Phishing URL Detection Based on TCN and Transformer

Mengyuan Liu, Yuchen Xie, Zhou Yu

Abstract— **The rapid development of the Internet has brought great convenience to people's daily life. However, the fraudulent behavior of lawless elements through phishing links has become more and more intense, and has seriously jeopardized the safety of people's lives and property. At present, the main technologies for detecting phishing links are based on blacklists, machine learning, but these technologies require a lot of manual labeling, which is time-consuming and unstable. After an in-depth study of the phishing link problem, we propose a network model TTCN, which firstly embeds URL links at character level and word level, then extracts feature representations from TCN and Transformer respectively, fuses these feature representations, and classifies them through fully connected output. The experimental results show that this model achieves 93.61% accuracy in recognizing phishing links, which can cope with the fraud problem generated by phishing links and helps to maintain network security and protect people's lives and properties.**

Index Terms—**Phishing Links, Deep Learning, Temporal Convolutional Networks, Transformer.**

I. Introduction

The rapid development of the Internet, on the one hand, greatly facilitates people's daily life, on the other hand, it exposes many drawbacks and various network security problems. In 2023, the China Internet Network Information Center released the 51st Statistical Report on the Development Status of the Internet in China, which showed that the proportion of Internet users who encountered phishing website fraud was 24.5%, up 0.7 percentage points compared with December 2021, and phishing website fraud is on the rise [1]. Unlawful elements carry out a large number of attacks through phishing, induce users to visit phishing links, and carry out illegal fraudulent behavior after stealing users' real information, which seriously violates people's life and property security. Therefore, in-depth study of the phishing link problem, put forward accurate detection and identification means to stop phishing attacks at the source, can reduce the occurrence of network fraud events to a certain extent, and has important practical application significance.

The most commonly used detection method at the beginning is based on the black and white list method, first establish a black and white list, and then judge the link to be detected, if it is in the black list, it is a phishing link, if it is in the white list, it is a legitimate link, the method is simple, but there is a lag, and it only applies to the situation where the link has been clearly linked as a phishing link, and it can not be judged in a timely manner for some unknown links. With the development of artificial intelligence technology, machine learning and deep learning are beginning to be used for detection. The machine learning detection method requires manual data labeling and manual extraction of URL features, which consumes more manpower and time in the feature extraction process. Deep learning, on the other hand, does not require much human intervention, and is able to realize automatic extraction of URL features, detection and identification through complex neural network models. Therefore, in this paper, by theoretically analyzing and researching the existing deep learning-based phishing link detection models, we propose a TTCN-based detection model, which first preprocesses the data of URLs from the character level and word level respectively, converts them into two-dimensional embedding vectors, and then classifies them through the TTCN model, to obtain the accuracy of the phishing link detection model.

Manuscript received August 25, 2023

Mengyuan Liu, School of computer science and technology, Tiangong University, Tianjin, China.
Yuchen Xie, School of computer science and technology, Tiangong University, Tianjin, China.
Zhou Yu, School of computer science and technology, Tiangong University, Tianjin, China.

II. Related Works

In this section, we discuss some of the commonly used phishing link detection methods.

A. Early Phishing Detection Methods

Prakash proposed a PhishNet model that first decomposes existing blacklist parent class entries to generate new subclasses of URLs, and then detects unknown malicious URLs by using an approximate matching algorithm from the perspective of IP addresses, hostnames, and directory structures [2]. This method increases computational cost, storage consumption, and maintenance difficulty as the number of malicious URLs in the blacklist increases. Rao use a combination of whitelisting and visual similarity to first filter legitimate and suspicious websites based on a whitelist, and then extract unique keypoint features to visually compare the similarity between legitimate and suspicious web pages [3]. The method targets known malicious URLs and is highly dependent on correct visual input, which may produce different results when the input webpage image features are changed.

B. Machine Learning

Joshi proposed an integrated machine learning based classification method to identify malicious URLs [4]. Firstly, static lexical features are extracted from URL strings and then

URL classification is performed using a random forest model with a decision tree estimator. Vahid extracted thirty types of features such as IP address, URL length, subdomain, etc., for each link in a web link dataset and then used a machine learning model for detecting phishing websites respectively [5]. Machine learning methods can correctly identify the type of unknown URLs to a certain extent, but they are highly dependent on manually extracted features, and the final prediction results are closely related to the selection of features. Especially when the dataset is complex and large, the time cost of manually extracting features is high.

C. Deep Learning

With the development of artificial intelligence technology, deep learning has achieved remarkable results in several fields. Le proposed an end-to-end model URLNet, which extracts characters or words from URL strings from different levels, taking into account the semantic representations of characters and words, and then applies convolutional neural networks to automatically learn the features for phishing website detection [6]. Yang designed a keyword-based convolutional gated recurrent unit neural network, replacing the original pooling layer with gated recurrent units for feature extraction, and detecting URL links from the perspective of attack types [7]. Farid proposed the Texception architecture, which adopts two paths in parallel, one path is used for extracting the character-level information, and the other is used for extracting the word-level information. Then phishing links are detected using multiple parallel convolutional layers [8]. Asiri also explored hybrid based phishing link detection that combines phishing links with other features such as HTML content, JavaScript and then uses deep learning models to determine whether it is a benign link or a phishing link [9].

III. METHOD

The TTCN detection model proposed in this paper consists of Embedding Layer, Time Convolutional Network Layer, Transformer Layer, Fully Connected Layer and Classification Output Layer. The model architecture is shown in Figure.1.

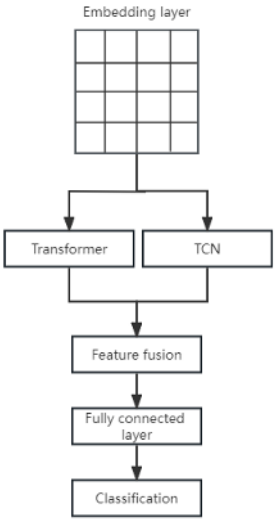

Figure 1: The model architecture diagram

A. TCN Branch

The word vector matrix obtained by the embedding layer processing is input to the time convolution network. In this path, this paper uses a three-layer temporal convolutional network to extract features from URL data, setting the number of channels to 200, the size of convolution kernel to 3, and setting the expansion factor of the expansion convolution to [1,2,4]. The powerful properties of convolution are utilized in combination with the expansion factor to achieve feature extraction across time steps. After stacked temporal convolutional layers, the output-tcn of the path is obtained, and finally the output-tcn is fed into the linear layer and subjected to the relu activation function for subsequent feature fusion with the output of another path.

B. Transformer Branch

The word vector matrix obtained by the embedding layer processing is fed into the transformer layer. In this path, four transformer encoding layers are stacked in this paper. The following describes one of the transformer encoding layers, firstly, we get the contextual relationship through positional coding, add the initial embedding with the positional embedding, pass the obtained word representation matrix L into the encoder, do three different linear transformations to get Q, K, V, then get the current output through dot product operation and softmax, finally get the output-transformer of this path after two layers of full connectivity as well as residual connectivity.

IV. EXPERIMENT

A. Sources of Data Sets

Zhang obtained normal URLs from Alexa website and collected phishing URLs from Phishtank and OpenPhish to form the dataset PhishTrim [10]. This experiment uses this dataset as well as some of the data provided by the ISCXURL2016 dataset, which is merged and organized to obtain a total of 52,317 sample data. Among them, there are 26764 normal URL samples and 25553 phishing URL samples, and the number of positive and negative samples is close to the sample balance. We divide the dataset into training set, validation set, and test set according to the ratio of 6:2:2.

B. Experimental Environment Configuration

Before conducting the experiment, we need to configure the experimental environment. We choose Pytorch as the development framework and develop in mainstream Python language. In order to improve the training efficiency and reduce the time consumption of model training, we selected RTX 3090 GPU on Autodl platform for the experiment. The specific information is shown in Table I:

Table I
Detailed configuration information

Environment Configuration	Configuration Information
Operating System	Windows10
GPUs	RTX 3090
Random Access Memory	8G
Toolkits	Anaconda
Development Language	Python3.7
Development Framework	Pytorch

C. Evaluation Indicators

Evaluation criteria such as precision, recall, and F1-Score are also used in addition to the regular accuracy rate when performing experimental evaluations. Where the accuracy rate is for the whole model to get the weight of all the correctly categorized results in the overall result.

The precision rate is the proportion of outcomes that the model actually predicts correctly out of the total number of outcomes predicted as correct.

Recall is the proportion of outcomes that the model actually predicts correctly out of the total number of outcomes that are actually correct.

The F1-Score combines precision and recall and is a reconciled average of the two.

D. Comparison of Experimental Models

In order to verify the advantages of the proposed model in this paper, URLNet from the literature [6] is chosen as a reference, and ablation experiments are conducted to compare with the TTCN model designed in this paper. Under the same embedding method, the comparison of the evaluation indexes of each model at the end of each training is shown in Table II.

Table II
Results of evaluation indicators for each model

Model	Accuracy	Precision	Recall	F1-Score
URLNet	91.91%	92.01%	91.85%	91.89%
Transfomer	85.83%	87.48%	86.14%	85.73%
TCN-BiLSTM	88.43%	90.02%	88.65%	88.35%
URLNet -Transformer	93.53%	93.71%	93.66%	93.53%
TTCN	93.61%	93.79%	93.71%	93.61%

From the experimental results in the above table, it is found that the accuracy, precision, recall, and F1 score obtained by the TTCN proposed in this paper after training on the same dataset are higher than other models. Among them, the accuracy rate of TTCN model is 1.7% higher than URLNet and 0.1% higher than URLNet-Transformer, which indicates that temporal convolutional networks can be applied to text processing and can extract relevant sequence features well. The accuracy rate obtained by the TTCN model is 5.18% higher than that of the TCN-BiLSTM model and 7.78% higher than that of the Transfomer model, indicating that transformer can capture the long distance dependent features of URLs more adequately than BiLSTM, and that the combination of transformer and TTCN has more effective feature extraction advantages than a single transformer, which side by side indicates that the URL detection model proposed in this paper achieves better classification results.

V. Conclusion

In this paper, we propose a phishing URL detection model based on the combination of TCN and Transfomer, which firstly preprocesses the text of the URL link to obtain a vector representation combining character-level and word-level, and then inputs it into the TCN model and the transformer model respectively, and combines the extracted features of both of them to perform the final classification of the website link. Experiments have proved that the model proposed in this paper has a better classification effect, but there are certain shortcomings, such as the experimental model uses feature fusion, which appears to have a large time overhead and other problems. For future research, we will continue to update the fishing URLs from the perspective of shortening the training time of the model, expanding the dataset, and adopting other word embeddings for training, such as One-Hot, Word2Vec, GloVe, and so on.

References

[1] 51st Statistical Report on the Development of the Internet in China, March 02, 2023, https://www.cnnic.cn/n4/2023/0303/c88-10757.html
[2] P. Prakash, M. Kumar, R. R. Kompella and M. Gupta, "PhishNet: Predictive Blacklisting to Detect Phishing Attacks," 2010 Proceedings IEEE INFOCOM, San Diego, CA, USA, 2010, pp. 1-5, doi: 10.1109/INFCOM.2010.5462216.
[3] R. S. Rao and S. T. Ali, "A Computer Vision Technique to Detect Phishing Attacks," 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, India, 2015, pp. 596-601, doi: 10.1109/CSNT.2015.68.
[4] A. Joshi, L. Lloyd, P. Westin, et al. Using lexical features for malicious URL detection -- a machine learning approach[J]. 2019.
[5] S. Vahid, M. Darabi, et al. Phishing Detection Using Machine Learning Techniques. arXiv 2020, arxiv:2009.11116.
[6] Le. H, Pham. Q, Sahoo. D, Hoi. S.C. URLNet: learning a URL representation with deep learning for malicious URL detection. arXiv 2018, arXiv. 1802.03162.
[7] W. Yang, W. Zuo and B. Cui, "Detecting Malicious URLs via a Keyword-Based Convolutional Gated-Recurrent-Unit Neural Network," in IEEE Access, vol. 7, pp. 29891-29900, 2019, doi: 10.1109/ACCESS.2019.2895751.
[8] F. Tajaddodianfar, J. W. Stokes and A. Gururajan, "Texception: a Character/Word-Level Deep Learning Model for Phishing URL Detection," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 2857-2861, doi: 10.1109/ icassp40776.2020.9053670.
[9] S. Asiri, Y. Xiao, S. Alzahrani, S. Li and T. Li, "A Survey of Intelligent Detection Designs of HTML URL Phishing Attacks," in IEEE Access, vol. 11, pp. 6421-6443, 2023, doi: 10.1109/ACCESS.2023.3237798.
[10] L. Zhang and P. Zhang, "PhishTrim: Fast and adaptive phishing detection based on deep representation learning," 2020 IEEE International Conference on Web Services (ICWS), Beijing, China, 2020, pp. 176-180, doi: 10.1109/ICWS49710.2020.00030.