ISSN 2723-6471

HTTP Traffic Analysis based on Multiple Deep Convolution Network Model Generation Algorithms

Bocheng Liu; Fan Yang *

School of Software, Nanchang University, Nanchang 330047, China
bcliu@ncu.edu.cn
* corresponding author

(Received: September 25, 2022 Revised: October 20, 2022 Accepted: November 12, 2022, Available online: December 23, 2022)

Abstract

In recent years, with the development of the Internet, social networking, online banking, e-commerce and other network applications are growing rapidly. At the same time, all kinds of malicious web pages are constantly emerging. Under the new situation, the network security threats are distributed, large-scale and complex. New network attack modes are emerging. With more and more diverse devices accessing the Internet, our life is more intelligent and convenient, but also brings more loopholes and hidden dangers. Some malicious web pages through a variety of means to lure users to open URL links and conduct malicious behavior. However, if we can detect the URL of the malicious web page and identify the malicious web page, we can avoid the problems of content variability and behavior tracking. Therefore, traffic analysis based on various deep convolutional network model generation algorithms arises at the historic moment, and becomes an important research issue in the field of Internet security.

Keywords: URL; Traffic Analysis; Deep Learning

1. Introduction

In recent years, there are countless malicious acts of using malicious web pages to commit crimes on the Internet. It is reported that nearly half of the web pages are potentially malicious, Malicious web pages launch malicious behaviors to users by sending a large number of emails containing malicious URLs, phishing and other means, resulting in the lack of security awareness of users suffering from varying degrees of harm. Therefore, how to effectively and timely detect malicious web pages has become an important problem to be solved. One common malicious act committed through malicious web pages is phishing attacks. These attacks involve creating a fake website that appears to be legitimate, such as a bank or online shopping site, and tricking users into entering their personal information, such as login credentials or credit card numbers. The attackers then use this information to steal money or identity.

Another malicious act is malware distribution. Malicious web pages can be used to download and install malware onto a user's device without their knowledge. This malware can range from simple viruses that disrupt a device's functionality, to more advanced malware such as ransomware, which encrypts a user's files and demands payment to unlock them. Malicious web pages can also be used to distribute spam emails or spam comments on social media. These spam messages often contain links to more malicious web pages or attempt to sell fake products or services.

Another malicious act is clickjacking, which involves hiding a button or link on a web page behind a legitimate element, such as an advertisement. When a user clicks on the legitimate element, they are actually clicking on the hidden button or link, which can lead them to a malicious web page or perform an unwanted action, such as liking a page on social media or signing up for a subscription. Malicious web pages can also be used for cyberstalking or harassment. These web pages can contain personal information about an individual, such as their location or contact information, and can be used to threaten or intimidate them.

Finally, malicious web pages can be used for distributing illegal content, such as child pornography or copyrighted material. These web pages often operate under the radar of law enforcement and can be difficult to track down and shut down. The main methods of malicious web page identification include detecting the URL of the webpage and detecting the content and behavior of the webpage. This paper focuses on the detection of malicious web URL technology, The next part will summarize the general technical steps of traffic analysis: through the use of deep learning model generation algorithm to detect HTTP traffic, use the text features and image features crawled by URL, construct a multi-layer learning network for URL structural features, and complete the detection and classification of malicious URL.

2. Algorithm Implementation

The HTTP traffic analysis technology model is shown in Figure 1. The input of the model is URL, and the output is the detection result of the URL.

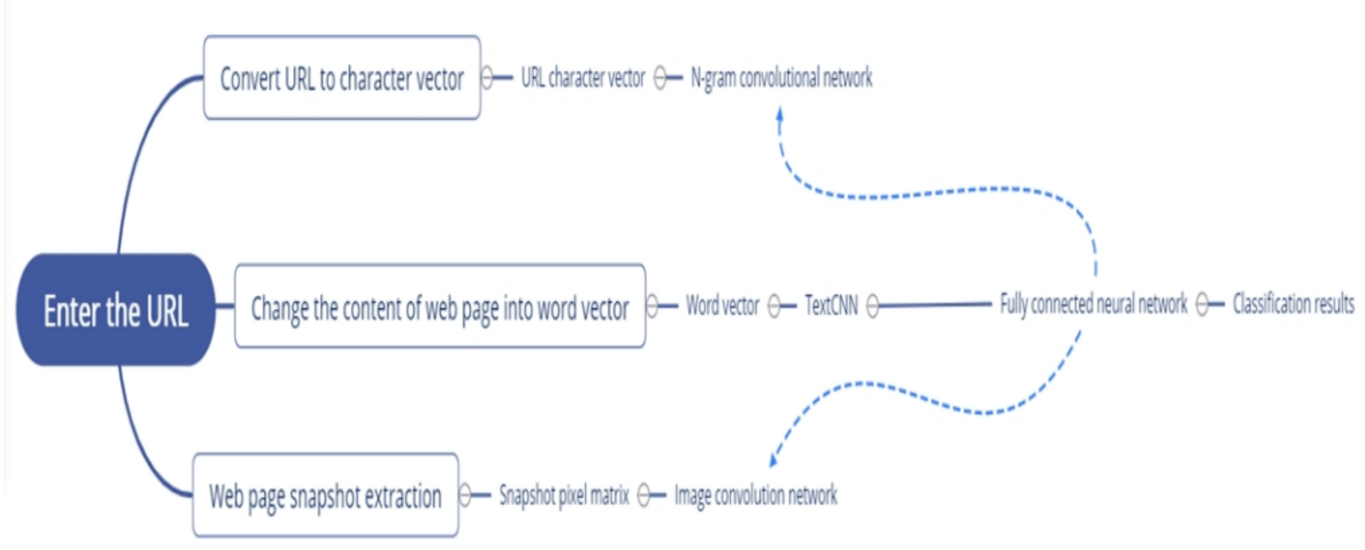
Convert URL to character vector
URL character vector
N-gram convolutional network
Enter the URL
Change the content of web page into word vector
Word vector
TextCNN
Fully connected neural network
Classification results
Web page snapshot extraction
Snapshot pixel matrix
Image convolution network

Figure. 1. Flow Analysis Technology Model

Input: the input model is a complete URL. However, considering the accuracy factor, we will extract URL features from three aspects: URL structured features (that is, the features reflected by URL string), web page text features crawled by URL, and web page snapshot features crawled by URL. In order to provide convenience for subsequent deep learning, in the preprocessing stage, we will abstract these three features into digital vectors. Output: we divide URLs into seven categories, normal URLs into one category and malicious URLs into six categories. Finally, the system will get a classification report for the input URL, and the specific classification is shown in Table 1.

Table. 1. Malicious URL Classification

Identifier	Meaning
Positive	Normal URL
Botnet	Botnet
Phishing	Phishing Sites
Pony	Pirated Websites
Suppobox	Trojan Horse Program
Suspicious	With Suspicious Information

ISSN 2723-6471

Virus	It Contains Virus

2.1. URL Structured Feature Extraction

First, convert the URL string into a numeric matrix. Figure 2 shows the conversion process from string to vector: a character corresponds to a multidimensional vector, so a URL string is converted into a number matrix.

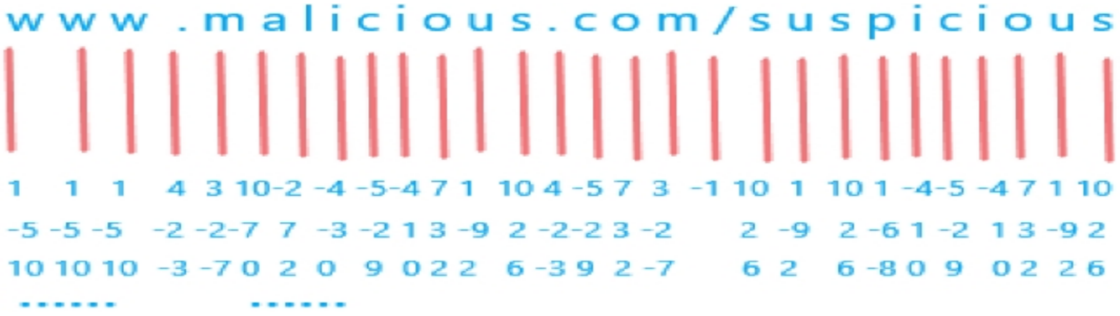

Figure. 2. Transformation process

In our system, symbols are divided into similar characters, lower case letters are divided into similar characters, and upper case letters are also divided into similar characters. After visualizing the character vector, we can find that the distance between similar characters is close. We use three, four and five convolution windows to convolute the character vector. Firstly, a convolutional network automatically induces pattern features from a large number of labeled URL character matrix inputs. Then every time there is a URL input, the neural network will match the input URL by convolution [1]. If the neural network finds a capital letter followed by a control character or a number, it will automatically compare with the pattern feature set to see whether it conforms to an existing pattern. The convolution process is shown in Figure 3.

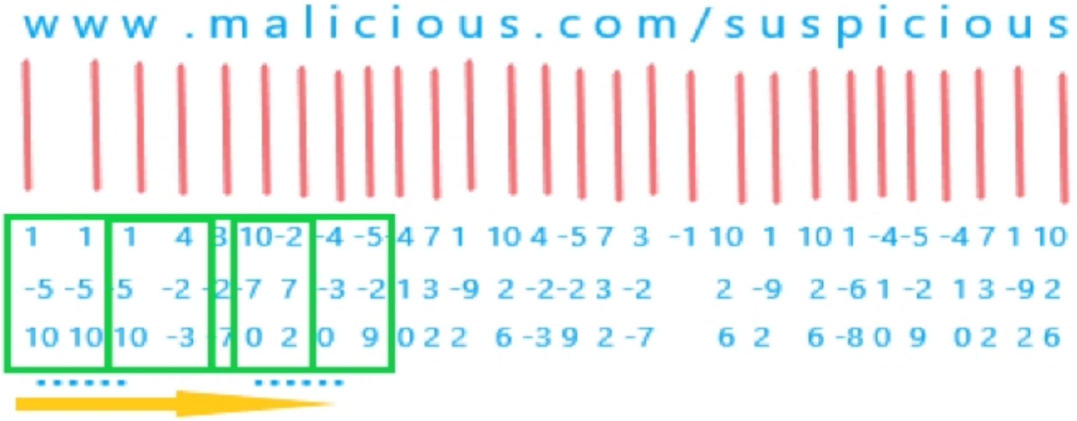

Figure. 3. Convolution process

2.2. Web Text Feature Extraction

By accessing the URL, you can get the content of the web page. It can then be converted to a word vector by word2vec [2]. Word2vec is a simple and efficient open source tool provided by Google. Its function is to convert text into word vectors, that is, to map words into continuous vector space. The workflow of word2vec can be simply described as receiving a segmented text, calculating a multidimensional word vector for each word according to the similarity and relevance of words in the text, and finally outputting the word vector. The output word vector contains all kinds of digital information of words, and these word vectors form a digital matrix, which can be processed by textCNN [3].

The purpose of traditional natural language processing is to find the language model (the whole Huffman tree), while the purpose of word2vec is to get the vector information of each word recorded on the leaf node. After the crawled web pages are transformed into word vectors, text convolution neural network (textCNN) is used for training.

A text convolution neural network consists of several convolution layers, pooling layer and full connection layer [4]. Each layer of convolutional neural network is arranged in three dimensions, including height, width and number of windows. The number of windows refers to how many different feature extraction windows we use, which can be set freely. Three different types of windows can extract three feature maps, and five different types of windows can extract five feature maps. As can be seen from Figure 4, after the convolution layer, the pooling layer does the maximum sampling on the extracted feature map to get a smaller feature map. The convolution and pooling can be repeated several times to obtain the final feature map. The last two fully connected layers give the final output.

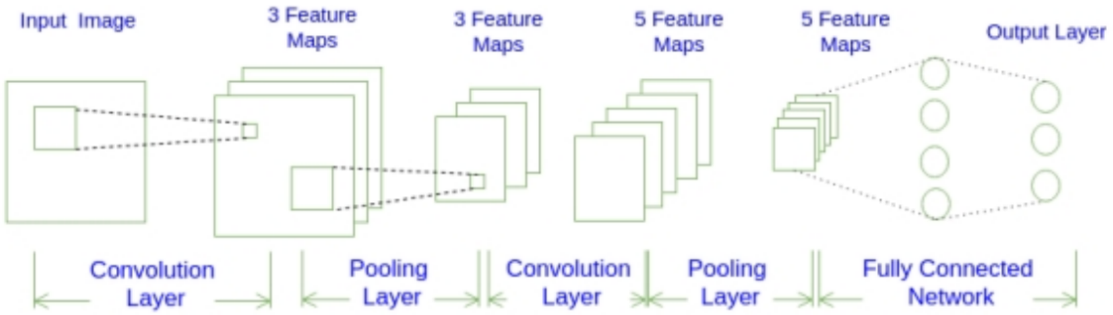

Figure. 4. Text convolution network diagram

In the actual operation, we select three kinds of windows with width of 3, 4 and 5, and the number of each window is set to 128, which can extract more comprehensive features and help to improve the accuracy of the final results. Figure 5 shows the model architecture of texCNN [5]. The final output of textCNN is a probability matrix, which indicates the probability that the new input text belongs to classification. In the ideal state, we want the final output to conform to the one hot characteristic, that is, in this one-dimensional matrix, only one value is 1, and the other values are 0 (for example, [0,0,0,0,1,0]). The closer the output matrix is to one hot, the higher the accuracy of the model is [6].

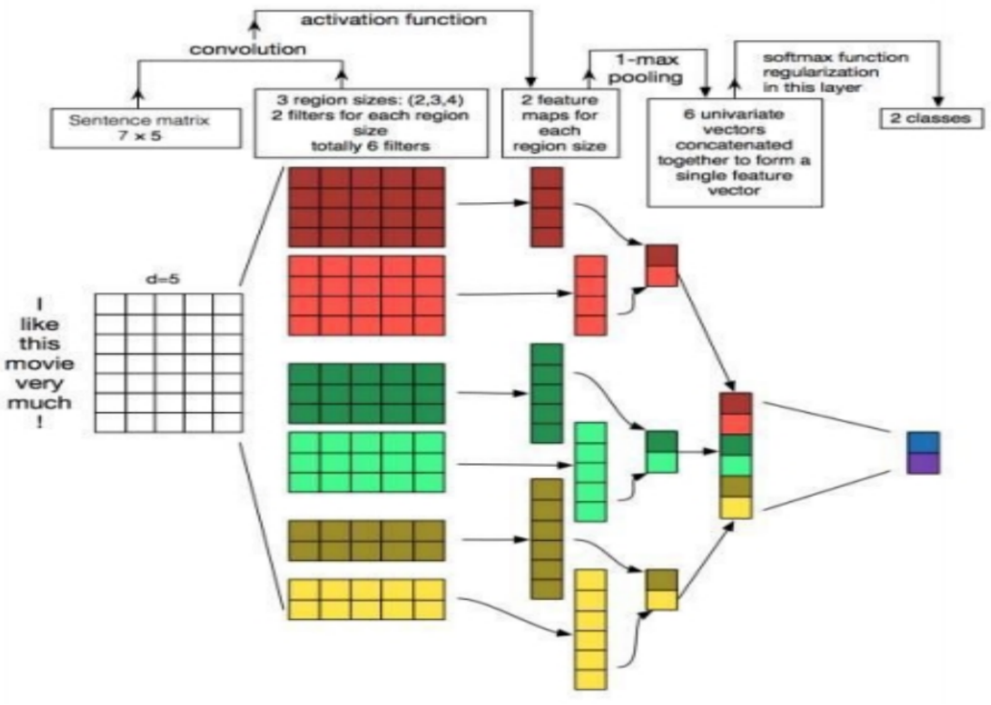
activation function
convolution
Sentence matrix 7 × 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
1-max pooling
6 univariate vectors concatenated together to form a single feature vector
softmax function regularization in this layer
2 classes
d=5
I
like
this
movie
very
much
!

Figure. 5. Model architecture of textcnn

2.3. Image Feature Extraction

The image is characterized by the snapshot of the web page corresponding to the malicious URL, and the browser module built in Python is used. Phantomjs processes the screenshot of web pages, and uses the function of PIL module to cut the obtained web screenshot to the same size (1200 * 900 pixels), which is suitable for the input requirements of the same size picture in the insectionv3 image processing [7]. Concept-v3 is a model of image classification based on CNN introduced by GoogleNet in, which is used to train the large visual recognition challenge data set of ImageNet in 2012. In this project, we use the concept V3 model with the lowest error rate and the framework of TensorFlow to build our own image recognition model. Compared with the same model, concept V3 uses GoogleNet. From the number of parameters, GoogleNet parameters are 5million, AlexNet parameters are 12 times of GoogleNet, and VGGNet parameters are three times of AlexNet. Therefore, GoogleNet is a good choice when memory or computing resources are limited [8]. The accuracy gain of GoogleNet mainly comes from the dimension reduction, and the concept module is fully convoluted. Each weight corresponds to a multiplication operation. After convolution decomposition, the number of parameters can be reduced for fast training, so that the size of filter banks can be increased to improve the accuracy.

2.4. Fully Connected Neural Network

The fully connected neural network of this system is a simple BP (back propagation) neural network. BP neural network has been very mature, and its research is also very comprehensive [9]. For each neuron, the accumulated stimulus is the sum of the amount of stimulus transmitted by other neurons and the corresponding weight. It is used to express thVe accumulation, the amount of stimulus transmitted by a neuron, and the weight of a neuron.

$$X_j = \sum Y_i \cdot W_i$$

When the accumulation is completed, it spreads stimulation to some neurons around it:

$$Yi = f(Xj)$$

The function here represents the activation function [10].

BP neural network is composed of many neurons. In short, the network can be divided into three layers. The input layer transmits the stimulus to the hidden layer, and the hidden layer transmits the stimulus to the output layer through the strength (weight) and transfer rule (activation function) of the connection between neurons. The output layer sorts out the stimulus processed by the hidden layer to produce the final result. If there is a correct result, then the correct result is compared with the generated result to get the error, and then the link weight in the neural network is modified by backstepping, so as to complete the learning process [11, 12].

In the training process of BP neural network, the system takes the three probability matrix sets obtained in the previous learning process as the abstract features of the three dimensions of URL, takes them as inputs, and uses the fully connected neural network for centralized learning. Finally, we can get a high-precision deep learning model to detect malicious URLs to distinguish malicious URLs.

3. Conclusion

In recent years, with the development of various deep convolutional network model generation algorithms, traffic analysis and monitoring have become more and more important in a variety of tasks. The application of deep learning has also achieved remarkable results, which provides ideas for network security protection. Using the HTTP traffic, the deep neural network is used to analyze, so as to get the malicious URL, so that the machine has a certain network protection ability. In the follow-up work, on the one hand, we will continue to improve the number and diversity of data, improve the performance of existing detection algorithms; on the other hand, we plan to develop a malicious URL real-time monitoring function on the basis of the above technology, to provide more technical support for network security protection.

References

[1] Nidhi Srivastav and Rama Krishna Challa. Novel intrusion detection system integrating layered framework with neural network. In Advance Computing Conference (IACC), 2013 IEEE 3rd International, pages 682–689. IEEE, 2013.

[2] Reyadh Shaker Naoum, Namh Abdula Abid, and Zainab Namh Al- Sultani. An enhanced resilient backpropagation artificial neural network for intrusion detection system. International Journal of Computer Science and Network Security (IJCSNS), 12(3):11, 2012.

[3] Mirsky Y, Doitshman T, Elovici Y, et al. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection// Network and Distributed System Security Symposium. 2018.

[4] Shibahara T, Yamanishi K, Takata Y, et al. Malicious URL sequence detection using event denoising convolutional neural network// ICC 2017 - 2017 IEEE International Conference on Communications. IEEE, 2017:1-7.

[5] M. Stevanovic, J.M. Pedersen. An efficient flow-based botnet detection using supervised machine learning. In: 2014 International Conference on Computing, Networking and Communications (ICNC), pp. 797-801.

[6] A. Nogueira, P. Salvador, F. Blessa. A botnet detection system based on neural networks. In: 2010 Fifth 4 International Conference on Digital Telecommunications, pp. 57-62.

[7] J. Mazel, P. Casas, Y. Labit and P. Owezarski. Sub-space clustering, inter-clustering results association and anomaly correlation for unsuper- vised network anomaly detection. In Proceedings of the 7th International Conference on Network and Service Management, Paris, France, 2011: 1-8.

[8] P. Casas, J. Mazel, and P. Owezarski. Unsupervised Network Intrusion Detection Systems: Detecting the Unknown without Knowledge. Com- puter Communications, April 2012, 35(7): 772-783.

[9] Viegas E, Santin A, Neves N, et al. A Resilient Stream Learning Intrusion Detection Mechanism for Real-Time Analysis of Network Traffic// GLOBECOM 2017 - 2017 IEEE Global Communications Conference. IEEE, 2018:1-6.

[10] Wan X, Sheng G, Li Y, et al. Reinforcement Learning Based Mobile Offloading for CloudBased Malware Detection// GLOBECOM 2017 - 2017 IEEE Global Communications Conference. IEEE, 2018:1-6.

[11] Wang J, Yang L, Wu J, et al. Clustering analysis for malicious network traffic// IEEE International Conference on Communications. IEEE, 2017:1-6.

[12] Xiao Fu, Ma Junqing, Huang xunsong, Wang Ruchuan. DDoS attack detection method based on KNN in SDN environment . Journal of Nanjing University of Posts and Telecommunications (SCIENCE EDITION), 2015,35 (01): 84-88