2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT) | 978-1-6654-4032-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/3ICT53449.2021.9581880

Malicious URL Detection using Multilayer CNN

Ashish Singh
School of Computer Engineering,
KIIT Deemed to be University,
Bhubaneswar-751024, Odisha (India)
ashish.singhfcs@kiit.ac.in

Pradeep Kumar Roy
Department of Computer Science and Engineering,
Indian Institute of Information Technology Surat,
SVNIT Campus. Ichchanath, Surat - 395007
pkroynitp@gmail.com

***Abstract*—Due to developing Internet-based technologies, the number of online domains and URLs is increasing globally. Parallel several cybersecurity threats and phishing attacks have been encountered while accessing these websites. Accessing a malicious webpage can create serious harm to the physical system. Data loss, privacy breach, credential theft and many security threats are entered while an Internet user clicks a malicious URL. Several defence and detection strategies have been proposed in the previous research works. But, the works have used a traditional classifier which is not adequate. This is because the size of the URL is huge, and the URL patterns are changed over time so finding the correlation between old and new patterns is almost impossible. Hence, this paper proposed malicious URL detection using multilayer Convolutional Neural Network (CNN). The proposed model first considered one layer of CNN. After that, to improve the accuracy, a two-layer of CNN will be used. The achieved result illustrated that malicious website detection accuracy is enhanced 89% to 91% when the model uses two layers of CNN.**

***Keywords*—Convolutional Neural Network, Deep learning, Malicious Website Detection, Malicious URL.**

I. Introduction

Nowadays, with the rapid development of information and communication technology, the Internet is an essential part of human life. All the business and daily life services are shift online and integrate remote storage facilities. People can access these services and perform online activities via the Internet by typing the URL in their browser. As the system connected with the Internet, they become more vulnerable to a wide range of cyber threats and attacks. Most of the time, people trust the website and enter their private information on the website without being fake or real. A fake or malicious website enables cybercriminal activities that lead to a ransomware attack, steal credentials, and financial fraud. These cybercriminal activities, the main root is the social engineering-based Phishing attack. According to the security report released by Microsoft in 2018 [1], phishing attack [2], [3] is the first choice of any attacker to start the attack on the system. In the phishing attack, the attacker builds a primary attack vector by designing a malicious website. This malicious website looks similar to the target/original website. When a legitimate user clicks on this malicious website, the user is redirected to the malicious website, and he feels the website is authentic and original. After entering the credentials or private information, the attacker analyses this information and can control the victim's computer. Thus, protect the system from phishing attack is the primary goal of any security system.

Several research works have been published in the previous years to protect the system from phishing attack as well as detect the malicious website/URL [4]–[9]. Such protection mechanism includes heuristic methods, content-based classification method, machine learning-based methods, feature engineering approach, and data mining approach. Nowadays, researchers move on to advanced detection approaches like deep learning methods. But, in the case of continuous growth in phishing attacks, steady rise in phishing sites, and introducing zero-day phishing attacks, these systems are also not performed well. Most of the research works identification technique depends on the manual rules, and the setting is entirely based on the human mind [10]. The setting may be biased if the authorized user acts as a malicious insider. The literature also identified the excessive amount of URL data and frequently changing pattern as a challenging problem while the detection system is developed. Machine learning and deep learning-based automated system can avoid human interference. But, parallelly, we need to address the computation cost and detection accuracy. Additionally, several works [11]–[13] are also identified that are based on the non-machine learning technique like reputation of the domain, ranking system, and blacklisted of malicious URL. But, these techniques are time-taking and does not provide promising accuracy.

The above discussion conclude two important points: 1) The previous available research works have some limitations, so it cannot fulfil the security requirements in current time and 2) to protect the online websites from the phishing attack, an automated detection mechanism is needed. An accurate malicious URL detection mechanism overcomes the security flaws and prevents the system from cybercriminal's activities. Thus, the objective of this work is to develop an automated detection mechanism that will detect the website/URL is malicious or not. We proposed a malicious URL detection mechanism based on the Multilayer CNN. The multilayer CNN based approach ensure effectiveness and reliability in the detection process. The major research contributions are as follows:

- The work focuses on the security problem that arises due to malicious websites.
- We proposed a malicious website/URL detection mechanism using multilayer CNN.

- The proposed detection mechanism first experimented on one layer of CNN. After that, two layers of CNN is used to improve the detection accuracy.
- The performance and detection accuracy of the model is evaluated and found promising results.

The remaining portion of this work is organized as follows: Section II discussed the previous research works related to malicious website detection. Section III elaborates the proposed multilayer CNN based malicious URL detection. The performance and results of the proposed model are discussed in Section IV. The paper is closed with the conclusion and future of the work in Section V.

II. Related works

This section discussed previous existing literature works that mainly focus on malicious website detection mechanism [14]–[18]. These works are helpful for the identification of research gaps and provide the base for developing the improved detection model. A feature engineering based malicious URL detection mechanism has been proposed in [19]. The methods use linear and non-linear space transformation approach. In this approach, linear transformation uses a two-stage distance metric learning mechanism, and nonlinear transformation uses the Nyström method for kernel approximation. It also includes multiple classifiers such as k-Nearest Neighbor, Support Vector Machine (SVM), and Multi-Layer Perceptron to improve the detection performance of the model. The result illustrated that the detection accuracy is achieved up to 86%. In [20] different webspam techniques has been discussed. The classification method uses the CNN algorithm. It also takes user perspective and screenshots of malicious web pages for the development of the detection model. The experiment was performed on a complex dataset as well as performed on a real-world web application.

Another study [21] presented a spam website detection method based on CNN, webpage screenshots, and end-user perspective. The semantic and visual information of the fetched URL has been used for analyzing the malicious URL [22]. The proposed model included a parallel neural joint algorithm that combined capsule network and independent recurrent neural network to utilize multi-modal features. At last, an attention mechanism is also added to improve the classification accuracy. The website features such as lexical, content, and host are used for analyzing the website genuineness [23]. These features are fed as an input in supervised machine learning classifiers such as random forest, decision tree, and DNN. They also proposed a concept of drift detection in which input data and target variables are changed over time so that the attacker cannot establish the relationship between both variables. PhishTank [24] dataset has been used for the implementation of the work.

A URL embedding (UE) model has been proposed in [10] for the detection of malicious URL in the cyber world. The unsupervised machine learning algorithm is used to obtain low dimensional features. The UE model explores the correlation and coefficients among different URLs, which is helpful for the representation of URL in a distributed fashion. A phishing website detection mechanism has been proposed in [25]. The model will detect phishing websites in two-stage. The first stage uses the Enhanced Probing Classification algorithm for malicious URL detection and the second stage uses Naive Bayes classifier for detection of URL set.

Artificial neural network learning techniques are also used for the detection of malicious web pages [26]. A framework has been introduced that analyzed discriminative features of the attacks to help reduce the false positive rate. The URL lexical and the page content features are included in the algorithm for improving the detection accuracy. In [27] unsupervised neural network learning algorithms are used for the detection of malicious websites. The competitive learning process is also used to enter training data into the network, and the Euclidean distance of all weight vectors is computed. Winner takes-all rule produced different results in the form of cluster.

Phishing attack identification techniques can detect malicious websites. Thus, in [28], the authors proposed a model that will catch the phishing attack so that the Internet users can be protected from different internet vulnerabilities. The web content and conceptual URL consistency are used for the development of such a detection mechanism. The static content analysis can help to determine the website is malicious or not. But, it cannot give much accuracy, so the authors [29] moved on client-side honeypot system in which runtime browsing information is deeply analyzed. Cross-layer detection mechanism [30] automatically detect the website is malicious or not. This approach contains both network layer traffics and application layer information and parallelly analyzed to improve system performance.

The following research limitations are identified from the above literature works:

1) Most of the malicious URL detection model uses machine learning approach, which required strong feature engineering for the accurate prediction.
2) Traditional malicious URL detection models rely on the manual rules and setting of these rules, and their thresholds are completely depend on the human experience. This can make the system biased, which will lead to creditability problems.
3) Traditional RNN malicious URL detection models considered only forward-to-backwards construction. It does not have reverse direction functions. Also, it requires more data for training [22].
4) The manual feature extraction in traditional ML algorithms require a huge amount of time for feature extraction and selection.

III. Proposed multilayer CNN

This section discusses the mechanism of the proposed malicious URL detection model. The ML-based approaches need pre-extracted features set to perform the detection task [31], [32]. The performance of the model entirely depends on the received set of features. The proposed model uses a

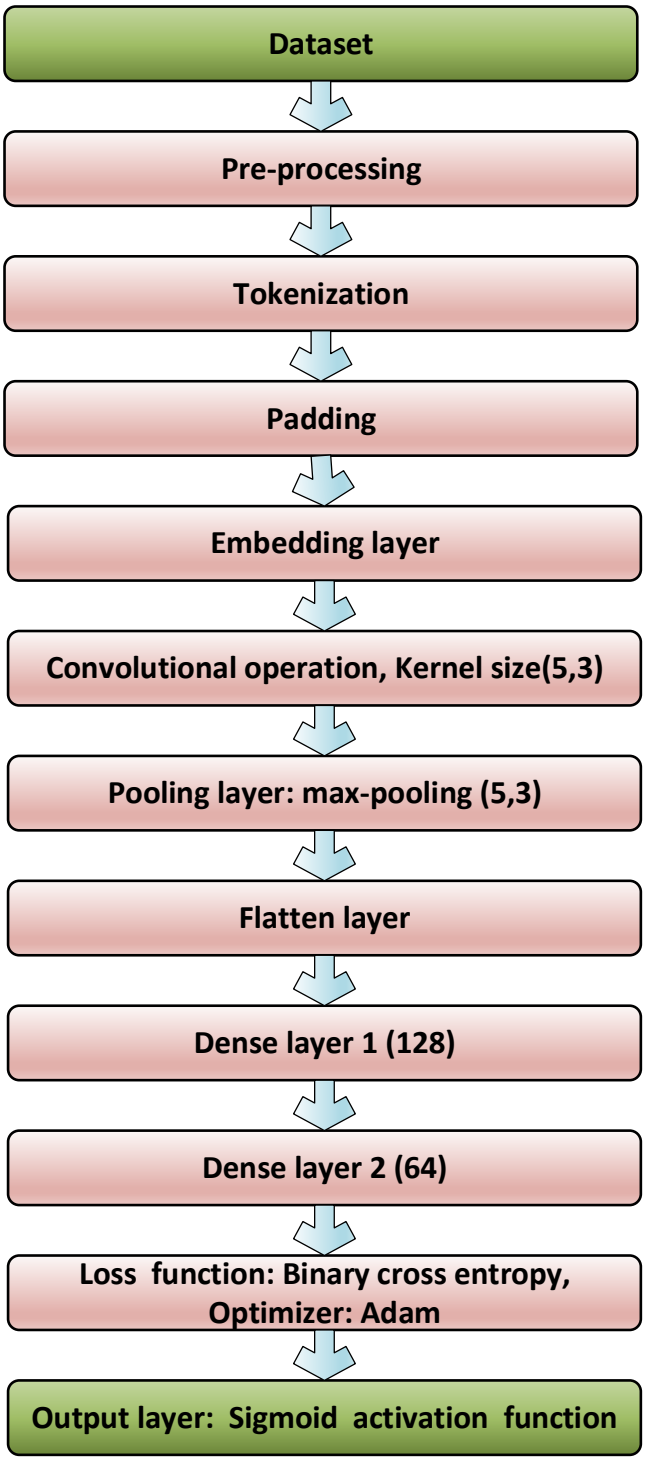

Fig. 1. Flow diagram of the proposed model

multilayer CNN model is used to overcome these issues and extract the hidden features automatically. The workflow of the proposed CNN model is shown in Figure 1. This research uses a dataset downloaded from the Kaggle repository consisting of 411247 URLs in which 18% URLs are malicious, and the remaining 82% URLs are non-malicious. The dataset is split into two parts: train and test with an 80:20 ratio. This means 80% of total URLs are used for model training, whereas with 20% of URLs, the trained model was tested.

The multilayer CNN can automatically extract the hidden patterns from the input URLs using several convolutions with different kernels' sizes. The multistage trainable neural network architecture consists of a self-extraction process that makes the model more efficient and independent from feature engineering. The proposed detection model involved the following primary operations.

1) Embedding Layer: An architecture of CNN is shown in Figure 2. In this figure, the first step is the Embedding layer which produced an embedding matrix. The embedding is the process of converting raw input to vector form. The Keras library supported an equal length of inputs. Hence, the input data are passed from padding operation in which data are converted into equal length. Further, using the embedding, converted the padded input to vector form. The outcome of word embedding's $(E(w_1), E(w_2), E(w_3), \cdots, E(w_n))$ are concatenated and formed a matrix 1 which contains 2D vectors (m*n), including the number of words (m) and the dimension of embedding (n). Following this approach, for each input URL, a matrix is formed.

$$V = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & e_{m2} & \cdots & w_{mn} \end{bmatrix} \quad (1)$$

2) Convolution operations: After the generation of the embedded matrix for input URL, convolution operations are performed. In this process, different sizes of the kernel are used. In the proposed model, the kernel size is 5 and 3. At a time, kernel size *k* is processed with the number of words and slide (stride) is shifted one position (by default). After that, the same operation will perform until the slider reaches the end of the word. The convolution process contains multiplication and addition operations. The Convolution Operation (CO) is represented by using Equation 2. In this matrix, f denotes the kernel, and * indicates the convolutional operator.

$$CO = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix} * \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \\ \vdots & \vdots \\ f_{n1} & f_{n2} \end{bmatrix} \quad (2)$$

The size of the feature map is computed as (*n*-*k*)+1. For instance, if the kernel size *k*=5 and *n*=56, then, the feature map size is (56-5)+1=52.

Rectified linear unit (ReLU) is the most implemented intermediate layer activation function (*f(r)*) [33]. This is a linear

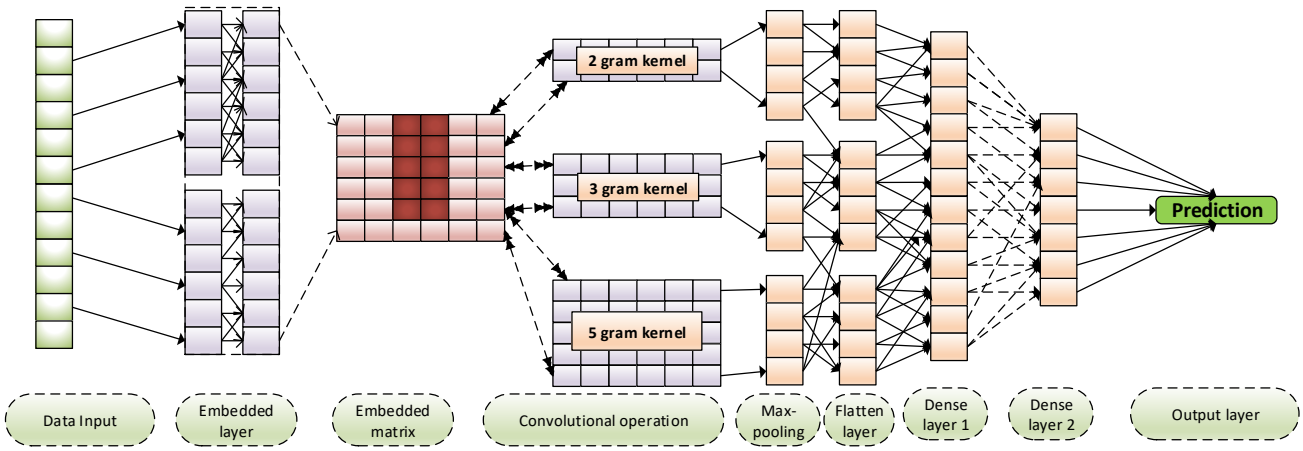

Fig. 2. Convolutional Neural Network for Malicious Website Detection

function in which both; the function and its derivative are monotonic. The range of ReLU function varies between [0 to ∞). It has less computational complexity compared to other activation function. The function output will be zero if any negative value is fed into the ReLU activation function; otherwise, the input and output both are the same in positive form. Mathematically, it can be represented by using Equation 3.

$$f(r) = \max(0, r) \tag{3}$$

3) Pooling: The convolutional process generates several features. But, some of the features are not contained any valuable information. Thus, a pooling operation is needed to extract the essential features from the set of features. Min, max [34], and average pooling is some of the pooling technique supported by Keras. This research considered the max-pooling approach. The pooling technique works as follows: we have considered feature map is *fm* and window size is *W*, the total number of the feature is computed for the next round is represented using Equation 4.

$$(F)_{next} = \frac{fm}{W} \tag{4}$$

Let consider fm=800 and w=5; then, according to Equation 4, the pooling technique fetched 800/5=160 features for the next round.

4) Flatten layer: The flatten process is started once the pooled feature is mapped. It is used to flatten the input in which the entire pooled feature map matrix is converted into a single column. Now, the output is fed into the neural network for further processing. For instance, if the input shape is (batch_size, 5,5), then the output shape of the layer will be (batch_size, 25).

5) Dense layer: At the end of the process, the fully connected dense layer consists of a feed-forward neural network. This implies that each neuron in the dense layer receives input from all neurons of its previous layer. It took the flattened data as an input and produced the results by performing matrix-vector multiplication. The matrix contains the actual value that can be trained and updated by using backpropagation. The final output of the operation is the 'm' dimensional vector. It means this is used for converting the dimensions of the vector.

6) Output layer: The output layer produced the actual predicted results in terms of the website is malicious or not. The output layer used several activation functions such as Sigmoid, Softmax, etc. The Sigmoid activation function has been used in our proposed model, which gives more accurate results than other activation functions. The value of the sigmoid function varies between 0 to 1. Mathematically, it can be a representation by using Equation 5. In this equation, *x* is the input value which is varies between -2 to +2. This equation illustrates that the slight change in *x* will produce a large variation in Sf(x).

$$Sf(x) = \frac{1}{1 + \exp^{-x}} \tag{5}$$

At the end of this operation, we can get the actual prediction result in terms of the website is malicious or not. We have followed all the operations as mentioned earlier for the development of our proposed detection model.

IV. Performance and Result Analysis

This section discusses the experimental outcomes of the proposed deep convolutional neural network. To evaluate the model, the precision, recall, F1-score metrics are used. The computation mechanism of these metrics is given in [22]. The number of malicious URLs predicted truly among the retrieved malicious URL termed precision, whereas The number of malicious URLs predicted truly among the total malicious URLs termed recall. The F1-score is the harmonic mean of precision and recall [35]. It is very much needed to receive the high precision value for the current problem, which indicates very few non-malicious websites predicted as malicious.

TABLE I
LIST OF PARAMETERS AND THEIR VALUES

Parameter	Value
Convolution Layer	Two
Kernel Size	5-, 3-gram
No. of Kernels	128,64
Pooling Type	Max Pooling
Pooling Window Size	5,3
Loss function	Binary cross entropy
Optimizer	Adam
Learning rate	0.01
No. of Neurons at dense layer	128,64
No. of Neurons at output layer	1
Output layer's activation function	Sigmoid
Internal layer's activation function	ELU and ReLU

TABLE II
MALICIOUS URL PREDICTION WITH ONE LAYER OF CNN

Url's Category	Pre	Rec	F_1
Non-Malicious	0.94	0.93	0.94
Malicious	0.90	0.88	0.89
Macro avg	0.92	0.92	0.92
Weighted avg	0.92	0.92	0.92

Firstly, the proposed CNN based deep learning model experimented with default parameters values, and later, the values are tuned. The tuned parameter's value with the model achieved better performance compared to the default setting are listed in Table I. Two convolution layers are used for the task, having 128 and 64 kernels of 5-gram and 3-gram, respectively. To pull the features, the max-pooling technique is used, having window size of 5 and 3. Binary cross entropy used as a loss function and Adam optimizer is used to optimize the network's weights. The dense layer consists of 128 and 64 neurons with ReLU and ELU activation functions. The model's output layer consists of the Sigmoid activation function, which yielded a value in the range of 0 to 1. The model is set to run for the 50 epochs with a batch size of 100. Finally, the trained model's weight is saved into a *.h5* file for further process.

Initially, one layer of the CNN model is executed. The outcomes of the model are shown in Table II. The precision (Pre), recall (Rec) and F1-score (F_1) form Non-Malicious class is 0.94, 0.93 and 0.94, respectively, whereas, for Malicious class it is 0.90, 0.88, and 0.89 respectively. The Weighted average Pre, Rec, and F_1 are 0.92, 0.92, and 0.92. By observing these outcomes, we conclude that the one-layer of CNN model prediction Non-Malicious URL has a satisfactory rate, whereas many Malicious URLs are misclassified.

To reduce the misclassification, we re-experimented the model by increasing the layer of CNN by one in the previous model. The outcomes of the 2-layer CNN model shown in Table III. The Pre, Rec, and F_1 value of Non-Malicious class is 0.97, 0.98 and 0.97, respectively, whereas for Malicious class it is 0.90, 0.91, and 0.91, respectively. The weighted Pre, Rec, and F_1 is 0.94, 0.93, and 0.93. Even though

TABLE III
MALICIOUS URL PREDICTION WITH TWO LAYERS OF CNN

Url's Category	Pre	Rec	F_1
Non-Malicious	0.97	0.98	0.97
Malicious	0.90	0.91	0.91
Macro avg	0.93	0.93	0.93
Weighted avg	0.94	0.93	0.93

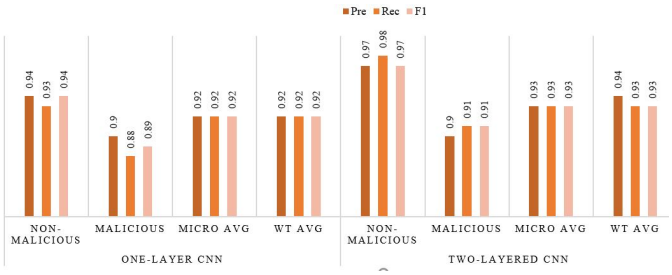

Fig. 3. Comparative performance of one-layer and two-layered CNN model

the performance of the two-layered CNN model is improved compared to the one-layer of CNN model, the misclassification rate for the Malicious class is high as compared to the Non-Malicious class. This indicates that, 2-layers of the CNN model also unable to predict all malicious URLs. However, the recall value of 0.91 indicated that the model successfully detects 91% of Malicious URLs. Comparative performance of one-layer CNN and two-layered CNN in terms of Pre, Rec, and F_1 can also be seen from Figure 3. This model can be tuned by changing the values of the hyperparameters or by analyzing Non-Malicious and Malicious categories URLs to improve the prediction accuracy. The number of CNN layers can also be tuned for better performance. This research does not utilize any pre-trained embedding model. However, using the pre-trained embeddings may also help to decrease the misclassification rate.

V. CONCLUSION AND FUTURE WORK

We developed an automated model to detect malicious URLs. A deep neural-based Convolutional Neural Network is used for the said task. Initially, a one-layer of CNN model is used. Later, a two-layered CNN model is used for the same. The proposed Multi-layered CNN model yielded the Pre, Rec and F_1 value for Malicious URL prediction as 0.90, 0.91, and 0.91, respectively, for the best case. This indicates that 91% of Malicious URLs are successfully detected; however, 9% of Malicious URLs are not detected by the system. In the future, the CNN model parameters such as number of layers, size of kernels, number of kernels, and optimizer can be tuned to get a better model for the said task. Also, the performance comparison with the traditional machine learning-based classification models- such as Naive Bayes, Random Forest, Support Vector Machine, Logistic Regression and similar ones can be made to check the relevance of automated feature extraction-based deep learning model.

REFERENCES

[1] Microsoft. MICROSOFT SECURITY INTELLIGENCE REPORT. *https://clouddamcdnprodep.azureedge.net/gdc/gdchZI1oF/original [accessed 2021-06-13]*, 24, JANUARY - DECEMBER 2018.

[2] Surbhi Gupta, Abhishek Singhal, and Akanksha Kapoor. A literature survey on social engineering attacks: Phishing attack. In *2016 international conference on computing, communication and automation (ICCCA)*, pages 537–540. IEEE, 2016.

[3] Huajun Huang, Junshan Tan, and Lingxi Liu. Countermeasure techniques for deceptive phishing attack. In *2009 International Conference on New Trends in Information and Service Science*, pages 636–641. IEEE, 2009.

[4] Juan Chen and Chuanxiong Guo. Online detection and prevention of phishing attacks. In *2006 First International Conference on Communications and Networking in China*, pages 1–7. IEEE, 2006.

[5] Bryan Parno, Cynthia Kuo, and Adrian Perrig. Phoolproof phishing prevention. In *International conference on financial cryptography and data security*, pages 1–19. Springer, 2006.

[6] Abdulghani Ali Ahmed and Nurul Amirah Abdullah. Real time detection of phishing websites. In *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 1–6. IEEE, 2016.

[7] Anjum N Shaikh, Antesar M Shabut, and M Alamgir Hossain. A literature review on phishing crime, prevention review and investigation of gaps. In *2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*, pages 9–15. IEEE, 2016.

[8] Ankit Kumar Jain and BB Gupta. A survey of phishing attack techniques, defence mechanisms and open research challenges. *Enterprise Information Systems*, pages 1–39, 2021.

[9] Siti Hawa Apandi, Jamaludin Sallim, and Roslina Mohd Sidek. Types of anti-phishing solutions for phishing attack. In *IOP Conference Series: Materials Science and Engineering*, volume 769, page 012072. IOP Publishing, 2020.

[10] Xiaodan Yan, Yang Xu, Baojiang Cui, Shuhan Zhang, Taibiao Guo, and Chaoliang Li. Learning url embedding for malicious website detection. *IEEE Transactions on Industrial Informatics*, 16(10):6673–6681, 2020.

[11] Yoshiro Fukushima, Yoshiaki Hori, and Kouichi Sakurai. Proactive blacklisting for malicious web sites by reputation evaluation based on domain and ip address registration. In *2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 352–361. IEEE, 2011.

[12] Andreas Dewald, Thorsten Holz, and Felix C Freiling. Adsandbox: Sandboxing javascript to fight malicious websites. In *proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1859–1864, 2010.

[13] Jian Zhang, Phillip A Porras, and Johannes Ullrich. Highly predictive blacklisting. In *USENIX Security Symposium*, pages 107–122, 2008.

[14] Suleiman Y Yerima and Mohammed K Alzaylaee. High accuracy phishing detection based on convolutional neural networks. In *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–6. IEEE, 2020.

[15] Rima Masri and Monther Aldwairi. Automated malicious advertisement detection using virustotal, urlvoid, and trendmicro. In *2017 8th International Conference on Information and Communication Systems (ICICS)*, pages 336–341. IEEE, 2017.

[16] Chun-ming WU, LI Min, YE Li, Xian-chun ZOU, and Bao-hua QIANG. Malicious website detection based on urls static features. *DEStech Transactions on Computer Science and Engineering*, (mso):1–7, 2018.

[17] Birhanu Eshete, Adolfo Villafiorita, and Komminist Weldemariam. Binspect: Holistic analysis and detection of malicious web pages. In *International conference on security and privacy in communication systems*, pages 149–166. Springer, 2012.

[18] Toshiki Shibahara, Kohei Yamanishi, Yuta Takata, Daiki Chiba, Mitsuaki Akiyama, Takeshi Yagi, Yuichi Ohsita, and Masayuki Murata. Malicious url sequence detection using event denoising convolutional neural network. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2017.

[19] Tie Li, Gang Kou, and Yi Peng. Improving malicious urls detection via feature engineering: Linear and nonlinear space transformation methods. *Information Systems*, 91:101494, 2020.

[20] Dongjie Liu and Jong-Hyouk Lee. Cnn based malicious website detection by invalidating multiple web spams. *IEEE Access*, 8:97258–97266, 2020.

[21] Dongjie Liu, Jong-Hyouk Lee, Wei Wang, and Yang Wang. Malicious websites detection via cnn based screenshot recognition. In *2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA)*, pages 115–119. IEEE, 2019.

[22] Jianting Yuan, Guanxin Chen, Shengwei Tian, and Xinjun Pei. Malicious url detection based on a parallel neural joint model. *IEEE Access*, 9:9464–9472, 2021.

[23] Siddharth Singhal, Utkarsh Chawla, and Rajeev Shorey. Machine learning & concept drift based approach for malicious website detection. In *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 582–585. IEEE, 2020.

[24] OpenDNS. Phishtank. *https://www.phishtank.com/ [accessed 2021-06-13]*.

[25] N Jayakanthan, AV Ramani, and M Ravichandran. Two phase classification model to detect malicious urls. *International Journal of Applied Engineering Research*, 12(9):1893–1898, 2017.

[26] Abubakr Sirageldin, Baharum B Baharudin, and Low Tang Jung. Malicious web page detection: A machine learning approach. In *Advances in Computer Science and its Applications*, pages 217–224. Springer, 2014.

[27] Dusan Stevanovic, Natalija Vlajic, and Aijun An. Detection of malicious and non-malicious website visitors using unsupervised neural network learning. *Applied Soft Computing*, 13(1):698–708, 2013.

[28] Nureni Ayofe Azeez, Balikis Bolanle Salaudeen, Sanjay Misra, Robertas Damaševičius, and Rytis Maskeliūnas. Identifying phishing attacks in communication networks using url consistency features. *International Journal of Electronic Security and Digital Forensics*, 12(2):200–213, 2020.

[29] Tung-Ming Koo, Hung-Chang Chang, Ya-Ting Hsu, and Huey-Yeh Lin. Malicious website detection based on honeypot systems. In *2nd International Conference on Advances in Computer Science and Engineering (CSE 2013)*, pages 76–82. Atlantis Press, 2013.

[30] Li Xu, Zhenxin Zhan, Shouhuai Xu, and Keying Ye. Cross-layer detection of malicious websites. In *Proceedings of the third ACM conference on Data and application security and privacy*, pages 141–152, 2013.

[31] Pradeep Kumar Roy, Jyoti Prakash Singh, and Snehasish Banerjee. Deep learning to filter sms spam. *Future Generation Computer Systems*, 102:524–533, 2020.

[32] Pradeep Kumar Roy, Asis Kumar Tripathy, Tapan Kumar Das, and Xiao-Zhi Gao. A framework for hate speech detection using deep convolutional neural network. *IEEE Access*, 8:204951–204962, 2020.

[33] Konstantin Eckle and Johannes Schmidt-Hieber. A comparison of deep networks with relu activation function and linear spline-type methods. *Neural Networks*, 110:232–242, 2019.

[34] Naila Murray and Florent Perronnin. Generalized max pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2473–2480, 2014.

[35] Pradeep Kumar Roy. Multilayer convolutional neural network to filter low quality content from quora. *Neural Processing Letters*, 52(1):805–821, 2020.