The Applications of Deep Learning on Traffic Identification

Zhanyi Wang
wangzhanyi@360.cn

Abstract

Generally speaking, most systems of network traffic identification are based on features. The features may be port numbers, static signatures, statistic characteristics, and so on. The difficulty of the traffic identification is to find the features in the flow data. The process is very time-consuming. Also, these approaches are invalid to unknown protocol. To solve these problems, we propose a method that is based on neural network and deep learning – a hotspot of research in machine learning. The results show that our approach works very well on the applications of feature learning, protocol identification and anomalous protocol detection.

Keywords: Traffic Identification, Feature Learning, Deep Learning, Protocol Classification, Anomalous Protocol Detection

1 Introduction

The scale and the density of network traffic are growing year by year. The types of protocol are more various. Identifying each flow data is an important problem both in enterprise network and internet. Port-based, signature-based and statistical-features-based identifications are the mainstream approaches. These approaches are either inaccurate or over-dependent on experiences of experts. People hope that the identification system is automatic as much as possible on the premise that it could achieve high accuracy.

In this paper, we will describe some new ideas for traffic identification. We make our effort to solve the problem in traditional procedure. First, we retrospect some classic methods in section 2. Then we introduce the framework of deep learning. We focus on the specific applications in section 4. The applications include automatic feature learning, protocol classification, anomalous protocol detection, and unknown protocol identification. The last parts are the conclusion and future work.

2 Traditional Methods of Traffic Identification

To solve the problem of traffic identification, many researchers explore from different perspectives. One of the earliest methods is working based on special or predefined ports [1]. For example, standard HTTP port is 80. The default port of SSL is 443. But as more and more new protocols don't follow the rule of port registration, the error rate is growing higher.

Signature-based traffic identification has been used after the year 2002. A signature is a portion of payload data that is static and distinguishable for applications, which can be described as a sequence of strings or hex values [2]. The error rate is theoretically lower than 10%. Signature-based method is simple and its efficiency relatively high, so most systems of protocol identification adopt it [3] [4] [5]. However, when a protocol specification changes or a new protocol produces, people must start over for finding valuable signatures. It will be very time-consuming and labor-intensive.

Recently, an approach of automatic classification based on statistical features and machine learning is very popular [6] [7] [8]. This approach depends on the features of traffic transmission, such as the time interval between packets, packet size, repeating pattern, and so on. Then the features are fed into some kind of classifier like Naïve Bayes, Decision Tree and Neural Network [9] [10] [11]. Generally, the training process is offline and time-consuming, while the process of applying model is real-time or near-real-time. It is very similar to the above method that the features depend on experiences of experts. The key point is how to choose appropriate features in a certain application scenario.

3 Artificial Neural Network and Deep Learning

3.1 Artificial Neural Network

Artificial Neural Network (ANN) [12] is an algorithm in machine learning. It is inspired by biological neural network in the brain. A general ANN model is shown as Figure 1.

As shown in the figure, ANN is presented as a system of forward computation of "neurons" in multi-layers. Each pair of neighboring layers is connected. The "neurons" in the same layer have no associations. The "neurons" are also called nodes. Solving an ANN is that we want to optimize the weight parameters between two neighboring layers and a bias parameter (+1 with ring in the feature), then use the model with optimal parameters for the real data.

We intend to focus on the applications rather than the details of the algorithm. ANN is widely used in Pattern Recognition (PR). For example, the classic uses are image classification, speech recognition, text categorization, and so on. Also, some people use it for reducing dimensionality of original features.

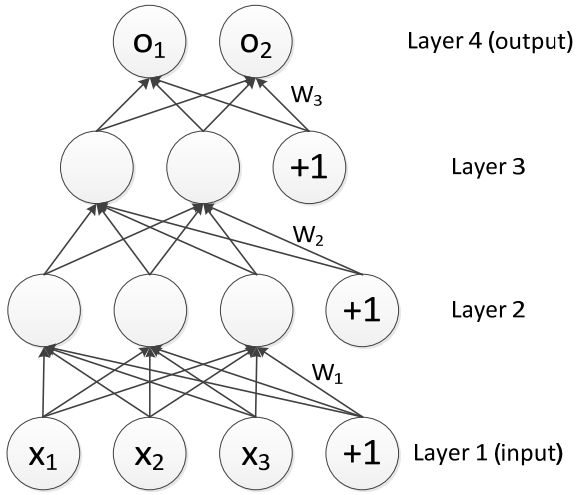

Figure 1. The schematic of ANN.

3.2 Deep Learning

Deep learning [13] is a branch of machine learning based on a set of algorithms. Some of the most successful deep learning methods involve artificial neural networks, such as Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), Deep Belief Networks (DBN) and Stacked Auto-Encoder (SAE) [14]. In recent years, it is more and more popular and applied to computer vision [15], speech recognition [16], and natural language processing [17]. Studies show that deep learning completely surpasses traditional methods in most of areas. Surprisingly, the error rate fell from 26% to 15% in ImageNet Challenge 2012 [18].

The most important advantage of deep learning is replacing handcrafted features with efficient algorithms for unsupervised or semi-supervised feature learning and hierarchical feature extraction. Autoencoder is a perfect example. It aims to learn an efficient, compressed representation for a set of data. The structure of autoencoder is shown as Figure 2.

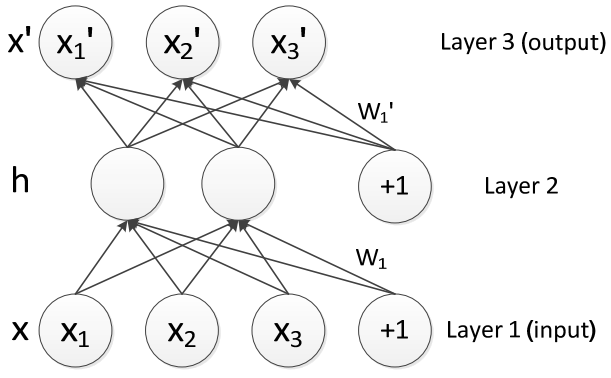

Figure 2. The schematic of autoencoder.

In form, autoencoder is just a kind of ANN. The number of layers is always three. The difference is that nodes in the output layer are the same as the input layer. The nodes in the middle layer are new features that are lower-dimensional representations. It means that data can be

reconstructed after complicated computations. The training process doesn't involve any labels, so it is unsupervised. Generally, data set without labels is very handy to collect. No label means small workload for people. That's the reason people are passionate about autoencoder.

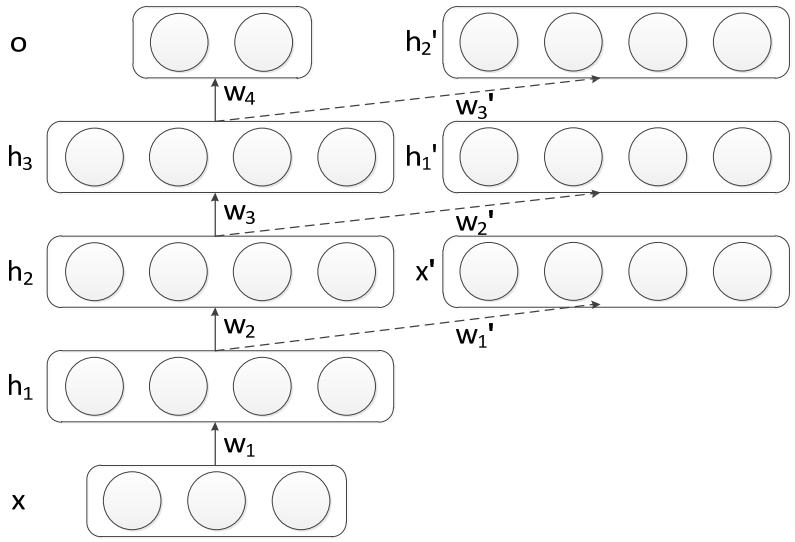

Figure 3. The schematic of SAE.

Further, these structures can be stacked to make up deep networks. As shown in Figure 3, each training results of the middle layer are cascaded. A new network structure which is called Stacked Auto-Encoder (SAE) is formed. Many new features in different depths are learned by this ways. We can append an additional layer with labels once we have large amount of tagged samples and want to train the model more precisely.

In the area of cyber security, a mainstream approach is defining features or signatures by experts' experience. As mentioned above, it looks like ANN and deep learning are omnipotent, but what about using them to our field?

4 Applications

4.1 Data Set

We collect TCP flow data from internal network and joint the payload bytes for every TCP session. A byte is represented by an integer from 0 to 255. Then we normalize the scale to [0, 1]. The length of each payload sequence is 1,000. We pick up about 0.3 million records after wiping off duplications for experiments. The number of protocol types in the data is 58. Because HTTP is apt to identify, this type of data which account for 80% is excluded.

Flow data is very similar to the examples of PR. We can image that a payload record is a picture or a document, and each byte is a pixel or a word. Since pictures and documents can be learned feature and classified very well by deep learning, it's reasonable to expect good performance in

information security.

4.2 Automatic Feature Learning

With regard to feature learning, it generally refers to feature extraction and feature selection. They are two terms in the area of PR. They are fully different. The former is getting features from original feature space to new space by transformation. The features of two spaces may be totally different. The latter is choosing some features from the original set. New features are just chosen, but not transformed. Both ANN and SAE can be used to do feature extraction and selection.

4.2.1 Feature Extraction

In the ANN model, the nodes in deep layers choose and merge the information of the shallow layers automatically. The process is known as training or parameter learning. The nodes in all the layers except the output layer can be taken as features. In the input layer, the nodes represent original features. In a hidden layer, the activations are features that have deeper signification. We can pick up the nodes in any layer as new features, even mix features from different layers.

SAE can be also used for feature extraction. The most remarkable difference between them is supervised or unsupervised. In ANN model labels are necessary while SAE are not. The softmax layer, which is the last layer in Figure 3, is dispensable, when the objects are the hidden features. That's determined by their algorithmic essence. Of course, features trained by SAE with labels will be more precise.

There are two advantages of feature extraction by ANN or SAE model. One is conductive to people to reduce the manual workload. Once inputs of the model and stopping criterion of the iteration are determined, the model will train automatically. The other is that when the training process is finished, the goal of dimensionality reduction is also achieved. Features are mapped to new space. Redundant information is filtered as well.

4.2.2 Feature Selection

We train the ANN model described in Section 3.1 with parameters above. When the error cost is very low (about 10^{-3}) and almost stable (50~60 epochs), we get the optimal parameters. The parameter $W_{ij}^{(1)}$ between the first two layers can be considered as the contribution of the original feature x_j to the hidden layer feature h_i. Once some $W_{ij}^{(1)}$ equals zero, that means x_j is useless to the further propagation. According to the magnitude of parameters, the contribution, that is, the importance of every byte in x can be evaluated.

We take the sum of all absolute weights $|W_{ij}^{(1)}|$ with regard to every node in the input layer as the value

$$v_j = \sum_{i=1}^{n} \left| w_{ij}^{(1)} \right| \quad (1)$$

where n is the number of nodes in the input layer, i.e. the length of payloads in our problem. Then sort the nodes by v_j in descending order. We pick up the top 25, the top 100, and the last 300 ones respectively, and show the results in Figure 4.

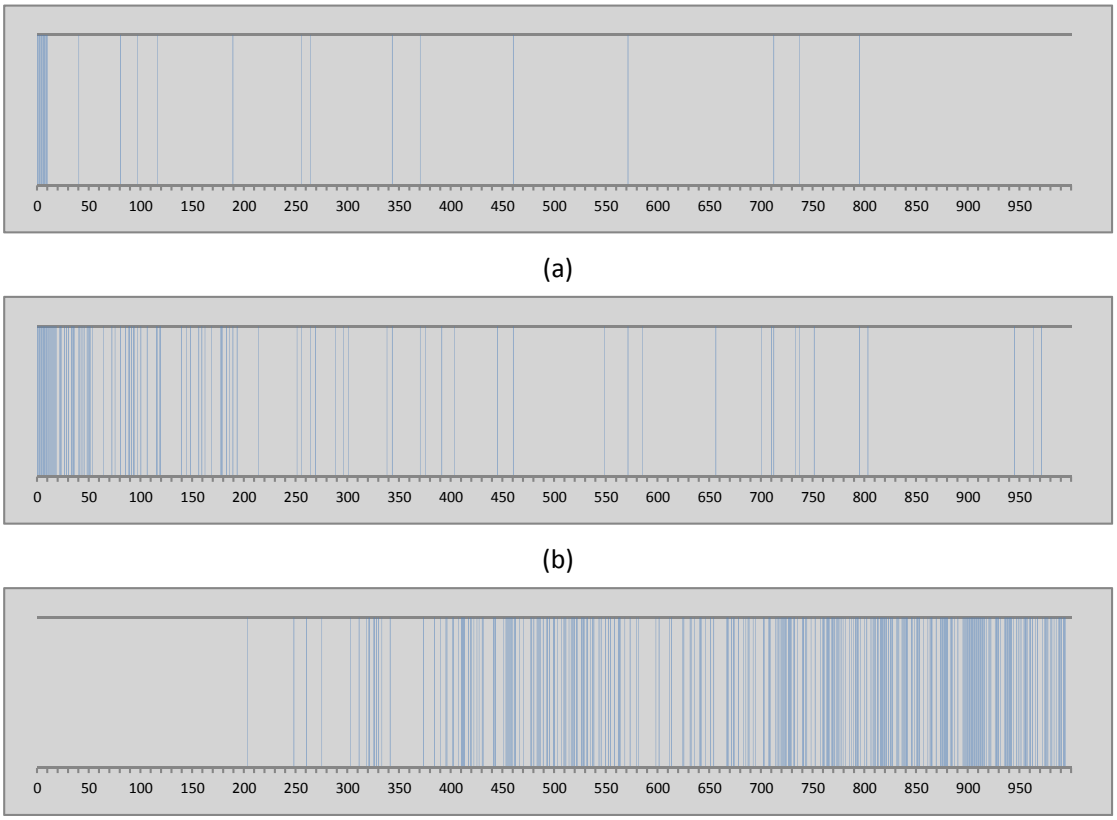

Figure 4. The distribution of byte features in payload data.

Figure 4 (a) shows the distribution of the most important 25 locations in the payloads. Obviously, the results intensively fall in the head of the queue. From 1 to 11, it is fully filled by vertical bar. It means that all bytes in the scale are help to distinguish the protocols. We also focus on the distribution of top 100 bytes in Figure 4 (b). Interestingly, but as is expected, most are centered in the front part. Furthermore, Figure 4 (c) shows the most useless 300 locations of bytes in the payloads. Generally, they fall after No. 300. The densest part is the range around No. 900.

From the above, the results are consistent with intuition and traditional methods. We can restore the payloads and find what the key bytes really are. That's helpful to security professionals for further analysis. That's the process of automatic feature selection by ANN which is supervised.

Autoencoder can be also used for feature selection. The process and formula are very similar to ANN. However, there are two differences between them. First, ANN here is for finding discriminative features across many categories, while Autoencoder is for finding features with something in common in just one category. This leads to opposite direction that we should pay

attention to. If we use the former, the bigger the value in Formula 1 is, the better feature can be chosen. In Autoencoder, the smaller, the better, because we want to hold some stable information in a category. Second, In ANN model, labels are necessary while Autoencoder is not. That's determined by their algorithmic essence.

4.3 Protocol Classification and Anomalous Protocol Detection

Once we have large amount of flow data and labels, the problem of protocol classification can be solved by either ANN or SAE. Usually, SAE with deep structure works better than ANN. Classification is the most direct application that we can think of. The results of classification on our data set list in Table 1.

Table 1 shows the results of top 25 most popular protocols.

Table 1. The results of protocol classification.

Protocol	Precision	Recall
SSL	0.9513	0.9763
HTTP_Proxy	0.9174	0.9090
MySQL	0.9989	0.9993
SMB	1.0000	1.0000
HTTP_Connect	0.9967	0.9930
Whois-DAS	0.9943	0.9777
Redis	0.9985	0.9974
SSH	0.9996	1.0000
Apple	0.9640	0.9728
Kerberos	0.9996	0.9996
DCE_RPC	1.0000	1.0000
NetBIOS	1.0000	1.0000
FTP_CONTROL	0.9970	0.9973
DNS	0.9989	0.9985
Skype	0.9779	0.9722
LDAP	0.9996	0.9992
AppleiCloud	0.9679	0.9689
AppleiTunes	0.9520	0.9617
MSN	0.9453	0.9230
GMail	0.9953	0.9973
BitTorrent	0.9992	0.9992
TDS	1.0000	1.0000
IMAPS	0.9814	0.9654
SMTP	0.9949	0.9883
RSYNC	0.9987	0.9993

We can classify any flow data to a predefined protocol. From the table, we find that whether for the precision or for the recall, the accuracy is enough to use in a real application. On the other hand, besides aiming to the true positive samples, the samples that are incorrectly classified are

also worthy of attention. The rare samples are very likely anomalous or disguised flow. Perhaps, from the incorrect results we can check if the features that human defined are really exact or reasonable.

4.4 Unknown Protocol Identification

There are many flows that can't be recognized clearly by traditional ways. In our experimental data, the proportion of flows which are labeled as "unknown" is over 17%. The fact shows 17% flow data can't be identified even the precise rules are made. Worst of all, "unknown" is the largest group expect the type of HTTP.

To the rest flows which are difficult to identify by traditional ways, deep learning models can distinguish more than half of them and give a probability of each flow. We count the number of the results whose probabilities great than 0.8. 6,716 results in ten thousand match the condition. We list the number and ratio distribution of the top 10 protocols in Table 2.

Table 2. The results of unknown protocol identification.

	number	ratio
SSL	1956	29.12%
DCE_RPC	1454	21.65%
Skype	873	13.00%
Kerberos	517	7.70%
MSN	360	5.36%
Google	311	4.63%
DNS	260	3.87%
RTMP	234	3.48%
TDS	202	3.01%
H323	170	2.53%

Even though we tight the condition of probability, such as greater than 0.9, 54.94% flows are still identifiable. That means that the models can achieve at least half the work that couldn't be done before.

5 Conclusions and Future Work

In this paper, we have discussed the applications of ANN and deep learning methods on the field of network traffic identification. We use the real data that are collected from our enterprise network. The results show that our approach works very well on the applications of feature learning, protocol classification, anomalous protocol detection and unknown protocol identification. Our approach solves the problem of non-automation and poor adaptation in traditional ways. At the deadline of this paper, we are studying the analysis of encrypted traffics. We plan to focus on the identification of P2P protocol.

References

[1] Touch J, Kojo M, Lear E, et al. Service Name and Transport Protocol Port Number Registry[J]. The Internet Assigned Numbers Authority (IANA), 2013.

[2] Park B C, Won Y J, Kim M S, et al. Towards automated application signature generation for traffic identification[C]//Network Operations and Management Symposium, 2008. NOMS 2008. IEEE. IEEE, 2008: 160-167.

[3] Kang H J, Kim M S, Hong J W K. A method on multimedia service traffic monitoring and analysis[M]//Self-Managing Distributed Systems. Springer Berlin Heidelberg, 2003: 93-105.

[4] Van Der Merwe J, Caceres R, Chu Y, et al. Mmdump: A tool for monitoring Internet multimedia traffic[J]. ACM SIGCOMM Computer Communication Review, 2000, 30(5): 48-59.

[5] Sen S, Spatscheck O, Wang D. Accurate, scalable in-network identification of p2p traffic using application signatures[C]//Proceedings of the 13th international conference on World Wide Web. ACM, 2004: 512-521.

[6] Zuev D, Moore A W. Traffic classification using a statistical approach[M]//Passive and Active Network Measurement. Springer Berlin Heidelberg, 2005: 321-324.

[7] Zander S, Nguyen T, Armitage G. Self-learning IP traffic classification based on statistical flow characteristics[M]//Passive and Active Network Measurement. Springer Berlin Heidelberg, 2005: 325-328.

[8] Sen S, Wang J. Analyzing peer-to-peer traffic across large networks[J]. IEEE/ACM Transactions on Networking (ToN), 2004, 12(2): 219-232.

[9] Tan K M C, Collie B S. Detection and classification of TCP/IP network services[C]//Computer Security Applications Conference, 1997. Proceedings., 13th Annual. IEEE, 1997: 99-107.

[10] Early J P, Brodley C E, Rosenberg C. Behavioral authentication of server flows[C]//Computer Security Applications Conference, 2003. Proceedings. 19th Annual. IEEE, 2003: 46-55.

[11] Williams N, Zander S, Armitage G. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification[J]. ACM SIGCOMM Computer Communication Review, 2006, 36(5): 5-16.

[12] Wikipedia: Artificial neural network http://en.wikipedia.org/wiki/Artificial_neural_network

[13] Wikipedia: Deep learning http://en.wikipedia.org/wiki/Deep_learning

[14] Vincent P, Larochelle H, Lajoie I, et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion[J]. The Journal of Machine Learning Research, 2010, 11: 3371-3408.

[15] Le Q V. Building high-level features using large scale unsupervised learning[C]//Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013: 8595-8598.

[16] Zeiler M D, Ranzato M, Monga R, et al. On rectified linear units for speech processing[C]//Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013: 3517-3521.

[17] Socher R, Pennington J, Huang E H, et al. Semi-supervised recursive autoencoders for predicting sentiment distributions[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011: 151-161.

[18] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural

networks[C]//Advances in neural information processing systems. 2012: 1097-1105.