Malicious URL Detection Algorithm Based on Multi Neural Network Series

Weirong Xiu [1], Chen Bian [2*], Donghua Zheng [1], Lizhu Ye [1], Lina Dai [1]
[1]*School of information Technology and Engineering, Guangzhou College of Commerce, Guangzhou 510700, Guangdong, China*
[2]*College of Internet Finance and Information Engineering, Guangdong University of Finance, Guangzhou 510521, Guangdong, China*
**Corresponding author.*

Abstract

Convolutional neural network based on attention mechanism and a bidirectional independent recurrent neural network tandem joint algorithm (CATIR) are proposed. In natural language processing related technologies, word vector features are extracted based on URLs, and the extracted URL information features and host information features are merged. The proposed CATIR algorithm uses CNN (Convolutional Neural Network) to obtain the deep local features in the data, uses the Attention mechanism to adjust the weights, and uses IndRNN (Independent Recurrent Neural Network) to obtain the global features in the data. The experimental results shows that the CATIR algorithm has significantly improved the accuracy of malicious URL detection based on traditional algorithms to 96.9%.

Keywords: *Malicious URL, CATIR, word vector feature, detection*

I. Introduction

With the development of the Internet, the Internet is becoming more and more closely connected with users' lives, and more and more personal information of users exists on the Internet, so the security of information on the Internet is especially important. Nowadays, the proliferation of Internet links also provides space for malicious URLs to flourish. Malicious URLs contain malicious advertisements, phishing websites, and network viruses, which can illegally steal users' personal information and threaten users' property security to gain personal benefits. However, such cybercrime has difficulties in forensics and tracking, and it is difficult to eliminate the development of malicious URLs at the root. Therefore, it is imperative to strengthen the research related to malicious URL detection to isolate users from network dangers and create a safe Internet environment as much as possible. From the technical aspect, Most of the current malicious URL detection methods come from the traditional single algorithm model, and the performance of detection results is poor. Therefore, improving the way of malicious URL detection has become an inevitable trend to prevent cybercrime, and is an important issue to be solved in the field of network security research.

Nowadays, there are many domestic and international studies related to malicious URL detection, and researchers have proposed a variety of malicious URL detection methods. In terms of page and feature extraction techniques, ZHOU Qiang and others [1] proposed a malicious URL detection for target URL based on the acquired system snapshot file. The method consists of rolling back the virtual machine to the startup state in response to a trigger event of the virtual machine and loading the page content of the target URL for malicious detection. sung in Kim et al[2] analyzed the malicious URL protection of attackers' behavior habits, extracted common features, and put them in three different function pools to determine the harm degree of unknown URL, and used similarity matching techniques to improve the detection rate. The method covers a large number of malicious URLs with a small feature set, and only needs to check the properties of the URL. Quan et al [3] proposed a new vocabulary method, which uses machine learning technology to classify URLs. This method is based on natural language processing functions. These functions use word vector representation, and the N-gram model on blacklist words as the main scheme, which can help the classifier distinguish benign URLs and malicious URLs, and achieve 97.1% high

ISSN: 0010-8189

accuracy in SVM. Tao Sinan et al [4] provided a method and device for detecting malicious URL, including receiving URL detection request, analyzing the content of URL addressed page in URL detection request, and determining whether the page is a non text page; When the page is a non text page, the image of the page displayed in the browser and addressed by the URL detection request is obtained. The image of the page is detected, and the page attributes are obtained. According to the URL attribute of the URL detection request and the page attribute of the URL detection request, whether the URL is a malicious URL is finally determined.

Machine learning algorithms occupy an irreplaceable position in the extraction of deep features and feature semantic information, related research based on deep learning algorithms is increasing. Jianguo Jiang et al.[5] proposed a character-level deep neural network-based online detection scheme that uses a natural language processing approach to map URL and DNS strings into vector form, and a convolutional neural network framework designed to automatically extract malicious features and train a classification model.Selva Ganapathy et al.[6] used a stacked constrained Boltzmann machines for binary classification by feature selection in deep neural networks, using IBK-kNN, binary correlation and Powerset with SVM labels for multi-category classification, tested on a sample of 27,700 URLs with good results.

Zeyu Li et al.[7] proposed an ensemble model based on machine learning algorithm model, and found that the random forest algorithm had the best detection effect through comparative experiments. Wang Weiping et al[8] proposed a malicious URL detection method combining multiple features to extract the classification features of each sample with the URL to be detected from the access interaction data, and use these to train the classifier model. Meng Zhang et al. [9] constructed a multi classifier detection model, implemented a malicious URL detection system for real-time data stream processing, and used the CNN network model of three classifiers for malicious URL detection to complete related research. Rong Wei et al [10] designed and implemented a malicious Web request detection system based on the CNN model for the URL structure of Web requests, drawing on the feature extraction principle of CNN model for image recognition, and automatically extracted the salient features of various malicious requests through the convolutional kernel of CNN, and then detected the various malicious requests to be detected from the huge number of user requests. Ran Li et al [11] reasonably abstracted and split the relevant detection processes, implemented a high-performance detection system using distributed techniques, and made the system scalable and easy to maintain by implementing the corresponding cluster scheduling function, and fused rule-based filters, filters based on XgBoost model and filters based on deep learning algorithms to achieve a high-performance malicious URL discriminative model. Li Mengyu et al [12] establish a URL-based malicious access identification model, and design and implement a system for malicious access identification based on it. In which the URL logs of users accessing a domain name are used as the research object, the performance characteristics of malicious access are mined from multi-dimensions, and the sample markers with inaccurate categories near the critical points in the Gaussian mixed clustering classification results are modified with manual assistance, and then the S4VM algorithm is used to output a detection model that can identify malicious access, according to the proposed identification model, a URL-based malicious access identification system is designed and implemented. Most of the past malicious URL detections are based on blacklisting techniques [13], reputation systems [14], host features[15][16], honeypot techniques[17], lexical features[18] and intrusion detection techniques[19].
With the rise of deep learning, where highly accurate and robust deep learning models have achieved significant results in several research areas, and the application of deep learning algorithm in malicious URL detection provides a new research direction for researchers of detection methods. The latest researchers have started to try to make appropriate adjustments to deep learning algorithms for the specificity of malicious URL detection, and detection accuracy has been improved to some extent, but most of the methods are based on a single algorithmic model, however, each algorithm has its strengths and shortcomings. The convolutional neural network in deep learning is a classical traditional machine learning algorithm, a neural network consisting of a convolutional layer, a pooling layer and a fully connected layer, which has the unique advantage of coping with the construction of a model with a similar mesh structure. In the CNN algorithm, the convolutional layer and pooling layer should first form a convolutional group for feature learning, and use the fully connected layer for judgment and classification. However, the correlation between input and output in the CNN algorithm is poor, which has a certain impact on the malicious detection effect; Attention mechanism focuses on the relationship existing between focus, context, is

learned the thinking pattern of the human brain, can extract semantic features, for modeling long-distance performance is better, strengthen the correlation ability, can well make up for the correlation between input and output of CNN algorithm, Therefore, the CNN algorithm and the Attention mechanism are jointly processed in tandem, which not only draws the capability of feature learning of CNN algorithm but also incorporates the strong correlation between input and output of Attention mechanism, providing a strong guarantee for the detection capability of the algorithm in this paper. Meanwhile, since the IndRNN algorithm using unsaturated function as the activation function obtained good results after learning training, and the IndRNN algorithm also performed well for processing sequences with short step lengths, the IndRNN algorithm was combined with the tandem joint algorithm of Attention mechanism and CNN algorithm for another tandem processing to obtain CATIR tandem joint algorithm, and the model was trained by fusing information features of host and URL and word vector features for analysis and detection of malicious URLs.

This section gives a brief introduction, reviews the research status of malicious URL detection at home and abroad, and introduces the importance, research purpose and detection algorithm of malicious URL detection; Section 2, the modeling process, feature extraction and execution steps of multi network serial fusion malicious URL detection algorithm are introduced in detail; Section 3 presents the experimental data Statistics and analysis are presented in Section 3, from experimental data, optimal parameter settings, experimental results and analysis, and comparison with other malicious URL detection machine learning algorithms; finally, a summary description of the research work is given.

The main elements of the work in this paper include

(1) The URL-based word vector features are extracted using natural language processing-related techniques, and the URL information features and host information features are extracted under the dataset. The extracted host and URL information features and word vector features are fully fused.

(2) We use CNN to get deep local features in the data and use Attention mechanism to adjust the weights, and concatenate CNN and Attention mechanism to get more effective data features; we use IndRNN to get global features in the data and it is processed in series with CNN and attention mechanism to get more comprehensive data information, which is used for the analysis and detection of network malicious URL.

II. Algorithm models

2.1 Characteristic Analysis

The extracted data feature information has a significant impact on the accuracy of malicious URL detection, so the effectiveness of the extracted data feature information is very important. Therefore, this paper extracts the URL information features and host information features, and uses the natural language processing technology to extract the word vector features based on URL for malicious URL analysis and detection experiments.

2.1.1 Host Information Features

Host information features are host-related information obtained from the attributes of the hostname in the URL, including the host's time, location, identity, and other information. By learning these obtained information through algorithms, effective host information features can be obtained for malicious URL analysis and detection experiments, and in this paper, we extract 20 kinds of host-related information from URL data to enhance the detection effect of host information features in this paper experiment.

2.1.2 URL information features

This URL information feature is obtained from the surface characteristics of the URL data. The URL data in the data set are some strings, numbers, etc., which are not very friendly to the learning of the algorithm. Through the extraction and transformation of the URL data, we can get the learning friendly URL information feature for the algorithm, so as to promote the training and learning of the algorithm model, and carry out the malicious URL

analysis and detection experiment, 21 kinds of URL related information are extracted from the URL data, which are the characteristics of the URL information in this paper to improve the detection effect.

2.1.3 Word vector features

Word vectors transform words into dense vectors in natural language processing. For similar words, the corresponding word vectors are also similar. It was first proposed by Hinton et al, and generally has rich semantic and contextual information [20] [21]. When the data set is URL, the URL needs to be processed into the data format understood by the computer, so the URL is first transformed into a vector or matrix, and then the algorithm is trained. In this paper, we extract word vector features from URL by classical training word2vec word vector model, in the hope that this feature will have rich semantic and context information, and the subsequent experimental data will also show this. The extracted word vector feature is to extract the URL from a new perspective and obtain more comprehensive features, which has a positive impact on the research of URL detection. Therefore, this paper combines word vector features, host and URL information features to improve the detection effect of malicious URL detection.

2.2 Feature Fusion

In order to get the hidden features of all kinds of features more fully to improve the detection effect of malicious URL detection, we need to learn the information features of host and URL to get block features with the help of algorithm, secondly, face features are obtained by learning word vector features and block features with the help of algorithm[22][23][24], and finally, face features are input to the CATIR tandem joint algorithm in this paper for training and used for malicious URL analysis and detection, and the overall schematic diagram is as follows Figure 1.

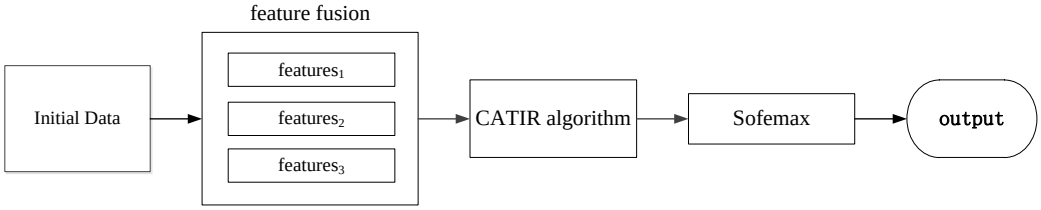

Fig 1: Overall schematic

2.3 Catir algorithm

In this paper, the CATIR tandem joint algorithm is obtained by first combining the CNN algorithm with the Attention mechanism, and then combining the CAT tandem joint algorithm with the IndRNN algorithm, which is called the CATIR tandem joint algorithm. CNN algorithm is a classical traditional machine learning algorithm. It is a neural network composed of convolution layer, pooling layer and full connection layer. It has unique advantages for dealing with similar network structure. In CNN algorithm, convolution group is composed of convolution layer and pooling layer for feature learning, and full connection layer is used for judgment and classification. First, the fused features are input from the input layer $M \in \{(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)\}$, $(x_1, x_2 \ldots x_n)$, $y_n \in (1,0)$ denotes the label of the input data, and the convolution layer is calculated as：

$$F_j^l = f(\sum_{i \in G_j} F_i^{l-1} * T_{ij}^l + N_j^l) \tag{1}$$

Style: *l* represents the number of layers, Gj represents the input features at a time, F represents each neuron, and N represents each offset vector.

Pooling layer is the key to reduce the size of the input matrix and accelerate the calculation speed in CNN algorithm, and can effectively prevent over fitting and reduce the dimension of features. The calculation formula of pooling layer is:

$$F_j^l = f(\frac{1}{m}\sum_{i\in G_j} F_i^{l-1} + N_j^l) \tag{2}$$

Style: *l* represents the number of layers, Gj represents the input features at a time, F represents each neuron, and N represents each offset vector, and m represents the window size of the pooling layer.

However, the poor correlation between input and output in the CNN algorithm has a certain impact on the detection effect, but the correlation between input and output in the Attention mechanism performs better and performs better to compensate for this drawback, Attention mechanism focuses on the relationship existing between focus, context, is learning the thinking pattern of the human brain, and is able to extract semantic features, and performs better for, Therefore, the CNN algorithm and the Attention mechanism are jointly processed in tandem here, drawing on the convolutional learning ability of the CNN algorithm and improving the detection ability of the algorithm in this paper by using the Attention mechanism to compensate for the disadvantage of the poor correlation of the CNN algorithm. The features learned from the CNN algorithm are input from the input layer $M \in \{(x_1, y_1), (x_2, y_2)...(x_n, y_n)\}$, $(x_1, x_2...x_n)$, $y_n \in (1,0)$ denotes the label of the input data, and the formula for calculating the weight of attention is：

$$W_i = \frac{\exp(M_i)}{\sum_{i=1}^{b}\exp(M_i)} \tag{3}$$

Style：W_i denotes the calculated attention weights，M_i denotes the feature input, while attention weighting is applied to T. The output feature S is calculated as：

$$S = \sum_{i=1}^{b} W_i M_i \tag{4}$$

The IndRNN algorithm using unsaturated function as the activation function gives good results after learning training and performs extremely well for processing sequences with short step lengths IndRNN algorithm is formulated as follows：

$$h_t = \sigma(Wx_t + u \bullet h_{t-1} + b) \tag{5}$$

Since the relationships between neurons here are not connected, the IndRNN algorithm is used to increase the connectivity between neurons in the same layer by superimposing multiple layers, and the computation process is shown in the following equation to obtain the hidden layer $h_{n,t}$ with the nth neuron：

$$h_{n,t} = \sigma(W_n x_t + u_n h_{n,t-1} + b_n) \tag{6}$$

The structure diagram of the IndRNN algorithm is shown in Figure 2. The Weight and ReLU activation functions are the roles of arithmetic and cyclic processing of feature realizations at each step, and the BN is the role of normalization of feature realizations at each step.

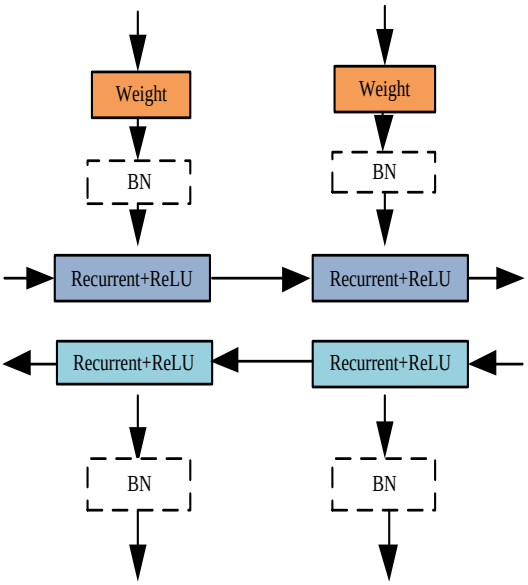

Fig 2: IndRNN structure

The overall structure of the CATIR tandem union algorithm is shown in Figure 3. The features are input to CNN algorithm, attention mechanism, and IndRNN algorithm respectively, and then CNN algorithm, attention mechanism, and IndRNN algorithm are combined in tandem to get CATIR algorithm, which goes through a fully connected layer and finally SoftMax classifier layer to classify and get malicious URL detection results.

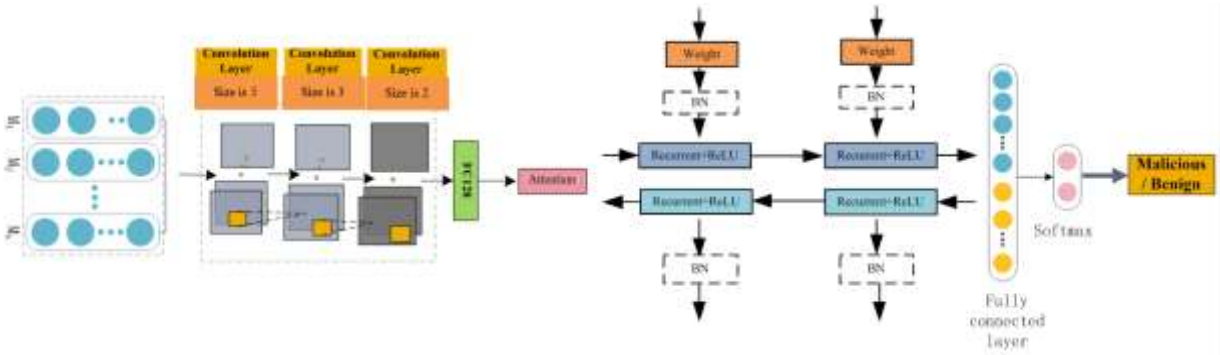

Fig 3: CATIR structure

III. Experiments and results analysis

3.1 Sample Data

The 20,000 URL dataset used in this experiment is composed of 10,000 URLs each from the public dataset PhishTank and the virtuous URLs crawled by the crawler. Since the data crawled by the web crawlers are cluttered, repetitive, and erroneous, the data are simply processed by weight reduction and denoising before feature extraction, and 10,000 benign URLs after processing are used in this paper, and 10,000 malicious URLs downloaded from the public dataset PhishTank are used to complete the URL dataset for this paper.

3.2 Experimental environment

Table 1 shows the software and hardware environment and configuration used in the experiment.

Table 1 Experimental environment

Parameters	value
Operating System	Windows 10 64位
CPU	Intel(R) Core(TM) i7-7770 CPU@ 2.8 GHZ 2.8GHZ
GPU	NVIDIA GTX 1050 4G
Memory	8G/ DDR3/2400MHz
Hard Disk	128G HDD+1T SSD
Python	3.6.1

3.3 Parameter evaluation

The parameter settings in the algorithm directly affect the performance of the algorithm in learning and training. Therefore, this paper obtains the optimal parameters of this algorithm through Numerous experiments. This section describes the optimal parameters for subsequent readers to understand the repeatability of the experimental data in this paper. In Table 2, we show the iteration times, batch processing capacity, the number of convolution cores, the size of convolution core, the size of pooling layer and the partition of test set. In the next section, the setting process of the optimal parameters of the iteration number will be illustrated.

Table 2 Setting of parameters

Parameters	value
filters	16
filters_size	7
batch_size	84
test_size	0.25
ep	10

3.4 Compared experimental analysis

This section will compare and analyse the experimental data results in terms of the validity of the iteration count parameter, the validity of the word vector features, the validity of the URL word vector features, and the detection results of other machine learning algorithms, and finally complete the analysis of the algorithm's detection results.

3.4.1 Validity of the number of iterations parameter

This section shows the process of setting the optimal parameters for debugging the variable number of iterations, and how to make the detection effect optimal when setting the number of iterations for different amounts of data under the same data set, which is analyzed and illustrated by the data results obtained from a large number of experiments.

From Fig. 4, with data volumes of 5,000, 10,000, 15,000 and 20,000, and with the parameters for the number of iterations set to 5, 10, 15 and 25, it is observed that the highest detection results are obtained with a number of

iterations of 10, regardless of the data volume, as the data is under-fitted when the number of iterations is less than 10 and over-fitted when the number of iterations is greater than 10.

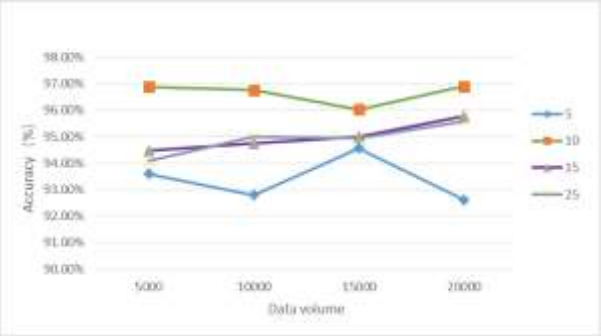

Fig 4: Comparison of different iteration

3.4.2 Validity of word vector features

The subsection conducts experiments by adding or removing word vector features at data amounts of 5000, 10000, 15000, and 20000 to illustrate the effectiveness of word vector features with the experimental data obtained for malicious URL detection.

From Fig. 5, when there is no word vector feature, the best detection result is 96.64%, independent of the data size, while with the addition of word vector feature, the best detection result is 96.9%, and the detection result is better at data size 5000, 10000, 15000 and 20000, so we can see the effectiveness of word vector feature for malicious URL detection.

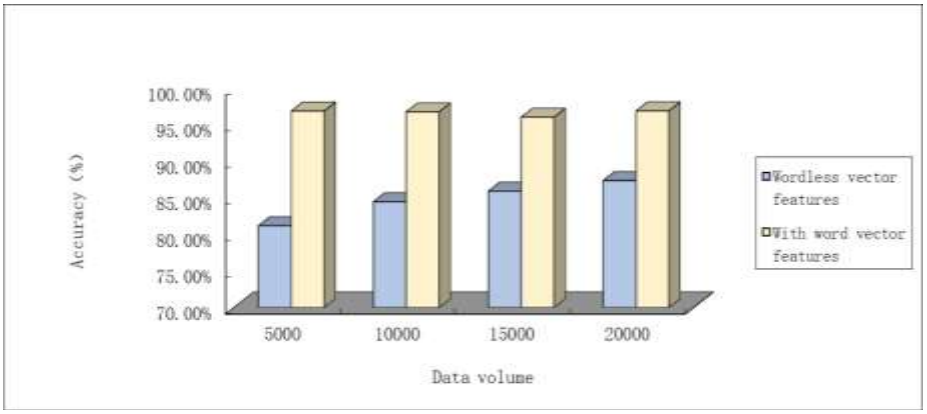

Fig 5: Effectiveness of word vector features

3.4.3 Validity of URL information features

This subsection conducts experiments by adding or removing URL information features at data amounts of 5000, 10000, 15000, and 20000, to illustrate the effectiveness of URL information features with the experimental data obtained for malicious URL detection.

From Fig. 6, the best detection result obtained without URL information features is 87.38%, and when URL information features are added, the best detection result obtained is 96.9%, and there are better detection results for data amounts of 5000, 10000, 15000 and 20000 respectively, so it can be seen that URL information features for malicious URL detection The effectiveness of the URL information features for malicious URL detection can be seen.

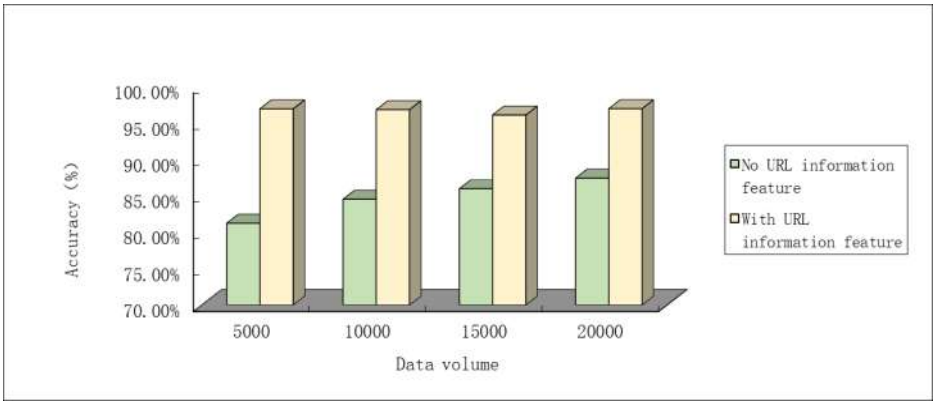

Fig 6: Effectiveness of URL information feature

3.4.4 Detection results of other machine learning algorithms

In this section, comparison with single machine learning KNN algorithm, Random Forest, GaussionNB, deep learning model CNN algorithm and IndRNN algorithm is performed to propose the effectiveness of CATIR tandem joint algorithm in malicious URL detection based on the validation results, as can be seen from the comparison of the detection results in Table 3, with the amount of data of 5000, 10000, 15000, and 20,000, the KNN algorithm obtained the best detection result of 82.08%, the best detection result obtained by the random forest algorithm is 96.85%, the best detection result obtained by the GaussionNB algorithm is 83.82%, the best detection result obtained by the CNN algorithm is 96.67%, the best detection result obtained by the IndRNN algorithm is 92.85%, and the best detection result obtained by the CATIR tandem joint algorithm is 96.9%, and no matter in which case the data volume is 5000, 10000, 15000, 20000, the detection result obtained by the CATIR tandem joint algorithm is better than the KNN model, random forest, GaussionNB algorithm, CNN algorithm, and IndRNN algorithm. The CATIR tandem federation algorithm has a higher detection result than the KNN model, random forest (rf), GaussionNB algorithm, CNN algorithm and IndRNN algorithm in detecting malicious URLs, regardless of the data volume of 5,000, 10,000, 15,000 and 20,000, and has a more obvious improvement. Therefore, it can be seen that the CATIR tandem federation algorithm improves the results of malicious URL detection.

Table 3 Comparison with other models

Number of samples	KNN	GaussionNB	CNN	IndRNN	CATIR
5000	78.65%	72.3%	89.3%	90.9%	96.88%
10000	80.33%	76.38%	94.45%	91.9%	96.76%
15000	82.08%	83.82%	95.1%	90.77%	96.03%
20000	81.84%	78.49%	96.67%	92.85%	96.9%

IV. Conclusion

In order to improve the accuracy of malicious URL detection in the network, this paper proposes a multi-network tandem fusion detection algorithm (CATIR), which first extracts word vector features through word2vec model and fuses them with URL information features and host information features to obtain feature information with richer semantic and contextual relationships; then the CNN and Attention mechanism network The CNN, Attention mechanism and IndRNN are finally fused in tandem to identify and detect malicious URLs through the fully connected layer and SoftMax classifier. The experimental results demonstrate that the detection accuracy of CATIR is improved by 9.43% on average compared to the traditional algorithm.

Acknowledgments

This work was supported by the 2021 Philosophy and Social Sciences Program of Guangzhou, 2021GZGJ145.

References:

[1] Q. Zhou, Malicious URL detection method and apparatus, terminal, and computer storage medium: US2018262522, 2018-09-13.
[2] Sungjin. Kim, Jinkook Kim, Brent ByungHoon Kang. Malicious URL protection based on attackers' habitual behavioral analysis. Computers & Security, DOI:10.1016/j.cose.2018.01.013, 2018.
[3] Quan. Tran. Hai, Seong. Oun. Hwang, Detection of malicious URLs based on word vector representation and ngram. Journal of Intelligent & Fuzzy Systems, 35(2).1-13, 2018.
[4] S. N. Tao, Method and device for detecting malicious URL.US9935967, 2018-04- 03.
[5] J. G. Jiang, J. M. Chen, Kim-Kwang. Raymond. Choo, Chao Liu, Kunying Liu, Min Yu, Yongjian Wang. A Deep Learning Based Online Malicious URL and DNS Detection Scheme. Springer International Publishing, 2018-05-16.
[6] Selvaganapathy, Nivaashini, Natarajan. Deep belief network based detection and categorization of malicious URLs. Information Security Journal: A Global Perspective, 27(3).145-161, 2018.
[7] Z. Y. Li, Y. Shi, Q. Xue, Malicious URL identification based on machine learning. Communication Technology, 53(02). 427- 431, 2020.
[8] W. P. Wang, S. Y. Wu, H. Song, S. G. Zhang, J. X. Wang. A malicious URL detection method combining multiple features. Hunan Province, China: CN109922052A, 2019-06-21.
[9] M. Zhang, Research on URL security detection technology based on machine learning. Harbin Institute of Technology, 2019.
[10] W. Rong, CNN-based malicious request detection. Xi'an University of Electronic Science and Technology, 2019.
[11] R. Li, Research on malicious URL detection method based on DPI data. Beijing University of Posts and Telecommunications, 2019.
[12] M. Y. Li, Design and implementation of a URL-based malicious access identification system. Beijing University of Posts and Telecommunications, 2019.
[13] H. Z. Sha, Q. Y. Liu, T. W. Liu, Z. Zhou, L. Guo, B. X. Fang, Survey of malicious web page identification. Journal of Computer Science. 39(03), 2016.
[14] Ahmad. Alqwadri, Mohammad. Azzeh, Fadi. Almasalha, Application of Machine Learning for Online Reputation Systems. International Journal of Automation and Computing. PP 1-11, 2021.
[15] B. W. Liu, Y. Q. Wang, G. Y. Lin, Phishing detection algorithm based on structured documents. Computer engineering and design. 40(10), 2019.
[16] Q. S. Bi, X. C. Liang, S. Q. Chen, Phishing website detection based on mRMR-RF feature selection and XGBoost model. Computer application and software. Computer application and software.
[17] J. Xu, Z. H. LV, Overview of Honeypot. Security science and technology. 2021, (02).
[18] G. B. He, Malicious intrusion detection technology in network information management. Electronic technology and software engineering. 2017, (07).
[19] O. B. Ma, X. J. Liu, X. D. Tang, Y. X. Zhou, Y. C. Hu, Malicious URL detection method based on semi supervised learning, Application of computer system. 29(11), 2020.
[20] Z. H. Yue, Z. Xue, X. W. Shen, Y. L.Wu. Research on PHP webshell detection method based on semantic analysis. Communications technology. 53(12), 2020.
[21] Park. SeJoon, Kang. ChulUng, Byun. Yung. Cheol, Extreme Gradient Boosting for Recommendation System by Transforming Product Classification into Regression Based on Multi-Dimensional Word2Vec. SymmetryVolume 13, Issue 5. PP 758-758, 2021.
[22] S. Q. Luo, S. W. Tian, L. Yu, J. Yu, H. Sun, Android malicious code detection based on texture image and active vector space. Computer Applications,38(4).1058-1063,2018.
[23] S. Q. Luo, S. W. Tian, H. Sun, et al. A study of malicious code classification strategy with deep belief network. Small Microcomputer Systems, 38(11). 2465-2470, 2017.
[24] L. Jia, Research on stock trend prediction based on LSTM and social media text information. Central China Normal University, 2019.

[25] Kumi. Sandra, Lim. ChaeHo, Lee. SangGon, Malicious URL Detection Based on Associative Classification. EntropyVolume 23, Issue 2. 2021

[26] C. C. Luo, S. Su, Y. B. Sun, Q. J. Tan, M. Han, Z. H. Tian. A Convolution-Based System for Malicious URLs Detection [J]. Computers, Materials & ContinuaVolume 62, Issue 1. PP 399-411, 2020.

[27] D. Ma, L. Wan, Q. Q. Cheng, Z. Q. Sun, Application of attention CNN in malicious code detection. Computer science and exploration, 15(04), 2021.

[28] Aditya. Khamparia, Sagar. Pande, Deepak Gupta et al. Multi-level framework for anomaly detection in social networking.Library Hi Tech,38(2) , 2020.

[29] A. Praveena, R. N. Devandra. Kumar, Sreeja. B. P A Simple Detection and Escaping Mechanism from Social Network Attacks.International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8(6s3), 2019.

ISSN: 0010-8189