Journal of Physics: Conference
Series

PAPER • OPEN ACCESS

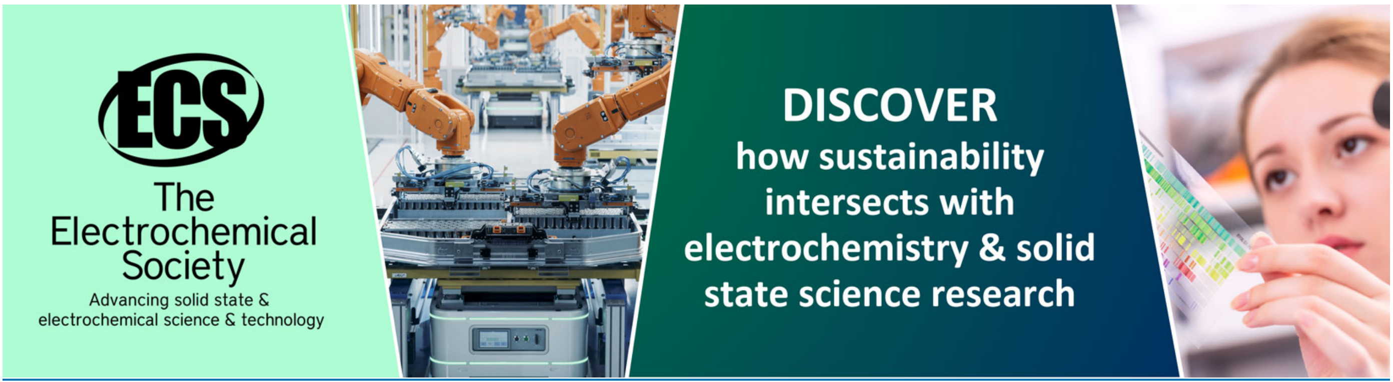
Network Traffic Feature Engineering Based on Deep Learning

To cite this article: Kai Wang *et al* 2018 *J. Phys.: Conf. Ser.* **1069** 012115

View the article online for updates and enhancements.

You may also like

- Network Traffic Anomaly Detection Method Based on a Feature of Catastrophe Theory
  Yang Yue, Hu Han-Ping, Xiong Wei et al.
- NBFTP: a dedicated data transfer system for remote astronomical observation at Dome A
  Si-Yuan Huang, , Ce Yu et al.
- Investigation of Traffic Classification Applied to an Astronomical Data Transmission Network of the XAO Using Deep Learning
  Jie Wang, Hai-Long Zhang, Na Wang et al.



ECS
The Electrochemical Society
Advancing solid state & electrochemical science & technology
DISCOVER
how sustainability intersects with electrochemistry & solid state science research

This content was downloaded from IP address 18.119.118.99 on 26/04/2024 at 17:40

Network Traffic Feature Engineering Based on Deep Learning

Kai Wang, Liyun Chen, Shuai Wang and Zengguang Wang

Shijiazhuang Campus of Army Engineering University, Shijiazhuang, 050003, China.
Email: wangkai8a8@gmail.com

Abstract. Aiming at extracting traffic features using manual selection and feature combination methods in current network traffic feature engineering, it is difficult to accurately extract the features of common traffic characteristics. A network traffic feature extraction method based on autoencoder model is proposed. The method first converts the first 144 bytes of the network data packet into a numeric code, and then acts as an input to the stacked autoencoder, then outputs a 49-dimensional feature through a 4-layer network encode. Using the dataset collected in laboratory to verify the method, experiments show that the feature extracted by this method can effectively extract network traffic characteristics, the extracted features are representative, and can use low-dimensional data to represent high-dimensional data.

1. Introduction

Network traffic feature engineering is the basic work of various network control and optimization, and it is also the basic requirement of QoS in network management [1]. The degree of feature selection and processing directly affects the classification and forecasting capabilities of the model. Literature [2][3][4] studies traffic classification from different algorithmic perspectives. However, in the basic feature engineering, they all based on manual selection and combination of features. Although the above algorithms have achieved good classification accuracy, it is difficult to adapt to the changing network protocol and traffic features due to the fixed feature selection. How to select generic traffic features becomes a fundamental issue.

2. Network Traffic Feature

Traffic features are used to describe and measure network traffic. It is used as an input of traffic classification algorithms and is an important bridge for IP packet recognition to network applications. Moore et al[6]. proposed that the port number, packet interval time, different bytes of the data packet and other features and their combinations have a total of 248 features, these features are widely used in various studies. According to the different levels of traffic features in the data flow, it can be divided into three types: packet header feature, load feature, and flow feature.

2.1. Packet Header Feature

The TCP/IP protocol divides the network into four layers. The data header of each layer consists of a specific format. It has specific meaning and is an important traffic feature. The header features exist in the link layer, network layer, and transport layer. The features of each layer are shown in Table 1.

2.2. Load Feature

The application protocol usually has a specific string to identify the application and other information, through the deep packet inspection technology to match the load characteristics and known application

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

characteristics in the network flow to determine the traffic class, this method has a high recognition rate. However, this method does not recognize the encrypted traffic at the application layer, and there is lag in the maintenance of the signature database.

Table 1. Features in Different Layer of TCP/IP Protocol

Layer	Features
Transport Layer	Source Port, Destination Port, Flags, Window Size
Internet Layer	Packet Length, Source IP Address, Destination IP Address, TOS, TTL, Fragment Offset, Flags
Link Layer	Packet Length, Frame Length

2.3. Flow Feature

The application of encryption technology is a challenge for traffic analysis. Since the payload content is encrypted, the characteristics are mainly reflected in the data flow. The traffic characteristics at the flow level are the statistics of the statistical characteristics shown by multiple packet groups in each flow [5]. Among the many flow features, the packet length and the packet arrival time interval and their statistics are frequently used [7].

3. Feature Engineering Based on Deep Learning

3.1. Autoencoder Model

An autoencoder model is a kind of neural network and it is an unsupervised learning model. It is usually used for dimensionality reduction or feature learning. As shown in Figure 1, the autoencoder consists of two parts, an encoder function $h = f(x)$and a decoder function $r = g(h)$ for reconstruct. The goal of autoencoder is to train a neural network makes $g(f(x)) \approx x$, stacking multiple autoencoders in series becomes a stacked autoencoder (SAE). The purpose of the stacked autoencoder is to extract the high-order features of the input data layer by layer, and gradually reduce the data dimension, thereby turning the complex input data into simple higher-order features. A typical SAE model is shown in Figure 2.

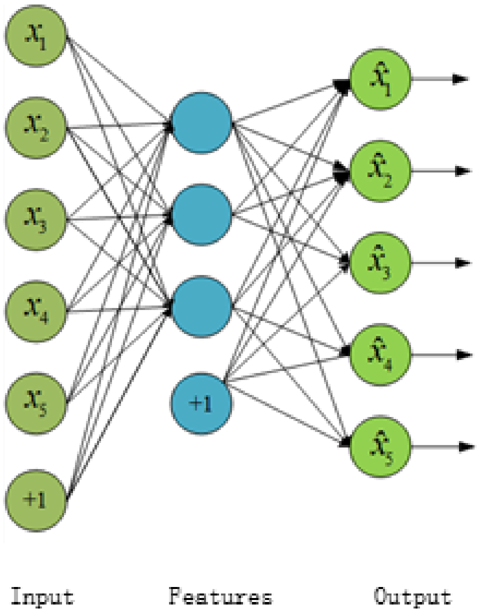


Figure 1. AutoEncoder

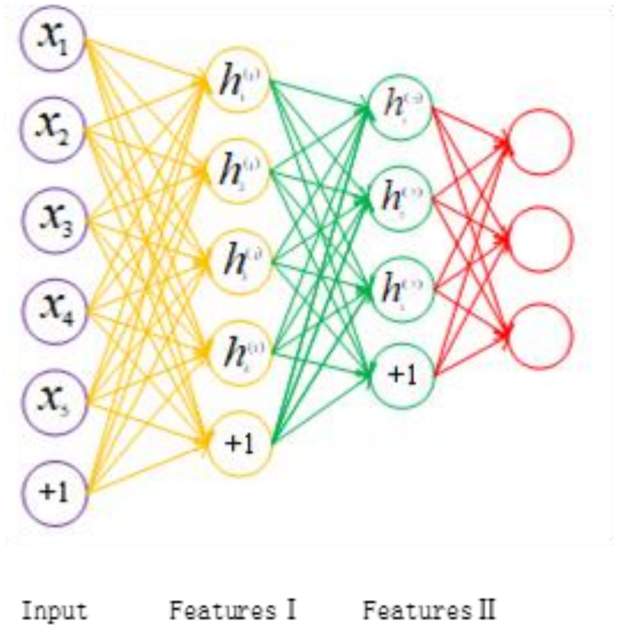


Figure 2. Stacked AutoEncoder

3.2. Auto Feature Engineering Based on Deep Learning

SAE has powerful feature extraction capabilities and can automatically extract features with higher weights in the original data. Compared with the features based on search algorithms, it has the ability of automatic processing. Using SAE to extract traffic features, the first is to convert the packets in the network traffic into values that the neural network can recognize. For each packet P in the flow, it can be expressed in bytes:

$$P = \frac{1}{255}\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{41} & a_{42} & \cdots & a_{mn} \end{bmatrix} \qquad \left(0 \le a_{ij} \le 255\right) \tag{1}$$

where, a_{ij} is the value of each byte of the packet. Figure 3(a) shows the format of a UDP packet encoded, and Figure 3(b) shows the format of a DNS packet encoded, these will be the input of the neural network.

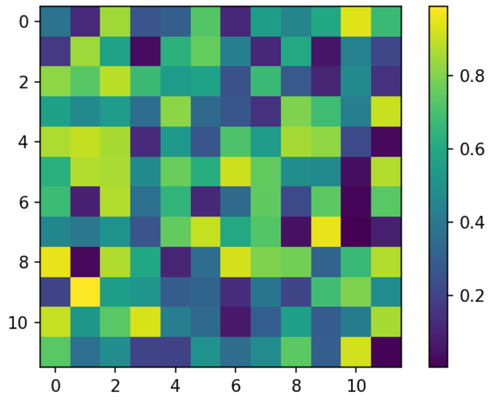


Figure 3(a). UDP Packet

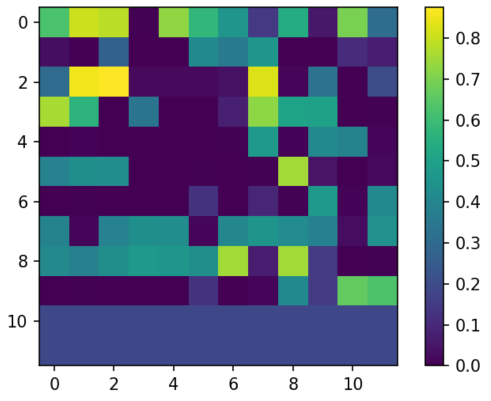


Figure 3(b). DNS Packet

In SAE, the output of the previous layer from the encoder is used as the input to the next level of the self-encoder. For each level of the input x to the autoencoder, the following functions are used for feature reconstruction:

$$y = f\left(Wx + b\right) \tag{2}$$

where, W is the weights of neurons in the encoder, b is the bias of neurons in the encoder. The decoder process of reconstruction defined as:

$$x' = g\left(W'y + b'\right) \tag{3}$$

where, W' is the weights of neurons in the decoder, b' is the bias of neurons in the decoder. To ensure that the reconstructed x' is consistent with the original feature x, using the cross entropy L as the loss function, and minimize L as optimization direction. To make the network converge quickly, parameters are updated using the Adam optimization algorithm.

Using the above principles to construct a four-layer encoder and four-layer decoder stacked autoencoder. Network input is set to 144 neurons, after 4 layers of network encode, the final encoding is 49-dimensional feature parameters. Use sigmoid function joins between layers.

4. Experiments and Analysis

4.1. Datasets

We use datasets collected in a small laboratory network environment to test the effectiveness of the method. Data collection using Wireshark software, running a program on a host, and using different types of applications access to Internet, the collected network flow constitute the dataset of this experiment, the dataset has 10 different protocol network data flow, the data set is described as follows:

Table 1. Describe of Dataset

Protocol	items
ARP/RARP	622
BitTorrent	532
DNS	280
FTP	861
HTTP	762
HTTPS	634
ICMP	265
MySQL	571
Telnet	921
UDP	855

4.2. Weights

The weights of each neuron in the SAE input layer and the first hidden layer reflect the weight of each byte of the data packet in the features of the autoencoder construction. The weight of each byte W_i defined as:

$$W_i = \sum\nolimits_{j=1}^{n} \left| w_{ij}^{(1)} \right| \tag{4}$$

where, *i* is the neurons of input layer, *j* is the neurons of first hidden layer, w_{ij} represents the weights of the *i*-th neuron of input layer and *j*-th neuron of first hidden layer, *n* is the number of the first hidden layer.

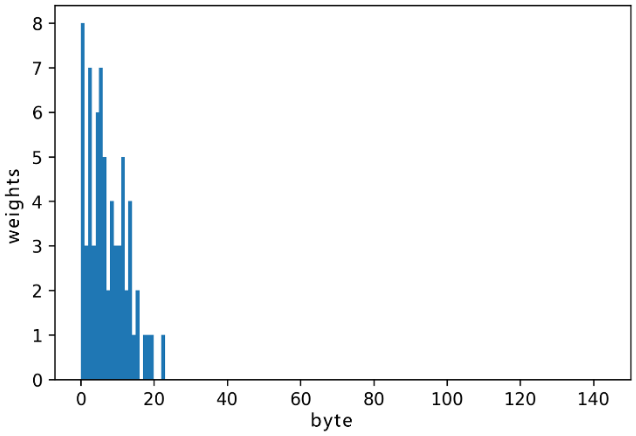
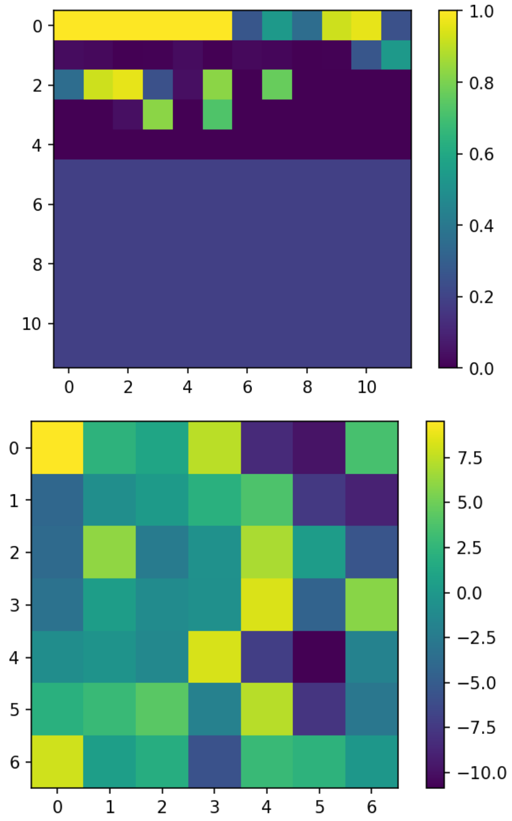


Figure 4. Weights of each byte

Figure 4 shows the weight of each byte in packets. The weight of the packet header is significantly higher, mainly because the feature fields of the protocol are mostly concentrated in the header of the data packet.

4.3. Comparison Between Decoded Data and Raw Data

To measure the ability of the SAE to construct characterization packets, we compare the similarity of the original data with the decoder data. The more similar the output data of the decoder is to the original data, the more representative the features of the SAE structure represent the data packet. Figure 5 shows an ARP packet before encoding, after encoding and decoding. The encoding and decoding of the graphics are very similar. It shows that the features constructed based on SAE can decode the original data better, and the SAE construction features are representative.



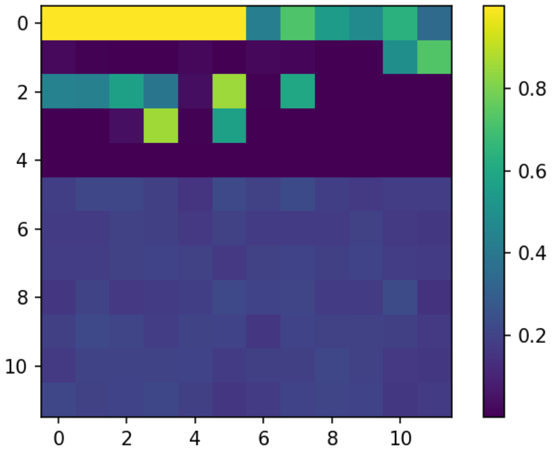


Figure 5. An ARP packet before encoding, after encoding and decoding

5. Conclusion

In this paper, aiming at current network traffic analysis methods, the feature engineering methods usually based on manual selection and combination of features, a traffic feature engineering method using a stacked autoencoder is proposed. This method converts network data packets into a numerical matrix and uses SAE reduces dimension, continuously extracts data features, and finally extracts 49-dimensional data features. The small-scale experiments in a laboratory environment show that this method has good capability of extracting data packets, and the extracted features can describe the original data well.

6. References

[1] Pramitha P, Yu C T and Colin F. A comparison of supervised machine learning algorithms for classification of communications network traffic 2017 *Proc. of the 24th Int. Conf. on Neural Information Processing (Guangzhou)* p 445-54

[2] Yang Z, Li L Z, Ji Q J. Network traffic classification using decision tree based on minimum partition distance [J]. //Journal on Communications 2012 *Jor of Communications* **33** 90

[3] Zhao Y, Tan Y. Improving for network traffic bayes classification method based on correlation information 2016 *Computer Engineering* **42** 80

[4] Liu J W, Zhao Y, Li S H. One method of network traffic classification based on improved k-means algorithm 2017 *Application of Electronic Technique* **34** 14

[5] Liu Z, Wang R Y, Cai X F. Survey on Traffic Features in Internet Traffic Classification 2017 *Application Research of Computers* **34** 14

[6] Moore A W, Zuev D. Discriminators for Use in Flow-based Classification 2005 *Technical Report, Intel Research* (Cambridge)

[7] Khalife J, Hajjar A, Diaz-Verdejo J. A multilevel taxonomy and requirements for an optimal traffic-classification model 2014 *Int. Journal of Network Management* **24** 101