62	97.66	98.14
6	98.80	97.92	98.36
7	98.89	98.14	98.51
8	98.97	98.27	98.62
9	99.09	98.38	98.73
10	99.13	98.54	98.83
11	99.17	98.61	98.89
12	99.22	98.68	98.95
13	**99.26**	**98.73**	**98.99**

TABLE IV. PROPOSED MODEL'S RESULT ON 20 REAL URLS

No.	URL	Correct Label	Detected Label
1	https://google.com	Safe	Safe
2	https://vnexpress.com.vn	Safe	Safe
3	http://www.xacnhanvay247.com	Malicious	Malicious
4	http://www.downloadappios.com	Malicious	Malicious
5	https://lolesports.com/	Safe	Safe
6	http://giaoducthoidai.vn	Safe	Safe
7	https://stackjava.com/	Safe	Safe
8	http://www.baohungyen.org.vn/	Safe	Safe
9	http://www.my-acb-bank.com/	Malicious	Malicious
10	http://phimvietnam610.ddns.net/	Malicious	Malicious
11	http://www.viet69xlxx004.ga/	Malicious	Malicious
12	http://www.dantri.com	Safe	Malicious
13	https://openai.com/blog/chatgpt	Safe	Safe
14	http://www.vietinbank-ipay.com/	Safe	Malicious
15	https://www.ff.sieuhack.work	Malicious	Malicious
16	https://www.khoataikhoanhack-garena.xyz/	Malicious	Malicious
17	https://moneygram-ripple24-7.weebly.com	Malicious	Safe
18	http://www.thanhtra.gov.vn /	Safe	Safe
19	https://rikadv469.cpctvn.com	Malicious	Malicious
20	https://app.memrise.com/	Safe	Safe

D. Discussion

In this section, we compare our malicious URL detection model with previous malicious URL/domain name detection methods. TABLE V. shows the performance comparison of our proposed model 6 previous models. It can be seen that our CNN-based model performs better than all given previous models. Although Crisan et al. [11] has the highest precision of 99.68%, its F1-score of 94.17% is not high because its recall is less than 90%. On overall, our model produces higher F1-score compared to that of top 3-previous proposals, including Crisan et al. [11], Hoang et al. [8] and Wang et al. [12]. F1-scores of our model, Crisan et al. [11], Hoang et al. [8] and Wang et al. [12] are 98.99%, 94.17%, 97.03% and 98.70%, respectively.

TABLE V. PERFORMANCE COMPARISON OF PROPOSED MODELS AND PREVIOUS DETECTION MODELS

Detection Model	Precision (%)	Recall (%)	F1-score (%)
Hoang et al. [7]	90.70	91.00	90.80
Hoang et al. [8]	97.08	96.98	97.03
Zhao et al. [9]	93.86	92.58	93.21
Cho et al. [10]	91.44	94.42	92.90
Crisan et al. [11]	99.68	89.24	94.17
Wang et al. [12]	99.30	98.10	98.70
Our CNN-based model	**99.26**	**98.73**	**98.99**

V. CONCLUSION

This paper proposes a model for detecting malicious URLs using CNN deep learning method. Our malicious URL detection model aims at improving the detection accuracy. Our experimental results confirm that our CNN-based malicious URL detection model outperforms all listed previous proposals on most performance measures. Specifically, the precision, recall and F1-score of our model and top 2-previous proposals (Hoang et al. [8], Wang et al. [12]) are 99.26%, 98.73%, 98.99%, 97.08%, 96.98%, 97.03% and 99.30%, 98.10%, 98.70%, respectively.

For future work, we will test our model with a large URL dataset and explore advanced deep learning techniques to build better detection models, such as LSTM and BiLSTM.

ACKNOWLEDGMENT

Authors sincerely thank the Cyber Security Lab, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam for the facility support to complete this research project.

REFERENCES

[1] Scarpati, J., Burke, J.: URL (Uniform Resource Locator), https://www.techtarget.com/searchnetworking/definition/URL, Accessed in June 2023.

[2] Sahoo, D., Liu, C., Hoi, S.C.H.: Malicious URL Detection using Machine Learning: A Survey. ArXiv, 2019, https://arxiv.org/pdf/1701.07179.pdf

[3] Hu, C. et al.: Recent Trends in Internet Threats: Common Industries Impersonated in Phishing Attacks, Web Skimmer Analysis and More, https://unit42.paloaltonetworks.com/internet-threats-late-2022/, Apr 2023.

[4] Rushton, J.: 50+ Phishing Statistics You Need to Know – Where, Who & What is Targeted, https://www.techopedia.com/phishing-statistics, July 2023.

[5] Cook, S.: Phishing statistics and facts for 2019–2023, https://www.comparitech.com/blog/vpn-privacy/phishing-statistics-facts/, June, 2023.

[6] Aljabri, M. et al., Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions, IEEE Access, vol. 10, pp. 121395-121417, 2022, doi: 10.1109/ACCESS.2022.3222307.

[7] Hoang, X.D., Nguyen, Q.C.: Botnet Detection Based on Machine Learning Techniques Using DNS Query Data. J. Future Internet 2018, 10, 43; doi:10.3390/fi10050043.

[8] Hoang, X.D., Vu, X.H.: An improved model for detecting DGA botnets using random forest algorithm, Information Security Journal: A Global Perspective, July 2021, DOI: 10.1080/19393555.2021.1934198.

[9] Zhao, H., Chang, Z., Bao, G., Zeng, X.: Malicious Domain Names Detection Algorithm Based on N-Gram. Journal of Computer Networks and Communications. Vol. 2019. Hindawi; doi: 10.1155/2019/4612474.

[10] Cho, D.X., Hoa, N.D., Nikolaevich, T.V.: Malicious URL Detection based on Machine Learning. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 1, 2020.

[11] Crişan, A., Florea, G., Halasz, L., Lemnaru, C., Oprisa, C.: Detecting Malicious URLs Based on Machine Learning Algorithms and Word Embeddings, 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 2020, pp. 187-193, doi: 10.1109/ICCP51029.2020.9266139.

[12] Wang, Z., Ren, X., Li, S., Wang, B., Zhang, J., Yang, T.: A Malicious URL Detection Model Based on Convolutional Neural Network. In special issue 'Communication Security in Socialnet-Oriented Cyber Spaces', Security and Communication Networks, Wiley, 2021. https://doi.org/10.1155/2021/5518528.

[13] Saleem, R.A., Madhubala, R., Rajesh, N., Shaheetha, L., Arulkumar N.: Survey on Malicious URL Detection Techniques, 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2022, pp. 778-781, doi: 10.1109/ICOEI53556.2022.9777221.

[14] Siddhartha, M.: Malicious URLs dataset, https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset. Accessed in June 2023.