Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/18/20230965

Detection of malicious websites across multiple classes using n-gram features and VGG based on URL analysis

Qichen Liu

Faculty of Engineering, The University of Sydney, Sydney, 2206, Australia

z12321a@mail.nwpu.edu.cn

Abstract. Due to the ubiquity of the internet, cyber-attacks implemented through websites have become a severe issue with high frequency and appreciable overall financial damage. Detecting malicious URLs has become one of the most common solutions to tackle this threat, which is widely applied in the market and researched. Inspired by relevant work on URL classification using n-gram techniques and convolutional neural networks in other research areas, a method for detecting malicious websites using n-gram statistical features of URLs and a VGG-style neural network has been developed, which aims to provide classification for multiple website classes with arbitrary URL input lengths. Experimental results show that the method proposed in this paper provides an average accuracy of 96.60% on the 5-class ISCX-URL2016 dataset and 96.33% on the 4-class Malicious URLs dataset, which is 1.5 times larger. A further comparison reveals that the accuracies are competitive with similar methods for binary classifications that also use either n-gram features or a VGG-based network.

Keywords: URL, multi-class, n-gram features, VGG.

1. Introduction

With the rapid development of networks over the past decade, the internet has permeated all aspects of daily life, making website usage a routine occurrence. However, as a side effect, the threat of malicious activities based on websites is also significant. In the first quarter of 2022 alone, up to 370 thousand harmful webpage links were detected daily [1], and in the third quarter, more than 1.2 million phishing attacks that heavily relied on websites were observed [2]. The damage caused by these attacks was substantial, causing over 52 million financial losses in just one state in the United States and creating more than 300 thousand victims in 2022 [3]. Under such circumstances, technologies for preventing website-based attacks deserve close attention. Webpage identification based on Uniform Resource Locators (URLs), commonly known as links, is one of the most common solutions for detecting malicious sites. This technique allows dangerous URLs to be identified and users to be warned before an attack takes place. Moreover, compared to other approaches such as network-traffic-based ones, URL-based methods are easy to capture, allowing websites to be predicted even before massive amounts of data are available. Additionally, URL-based methods have a low achievable response time [4], and their computing resource consumption makes them suitable for use on a larger scale.

URL-based identification has been widely adopted by the industry, with mature products available to the public. Google Safe Browsing warns users when they access dangerous site or file links, and it has been used on more than 5 billion devices. Safe Links from Microsoft allows URLs in emails to be

DOI: 10.54254/2755-2721/18/20230965

checked at the time of click [5]. Other products, such as McAfee WebAdvisor [6] and Tencent URL Safe [7], also perform similar functions to prompt users about dangerous URLs. The success of these applications proves the effectiveness of URL-based methods.

In this article, we propose a malicious website URL classification method based on n-gram features and a VGG-style Convolutional Neural Network (CNN). Inspired by the idea of using image recognition techniques for different tasks, we implement a neural network modified from VGGNet [8], one of the most well-known image classification networks, to extract high-level patterns from "images" of n-gram features and perform identification of malicious URLs. Furthermore, our method is designed to deal with URLs of arbitrary lengths and under multiple website classes, providing more flexibility than similar n-gram-and-neural-network methods. The remainder of this paper is organized as follows: Section 2 analyzes related work on URLs and their machine-learning classification, Section 3 introduces the proposed method, Section 4 discusses the performance experiment of the method, and Section 5 specifies conclusions and future work.

2. Related work

2.1. URL components

As a compact string representation for a resource, a typical Internet URL, such as the one in Figure 1, may contain five major components. Scheme instructs the protocol for communication; network location describes the site where a resource is stored at, which sometimes may contain additional access information like the port number and login details; path further indicates the way to find a resource on the host site; query string is an optional part where detailed request parameters are specified [7]. For example, in Figure 1, the keywords of our Internet search are included in the query string, allowing the search engine to know what information we are looking for. Finally, a fragment in an URL can be appended to indicate web browsers to jump to a specific part after a webpage is loaded.

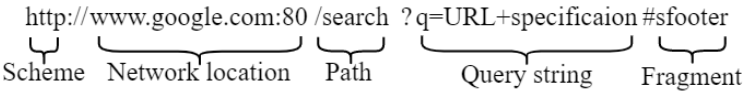

Figure 1. The components of a typical URL.

2.2. Feature extraction

Feature extraction is a crucial process in machine learning, where raw data is transformed into structured features that represent the original data. In the context of URL classification, various feature extraction methods have been examined in recent years. Gupta et al. extracted nine lexical features, such as total length, number of top-level domains, and number of dots, to represent URLs. Ren, Jiang, and Liu employed natural language processing (NLP) techniques to extract semantic information from URLs by breaking them down into words separated by special symbols like dots and colons. They then introduced a Word2Vec layer that converts words into vectors while preserving word relationships, to generate URL representations.

Similarly, Korkmaz et al. used NLP techniques to perform character-level n-gram extraction, collecting the occurrence of single characters and two-to-five-character combinations as features. The use of character-level information has been proven to be an effective method, as demonstrated in multiple studies. L. Zhang and P. Zhang applied the skip-gram strategy to learn the character relationships in URLs. They combined multiple consecutive characters from URL strings into character embeddings, which were then used in conjunction with a bidirectional long short-term memory (Bi-LSTM) network to extract additional information and concatenated with character embeddings as URL representations. Grambedding, proposed by Bozkir, Dalgic, and Aydos, employed an n-gram strategy for embedding where N-character combinations were analyzed throughout the URL and each represented by an ID number sequentially, forming a vector of ID numbers to represent the original URL.

DOI: 10.54254/2755-2721/18/20230965

2.3. Neural networks

Various researches of the application of neural networks on URL classification have been conducted. One of the frequently used networks is long short-term memory (LSTM). As a type of recurrent neural networks, LSTM is able to handle contextual information of words in sentences in NLP, which is similar to character sequences in URLs. Vecile et al. applied an LSTM model modified for the binary classification of malicious URLs [9]. The research from Ren, Jiang and Liu adopted a bidirectional LSTM with an attention layer, which provided 98.06% accuracy on their dataset [10]. A similar network was also implemented by Bozkir, Dalgic and Aydos to perform classification at the final stages [11].

With the capability of capturing high-level patterns from their input, convolutional neural networks (CNN) were also applied as URL classifiers. Research from Korkmaz et al. showed the capability to deal with n-gram statistics of URLs with 1-D convolutional neural networks . Besides, according to the comparison from Alshingiti et al., the accuracy of CNN classifier can outperform LSTM and LSTM+CNN classifiers with a certain same set of extracted features [12]. Researchers also realized the potential of applying outcomes from other areas in URL classification. For example, a VGG16 network, which was originally proposed for image recognition [13], was modified by Li et al. and used to process comprehensive embedding data extracted from URLs with Bi-LSTM and other techniques [14].

3. Method for detection

In this paper, a method to classify URLs based on their character-level n-gram statistics using a VGG-style network is introduced. The method is designed with an expectation that (1) requests only an URL as the input information, and (2) is flexible to work on datasets having arbitrary URL lengths and number of classes without structural modification. In the following section, details will be discussed.

3.1. Data preprocessing

The contents of each original URL are processed before further steps. Firstly, uppercase characters are converted to lowercases, ensuring case differences make no effect to following parts. Secondly, schemes and starting “www” in network locations are removed, since they are repetitively used in all kinds of links and does not indicate them to be malicious or benign. Thirdly, characters are filtered to ensure only numbers, alphabets and a selected list of symbols including “-!#$&*+,./:;<=>?@_`{}|~” are kept. The filtering makes sure to maintain a consistent list of analyzed characters and removes illegal ones in URLs like Asian characters. Fourthly, all symbols are converted to the same character “/”, making sure these characters without semantic meanings only act as separators and are indifference to subsequent steps.

3.2. Feature extraction

N-gram refers to contiguous N elements in an array, where the elements can be characters, words and so on. In our method, a character-level bigram strategy is applied, which is similar to the one in the research of Korkmaz et al. With such strategy, consecutive two-character combinations of each input preprocessed URL are analyzed [15]. For example, from an input “ap/org”, combination “ap”, “p/”, “/o”, “or” and “rg” will be analyzed and their occurrences are recorded. Going through the dataset, the number of all existed combinations and their occurrence statistics are collected. With the limited 37 types of characters after preprocessing, the number of possible combinations is no more than 1369 regardless of the length of the original URLs. The collected occurrence times are further normalized to range 0 to 1, and processed with principal component analysis (PCA). PCA is a technique to create new uncorrelated variables that successively maximize variance from the original ones [16], making it possible to truncate the variable number to a specific one without losing significant information. In the implemented method, 1024 variables are kept after applying PCA. These features are further reshaped into 32*32 2-D “images” as the input feed into the following neural network for classification.

DOI: 10.54254/2755-2721/18/20230965

3.3. Classification

A VGG-style network is adopted to perform classification, being expected to accurately capture high-level patterns of different URL types from their n-gram feature images. The network structure is shown in Figure 2.

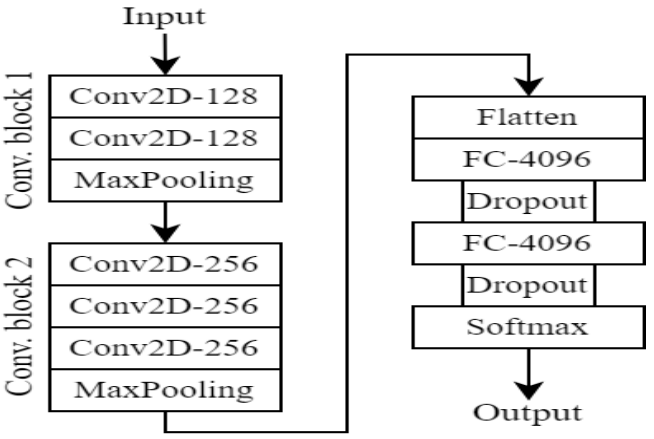

Figure 2. The structure of adopted VGG-style network.

Firstly, two convolutional (conv.) layer blocks are involved, with the first block having two 128-filter conv. layers and a 2*2 max pooling layer, while the second having three 256-filter conv. layers and a 2*2 max pooling layer. The number of conv. blocks is reduced from the original five in VGG16 network to two, ensuring that the output images size of our blocks, 8*8, is close to 7*7 from the original network who has a larger input size. Meanwhile, all conv. layers use Leaky ReLU [17]. as their activation functions and their filters have the same 3*3 size. After conv. layer blocks, a flatten layer is appended for reshaping data, then two fully connected layers with 4096 units are used and a softmax layer is followed for outputting classification result. For each fully-connected layer, an additional 0.5-ratio dropout layer is added after it to mitigate potential overfitting in neural networks, which is also used in the similar practice from Li et al. [14].

4. Experimental results

4.1. Experiment environment

The experiment is carried out on a Google Colab runtime having a 4-core Intel(R) Xeon(R) CPU @ 2.20GHz CPU and a Tesla T4 GPU with 25.5GB RAM and 15.0 GB VRAM. The experiment code is written in Python 3.9.16 with TensorFlow 2.12.0, Scikit-learn 1.2.2, Pandas 1.5.3 and Numpy 1.22.4 library [18].

4.2. Datasets

Following datasets are used in this experiment:

ISCX-URL2016: from this dataset, totally 155250 website URLs are collected and used with their type labels, including 35377 benign, 95307 defacement, 2691 malware, 9955 phishing and 11920 spam. As the focus of this research, the dataset is used in both model development and evaluation [18].

Malicious URLs dataset: this is a dataset with 651191 samples from a subset of ISCX-URL2016 and multiple other sources. Four classes, benign, defacement, phishing and malware, are included. For this experiment, a 400k-sample subset is collected. The dataset is involved to evaluate the performance of the method on a larger dataset [19].

For each dataset, five times of run were performed in a sequence, with random dataset separation on a 3:1:1 ratio for training, validation and testing. The neural network was trained through 6 epochs with a batch size of 256, using an Adam optimizer at 0.001 learning rate. One time of training history on

DOI: 10.54254/2755-2721/18/20230965

ISCX-URL2016 dataset is shown in Figure 3 below with training accuracy, training loss, validation accuracy and validation loss.

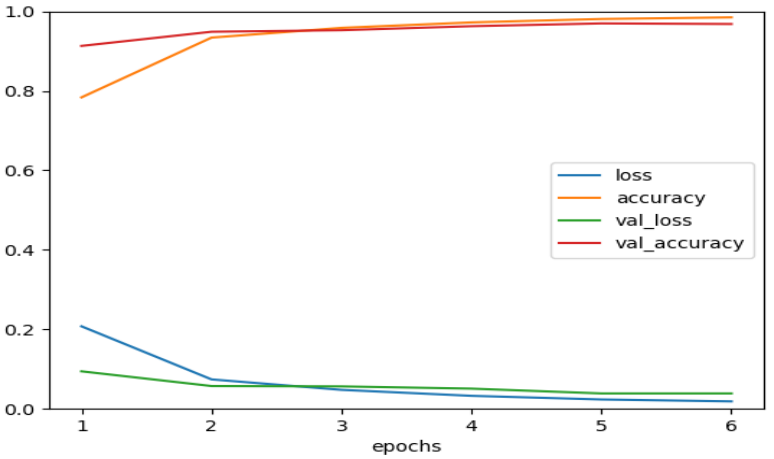

Figure 3. History for one time of training on ISCX-URL2016.

4.3. Performance evaluation

The accuracy, macro precision, macro recall and macro F1 score were collected from the results for evaluation. Accuracies based on five runs on each dataset are shown in Table 1 below. On ISCX-URL2016 dataset, decent 96.60% average accuracy was reached, with the highest value at 96.71%. On Malicious URLs dataset, though sample class number is decreased to 4, a 1.27% decrease on average accuracy was observed. Such a decrease should come from the increased diversity brought by the 158%-larger dataset size comparing with ISCX-URL2016.

Table 1. Classification accuracies on datasets.

Dataset (Class No.)	Accuracy (%)	
	Average	Highest
ISCX-URL2016 (5)	96.60	96.71
Malicious URLs (4)	95.33	95.63

Remaining measurements including precisions, recalls and F1 scores are shown in Table 2. Similar to the accuracies, with the more than 1.5-time larger size of Malicious URLs dataset, a decrease of 0.0371, 0.0862 and 0.0727 was observed on average precision, average recall and average F1 score respectively, comparing with the performance on ISCX-URL2016. The decline on average recall is more significant than the one on precision, showing the model is more likely to miss some malicious websites with the size-increased dataset.

Table 2. Precisions, recalls and F1 scores on datasets.

Dataset (Class No.)	Average			Highest		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
ISCX-URL2016 (5)	0.9450	0.8925	0.9157	0.9588	0.9017	0.9188
Malicious URLs (4)	0.9079	0.8063	0.8430	0.9336	0.8311	0.8517

Furthermore, comparison of results between three URL classification methods having similar components are shown in Table 3. Despite the potential influence of content differences between

DOI: 10.54254/2755-2721/18/20230965

datasets, the method in this article provides a 7.7% higher accuracy than the one from Korkmaz et al. on a more complex five-class task, both using features from n-gram statistics and classifying with a convolutional network. However, with 0.24% lower accuracy and 0.0502 lower F1 score, the method in this article shows a slightly lower performance comparing with the one of MUI-VB. Though the two methods both adopt VGG-style networks, MUI-VB uses a deeper CNN with two additional conv. layers and an extra pooling layer, and a comprehensive feature extraction process including another Bi-LSTM network to extract multiple types of embedding features. Beside the number of class of datasets, such significant method complexity difference can be the major reason affecting the performance.

Table 3. Comparison between methods with similar components.

Method	Dataset	Class No.	Accuracy (%)	F1 score
Korkmaz et al. [15]	High-risk URL	2	88.90	N/A
MUI-VB [14]	Alexa & PhishTank	2	96.84	0.9659
This article	ISCX-URL2016	5	96.60	0.9157

5. Conclusion

This article introduces a method for analyzing the URLs of malicious websites. Using N-gram statistics as features, a VGG-style network can classify both benign and multiple types of malicious website URLs accurately, regardless of their lengths. The method is tested on two datasets of different sizes, the ISCX-URL2016 dataset and the Malicious URLs dataset, achieving an average accuracy of 96.60% and 95.33%, respectively. Compared to two other binary classification methods with similar elements and tasks, this method shows good performance, with an accuracy only 0.24% lower than the best method while working on a dataset with three more classes. However, limited by time and resources, this article's contribution is restricted, and there are still areas for future work. Alternative approaches for dimensionality reduction, apart from PCA used in this article, should be explored. The PCA method requires loading all data into memory for analysis, which may require approximately 8G memory space for a 155k-URL dataset and more if working with more samples. Additionally, more detailed tuning, such as using larger N for N-gram and a deeper neural network with larger image input, can be researched to improve potential performance.

References

[1] Ortega O B, Segura J R. Protocolo básico de ciberseguridad para pymes[J]. Interfases, 2022 (016): 168-186.

[2] Wang C, Chen Y. TCURL: Exploring hybrid transformer and convolutional neural network on phishing URL detection[J]. Knowledge-Based Systems, 2022, 258: 109955.

[3] Sharif M H U, Mohammed M A. A literature review of financial losses statistics for cyber security and future trend[J]. World Journal of Advanced Research and Reviews, 2022, 15(1): 138-156.

[4] Gupta B B, Arachchilage N A G, Psannis K E. Defending against phishing attacks: taxonomy of methods, current issues and future directions[J]. Telecommunication Systems, 2018, 67: 247-267.

[5] Lakshmi V. Beginning Security with Microsoft Technologies[J]. Beginning Security with Microsoft Technologies, 2019.

[6] Day G. Security in the Digital World: For the home user, parent, consumer and home office[M]. IT Governance Ltd, 2017.

[7] Dong R, Zhang Y, Zhao J. How green are the streets within the sixth ring road of Beijing? An analysis based on tencent street view pictures and the green view index[J]. International journal of environmental research and public health, 2018, 15(7): 1367.

[8] Wang L, Guo S, Huang W, et al. Places205-vggnet models for scene recognition[J]. arXiv preprint arXiv:1508.01667, 2015.

[9] Vecile S, Lacroix K, Grolinger K, et al. Malicious and Benign URL Dataset Generation Using Character-Level LSTM Models[C]//2022 IEEE Conference on Dependable and Secure Computing (DSC). IEEE, 2022: 1-8.
[10] Ren F, Jiang Z, Liu J. A bi-directional LSTM model with attention for malicious URL detection[C]//2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, 2019, 1: 300-305.
[11] Bozkir A S, Dalgic F C, Aydos M. GramBeddings: A New Neural Network for URL Based Identification of Phishing Web Pages Through N-gram Embeddings[J]. Computers & Security, 2023, 124: 102964.
[12] Alshingiti Z, Alaqel R, Al-Muhtadi J, et al. A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN[J]. Electronics, 2023, 12(1): 232.
[13] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
[14] Li J, Wang D, Zhao C, et al. MUI-VB: Malicious URL Identification Model Combining VGG and Bi-LSTM[C]//Proceedings of the 2022 3rd International Conference on Control, Robotics and Intelligent System. 2022: 141-148.
[15] Korkmaz M, Kocyigit E, Sahingoz O K, et al. Phishing web page detection using N-gram features extracted from URLs[C]//2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE, 2021: 1-6.
[16] Jolliffe I T, Cadima J. Principal component analysis: a review and recent developments[J]. Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences, 2016, 374(2065): 20150202.
[17] Maas A L, Hannun A Y, Ng A Y. Rectifier nonlinearities improve neural network acoustic models[C]//Proc. icml. 2013, 30(1): 3.
[18] Url T. Gesamtwirtschaftliche Auswirkungen der Exportgarantien in Österreich[J]. WIFO Studies, 2016.
[19] Johnson C, Khadka B, Basnet R B, et al. Towards Detecting and Classifying Malicious URLs Using Deep Learning[J]. J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl., 2020, 11(4): 31-48.