Payload-based Packet Classification using Deep Learning

Michael Shell[1], Homer Simpson[2], James Kirk[3], Montgomery Scott[3], and Eldon Tyrell[4], *Fellow, IEEE*
[1]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA
[2]Twentieth Century Fox, Springfield, USA
[3]Starfleet Academy, San Francisco, CA 96678 USA
[4]Tyrell Inc., 123 Replicant Street, Los Angeles, CA 90210 USA

The abstract goes here.

***Index Terms*—Packet Classification, Deep Learning, CNN, RNN**

I. Introduction

THIS demo file is intended to serve as a "starter file" for IEEE Transactions on Magnetics journal papers produced under LaTeX using IEEEtran.cls version 1.8b and later. I wish you the best of success. Next subsection will mention xxxxxx II-B

mds
August 26, 2015

A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here

Subsubsection text here.

II. Related Work

Recently, there are many researches and technologies for packet classification in networks. In the existing research, there is a rule-based packet classification method. Recently, research on deep learning has been developed and research on packet classification using deep learning has been actively carried out. Packet classification using deep learning is a method of automatically classifying packets without human intervention. In this chapter, we study the rule-based packet classification research and the packet classification studies that utilized deep-learning.

A. Rule-based Packet Classification

Rule-based packet classification is a method of classifying packets entering the network according to predefined rules. The rule-based packet classification method is classified by using the header information of the network packet. Therefore, rule-based packet classification is performed based on the source and destination IP and port of the packet header.

So, there are limitations to the rule-based packet classification method. Since the packet is classified using the information of the packet header, if the information of the packet that doesn't match the packet is received, the packet can't be classified or classified differently. In addition, because of rule-based packet classification with IP and port information of packet header, there are local limitations. Therefore, when a new network accesses or packets of a new network occur, there is a problem that a new rule must be redefined.

fangfan Li et al. [] At least it was used to lower the dependency of packet header information. Using this approach, they found that our packets haping device uses HTTP and TLS-handshake fields in their matching rules, and only for the first packet in each direction. If there is similar information in the header information of the incoming packet using the header information found in the first packet, it is classified as the packet of the same type. Although the IP and port number of the packet is used less, the method of classifying the subsequent packets by using the header information of the first packet also depends on the header information.

B. Classify Packets Using Deep Learning

Several studies using deep learning have been conducted recently. Also, researches to utilize machine learning in networks are actively being conducted. Accordingly, studies are being actively carried out to perform packet classification using deep learning of network.

Wei Wang et al. uses CNN model to classify malware traffic and general traffic. First, if 5-tuple (source IP/port, destination IP/port, protocol) are the same among the packets, one flow is defined as one dataset. In addition to the flow dataset, packets are defined as a session set as a dataset. The session dataset is a case where 5-Tuple is paired with the same flow and the same source IP/port and Destination IP/port cross each other. When data is divided into actual flow and session, data set is constructed by removing information of IP and MAC address. The reason is that IP and MAC address can show certain characteristics. The flow and session data sets constructed above are composed of 28 * 28 data similar to the MNIST dataset. The constructed dataset is used to learn CNN model to classify malware traffic and general traffic. In this paper, the result of classification of malware traffic and general traffic is 100

M. Lopez-Martin et al. used packet classification using CNN and RNN combination of deep learning model. The packet data is extracted from the header information and the payload data in the packet using the DPI Tool and used as learning data. The extracted learning data was used as input data to the combined

Manuscript received December 1, 2012; revised August 26, 2015. Corresponding author: M. Shell (email: http://www.michaelshell.org/contact.html).

model of CNN and RNN. They showed that CNN and RNN combined better than CNN and RNN models. In both of the above papers, learning was performed on the deep learning model by adding the header information of the packet to the dataset. In such a case, the classification accuracy may be high because the header information can be certain information that characterizes the data to some extent. Therefore, in this paper, we will perform learning only with payload data of application layer except header information of packet.

III. Data Preprocessing

In this section, the dataset used to perform the traffic classification is introduced to preprocess the PCAP file provided by Broadband Communication Research Group [] for the deep learning learning. A PCAP (Packet Capture) file captures and stores network packets using programs such as wireshark and TCPdump. Therefore, in this paper, traffic dataset provided by Broadband Communication Research Group is divided into flow unit and packet unit, and preprocessing packet is used as learning data.

A. Data Split

he size of the original PCAP file provided is about 59 GB, and there are a total of 769,507 flows in the file. Figure 1 is a graph showing the number of flows per application. The info file provided with the PCAP file, there is a labeling of the traffic data. Labeling is divided into three categories: application type, protocol, and application name for the corresponding traffic flow. Because of the labeling file, accurate label information corresponding to the group truth in the flow-based traffic classification can be obtained by using the deep learning model proposed in this paper.

For the preprocessing of the supplied data, 8 kinds of application were selected based on the number of flows. The 8 applications selected the application that has traffic of the actual network and the number of flow is more than 2000. The 8 most common Label names are the Remote Desktop Protocol (RDP), Skype, SSH, Bittorrent, HTTP-Facebook, HTTP-Google, HTTP-Wikipedia and HTTP-Yahoo. The application layer payload of the selected flow internal packet is filtered and extracted, and 8 application layer payload data files are generated using the extracted payload data.

B. Learning Data Generation

he process of converting the application layer payload data file created in the above section into input data suitable for deep learning learning will be described. The application payload data file is divided into per packet unit and per flow unit, and each learning data is generated. Flow is the same as the 5-tuple of the packet's header information, and packets generated within 3600 seconds of the previous packet are bundled into the same flow.

Each packet is extracted from each application layer payload data, and the elements of the payload data of the packet are grouped by 4 bits into one unit of learning data. Therefore, one pixel of the learning data represents the number of 0 to 15.

The learning data is generated by the packet unit and the flow unit for each application. The learning data for each application per packet generates learning data by arbitrarily extracting packets of 8 applications (RDP, SSH, Skype, BitTorrent, Facebook, Wikipedia, Google, Yahoo) from the application layer payload data file. In each application, 10000 random packets were extracted to extract a total of 80,000 learning data. One packet of each randomly extracted application is resized according to the size of the payload. Therefore, the payload size of each application packet is extracted as 36 (6 * 6), 64 (8 * 8), 256 (16 * 16), and 1024 (32 * 32). Figure 2 shows 16 * 16 of the payload data of an arbitrary packet for each extracted application. If the extracted payload size is smaller than the set size, the size is adjusted by zero padding to match the set size.

The learning data of the flow unit is arbitrarily selected for each application in the application layer payload data file. The selected flow fetches the packets from the first N packets as the number of packets (N) per predetermined flow. They are packets that has undergone a preprocessing process as in the learning data of each packet unit. The learning data of the flow unit extracts 2000 flows for each application from the application layer payload data file and has a total of 16000 flows. The number of N is set to 30, 60, and 100, and packets of each flow are fetched by the corresponding number to generate learning data.

Then, it is expressed as a one-hot vector with 8 lengths so that each of 8 applications can have label data. A one-hot vector label is a vector with a value of only one element. A one-value index can be defined as a label representing an application. Therefore, two sets of learning data are used as the learning data packet or flow data and label data indicating the application data.

IV. Deep Learning Models

This chapter describes learning models that are used to classify network traffic using deep learning. The Deep Learning model used is the Convolution Neural Network (CNN), the Residual Network (ResNet), the Recurrent Neural Network (RNN), the Long Short-Term Memory LSTM) and the Convolution and Recurrent Neural Network (CNN + RNN). The Deep Learning model was supported by Keras, and the ResNet and CNN + RNN models were generated using Keras' CNN and RNN models simultaneously.

CNN and ResNet models are commonly used for information extraction, sentence classification, face recognition, and image classification. CNN is a structure that extracts characteristics of data and grasps patterns of features. In the case of RNN and LSTM, it is a model specialized for repetitive and sequential data learning. Therefore, the previous learning data is reflected in the current learning using the circulation structure. It is generally used for the composition of speech, wave and text.

Therefore, in the case of CNN and ResNet, it is used to classify using imaged packet unit data generated through preprocessing. RNN, LSTM, and CNN + RNN models are used to classify sequential data, so they are used to classify

learning data in flow units that contain sequential information of network traffic.

A. Convolution Neural Network Architecture

CNN model among the deep learning models for classifying network traffic will be described. The model architecutre of CNN is composed of input layer, Convolution layer, and Pooling layer and Fully connected layer. The input layer uses the payload and label of the packet converted into learning data. Packets are used as input data in the input layer in the form of N N (N = 6, 8, 16, 32) like images. Then, the feature of each packet data is convolved through the kernel of two Convolution layer, and output is generated through filter and activation process. In the pooling layer, it is the process of reducing the size of the output through the convolution process. It simply reduces the size of the data, cancels noise, and provides consistent features in fine detail. Finally, the Fully connected layer extracts the prediction value according to the last 8 classes by activation.

Figure 3 is CNN architecture.

B. Residual Network Architecture

fefefef

C. Recurrent Neural Network Architecture

NN is a network architecture that can accept inputs and outputs regardless of input data length, and can be implemented variously and flexibly as needed. Therefore, the architecture of RNN used in this paper is composed of multi-layer as shown in figure 4. In the RNN, the number of packets per flow (30, 60 and 100) is received at the input layer in order to learn flow unit data. The number of units to be set is then output to the number of applications learned in the output layer through the RNN cell.

D. Long Short-Term Memory Architecture

n addition to the existing RNN model, LSTM determines whether to keep the weight value by adding another feature layer called a cell state. Through this, we solve the phenomenon that the weight value is not maintained as the distance between information and information of one input data in the existing RNN becomes longer, and the learning ability decreases. LSTM is more persistent than existing RNN because it keeps updating the past data. The cell state is responsible for adding or deleting information. The structure of the LSTM model is configured as shown in figure 5. It is a single layer different from RNN.

The advantage of LSTM is that each memory control is possible and the result can be controlled. However, there is a possibility that the memory may be overwritten, and the operation speed is slower than that of the conventional RNN. Therefore, it is composed of single layer different from existing RNN model.

E. CNN + RNN Architecture

fefefef

V. Model Tunning

eeeeeeeeeeefefefefefefefef

A. Data Split

fefefefefefefefefef

VI. Conclusion

The conclusion goes here.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.