A Research on the Identification of Internet User Based on Deep Learning

Hong Shao, Liujun Tang, Ligang Dong(✉), Long Chen, Xian Jiang, and Weiming Wang

School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China
1564027103@qq.com, tlj2016@126.com, {donglg, jiangxian, wmwang}@zjgsu.edu.cn, smllchuju@163.com

Abstract. In the environment of big data, analyzing internet user behavior has become a research hot spot. By profiling the normal online behavior data of network users to learn their online habits and preferences, is not only helpful to provide network users with more efficient and personalized network services, but also to update the network security policies. Because there is no identification of network users in network management, network administrators need to develop and deliver relevant network services manually to user base on the network user Internet Protocol (IP) address. Therefore, this paper proposes the utilization of deep learning technology to identify network user automatically after fully understand the behavior of network user. At the first, a network identification model based on Deep Belief Network (DBN) is proposed. Then, we apply the Tensorflow framework to construct a DBN model suitable for network user identification. Finally, an experiment with real data set was undertaken upon the model to verify its accuracy on identifying network users. It is found that DBN-based identification model can achieve a high classification accuracy of user identity by constructing deep network structure.

Keywords: Deep learning · Deep belief network · User behavior profile

1 Introduction

Today, in the environment big data, profiling network users' behavior has attracted many research organizations and network security researchers. By analyzing the traffic characteristics of network users, it is possible to understand their online behavioral habits and preferences, so as to provide the network users with more efficient and personalized network services and bring about a better using experience. Meanwhile, it can also provide a basis for updating network security policies.

This work was supported by a grant from the Key Research and Development Program of Zhejiang (No. 2017C03058), Zhejiang Provincial Key Laboratory of New Network Standards and Technologies (NNST) (No.2013E10012).

L. Meng and Y. Zhang (Eds.): MLICOM 2018, LNICST 251, pp. 73–80, 2018.
https://doi.org/10.1007/978-3-030-00557-3_8

At present, the research on behavioral profiling of internet users mainly focuses on two directions [13]: abnormally behavior analysis and normal behavioral preference profile. Abnormal behavioral analysis of network users is to find the data information that does not within "normal behavioral pattern" scale. Accordingly, we can achieve the goal to maintain network security and prevent potential threats. Another research direction–analysis of network user behavioral preference has been widely developed on the internet. The purpose of this analysis is mainly to provide users with accurate marketing service and then produce network service [2] of high quality.

Most of the research on network users' behavior is to make required statistic or predictive analysis upon their normal online behavioral data, except of making automatic identification of network users. When developing and issuing network services strategy, they need to do this manually according to the network user's IP address. However, the user's IP address may change dynamically, which requires network administrators to identify artificially. This obviously increases the workload of network administrators. Besides, excessive human intervention will increase the probability of making mistake in managing network. Therefore, this paper proposes the applying of deep learning technique to identify network users automatically.

The remainder of the paper is structured as follows: Sect. 2 reviews the related research on network behavioral profiling. Section 3 explains the network identification model based on Deep Belief Network (DBN) [12], including its training phase and classification phase. The details about how to use the Tensorflow framework [6] to build the DBN-based model, how to construct data set and how to determine model parameters are presented in Sect. 4, followed by an experiment in Sect. 5-examining the classification accuracy of the model with test dataset. The last section is the conclusions of this article.

2 Related Research

At present, the research on behavioral profile of internet users mainly focuses on two fields: abnormal behavior and normal behavioral preferences. The common method of profiling network users' behavior is using cluster to profile. Celebi [4] improved the performance of the k-means clustering algorithm by applying windowing techniques to the clustering process. Tan [11] introduced an implicit semi-Markov model into a piece-wise k-means algorithm to train the algorithm. Ayeldeen [3] utilized vocabulary-similarity based k-means algorithm to improve the accuracy of estimation of the similarity. Ruijuan [10] proposed a user abnormal behavior profiling approach based on neural network clustering to solve the over-fitting and flooding of feature information. Guan [5] proposed and implemented to profile user behavioral preferences based on Hadoop distribution platform. Researchers [9] proposed a personalized service pattern of library that can capture readers' characteristics accurately. This pattern can provide readers with efficient and economical personalized service and have a high user satisfaction. Ma and other researchers [1] proposed an improved BP [8] neural network algorithm based on artificial bee colony algorithm, which can improve the efficiency and accuracy of profiling different users' behavior effectively.

This paper proposes for the first time to apply deep learning technology to learn the underlying relationships among user's behavior characteristics, so as to achieve a high identification precision of network user. Whilst it may provide a reference for further study about applying deep learning technology to network users' behavior analysis.

3 Construction of Users' Identification Model

This section bases on the TensorFlow framework to build a DBN-based model for network users identification. Its structure is divided into four parts: data collection, data pre-processing, determination of the model-parameters and training of the DBN-based model. First, construct an initialized DBN-based model with the TensorFlow. Second, we determine the number of hidden nodes and hidden layers of the DBN-based model through experiments. Then use the training dataset to train the DBN-based model, including unsupervised training [7] and supervised training [8]. After training, we obtain the DBN-based model with better weight parameters. Finally, utilize the test dataset to evaluate the classified effect of the DBN-based model. The input of the model is real network traffic data of a college. The output of the model are three categories, representing teachers, postgraduates and undergraduates respectively.

3.1 Data Collection

In this experiment, the traffic data from college's user generated over a seven-day period was used as the dataset utilized by the network user identification model based on deep learning. In order to profile the behavioral characteristics of network users better, we will use sFlow art to collect network traffic data. The dataset is then divided into seven sub-sets (i.e., the Dataset1–Dataset7); each data set contains 50000 streams.

3.2 Data Pre-processing

Since the network traffic data may present problems in format, information integrity and so on, the raw network traffic information we obtained can't be used directly as the input for the research model. So firstly, this experiment processes the packet by cleaning data noise, filling missing value and operating redundant data; These datasets contain features and tags. Eigenvalues of high quality are the premise of model classification experiments; the quality of feature values will directly affect the classification effect of the model. When examining the identification performance of the model, we need to compare the output of the model with the tags of the test dataset. Therefore, we have to normalize the feature sets and carry on one-hot encoding upon the tag sets to make the dataset suitable for the input of the model. Then use the processed data-Dataset1–Dataset2 as the unsupervised training data set (without labels) for model; select a single Dataset3 set of data (including tag) for supervised fine-tune phase; Dataset4–Dataset7 are used as the test data set (with label) for examining model performance.

Features of the datasets are as those extracted in reference [3] of its network user behavioral profiling. See features detail in Table 1.

Table 1. Features of experimental dataset

Time stamp	Source IP address	Destination IP address
Agreement type	Flow size	

We select 3 types of identities: teachers, graduates and undergraduates as the data label; we collect 15 IPs previously, 5 IPs of each identity-type. The specific data is as the Table 2 below.

Table 2. Experimental tag

Teacher		Postgraduate		Undergraduate	
Label	IP	Label	IP	Label	IP
1	10.20.0.161	6	10.20.216.158	11	10.20.3.168
2	10.20.0.164	7	10.20.216.21	12	10.20.3.172
3	10.20.0.200	8	10.20.216.77	13	10.20.3.173
4	10.20.216.31	9	10.20.216.67	14	10.20.3.174
5	10.20.1.42	10	10.20.216.78	15	10.20.3.184

3.3 Determination of the Model Parameters

Before the unsupervised training of DBN-based model, we need to determine the number of hidden nodes and hidden layers. The number of hidden layer nodes will affect the model's effect on the abstract expression of features; the number of hidden layers will be directly related to the depth of the DBN-based model. And the increase in the number of hidden layers is conducive to a more comprehensive study of the characteristics. Since the problem of taking the number of different nodes and different hidden layers into account together is complicated, this section will firstly study the determination of the number of suitable hidden nodes when the model contains two hidden layers. Then, when the number of hidden nodes is appropriate, we study the appropriate number of hidden layers.

Determination of the Number of Hidden Nodes

At the first step, we can obtain its value in a range with reference to empirical formula (e.g., formula 1) that conventional neural networks utilized for determining the number of hidden nodes of each layer. Through several times of experimental comparison about the classification effect among models which vary in the number of hidden nodes, we can eventually acquire the number of hidden nodes corresponding to the model with the best classification performance; put it as the numerical value for later experiment.

We select the DBN-based model only containing 2 hidden layers as initial model. After calculating with the formula 1 we get the range of n, [1, 14]; we set numerical values of n as consecutive integer in the range of 1 to 14. In order to perform fully comparable experiment, we selected another 6 points-16, 20, 21, 22, 26, 30.

$$n = \sqrt{m + p} + a \tag{1}$$

In the equation above, "m" represents the number of input feature items; "p" stands for the number of output label items; "n" is the number of hidden layer nodes, and a symbolizes an integer within [1, 10].

When there is difference in the number of different hidden layer nodes, the overall classification accuracy of the model will be variant; the result is shown in Fig. 1 below. The sum of the unsupervised and supervised training time of the model is used as the total training time of the model. The total training time of the models containing different number of hidden nodes is compared and shown in Fig. 2 below.

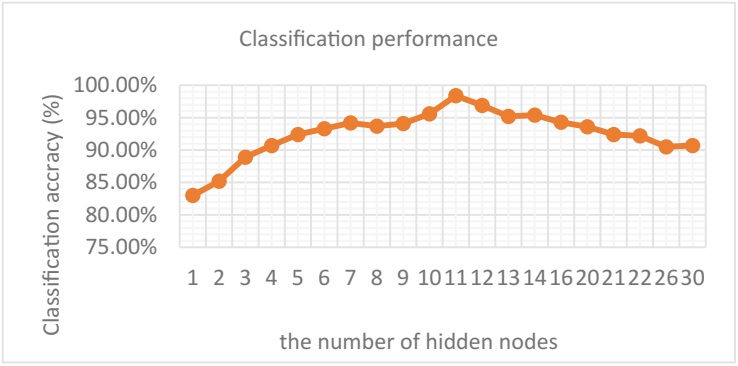

Fig. 1. The classification performance of the models with different number of hidden nodes

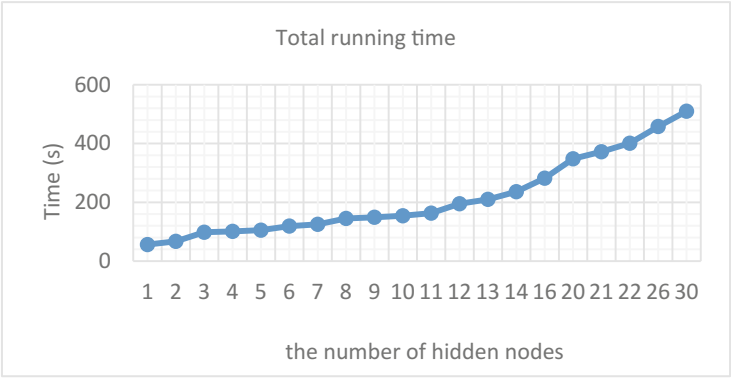

Fig. 2. Total running time of the models corresponding to the different number of hidden nodes

As what's shown in Fig. 1, in the range of [1, 7], the classification accuracy rate of the model grows rapidly, and then slows down. When n = 11, the classification accuracy of the model reached a maximum of about 98.4%.

On the basis of the result in Fig. 2, we may come to the conclusion that with the growth in the number of hidden nodes, the total run-time of the models is increasing. Especially when n > 11, its total run-time would be in a faster growing trend. Due to this, we set the number of hidden layer nodes applied to the DBN-based model to 11 (i.e., n = 11).

Determination of the Number of Hidden Layers

The number of hidden layers will directly affect the depth of the DBN model. The researchers [10] proved that the accuracy of the classification can be further improved by increasing the number of hidden layers of the DBN model to improve the abstraction ability of the data features. But not the more the number of hidden layers is, the better the classification effect will be.

According to the analysis before, the number of hidden nodes of each layer is set to 11. We will train the model respectively with different number of hidden layers, from 1 to 6. By having train on DBN-based models containing different number of hidden layers, we can obtain a group of corresponding DBN-based models. Then use the test dataset-Dataset 4–Dataset7 (Including tags) to examine those DBN-based models respectively; analyze the classification accuracy and total running time of different models so as to select the proper number of hidden layers corresponding to the model that achieves better classification effect as the final number of hidden layers for the DBN-based model. Table 3 shows the details about the classification accuracy and total running time of different numbers of hidden layers.

It can be seen from the table above that the total running time of every DBN-based models with 11 nodes of each hidden layer increases as the number of hidden layers grows. However, the classification accuracy of models increases first and then decreases with the increase of the number of hidden layers. When the number of hidden layers is n = 4, the classification accuracy of the model reaches the highest value of about 98%. And when n = 4, the running time of model is about 502 s. The time consumption is not big. Therefore, this chapter determines to build a DBN-based model with 4 hidden layers.

Table 3. Performance of the models

The number of hidden layers	Classification accuracy	Running time (s)
1	93.2%	201.1
2	95.1%	308.7
3	96.4%	446.5
4	98.1%	502.2
5	96.7%	593.7
6	94.6%	742.2

As the demonstration above, through the analysis of many experimental results, this paper finally determines to train the DBN-based model with 4 hidden layers and 11 hidden nodes of each hidden layer.

3.4 Training Phase of the Model

Process training on the constructed DBN-based model. The first step is to carry on unsupervised training with unlabeled-datasets, constructing a model initially. Then utilize Dataset3 (including tag) to adjust the model after being processed unsupervised training. That is, the weights trained in the unsupervised phase are transferred to the BP neural network for reverse fine-tuning of the DBN-based model. As a result, a fully trained DBN-based model is obtained which means we have constructed a complete identification model based on DBN.

4 Performance Examination of the Model

4.1 Experimental Environment

The experimental platform for this paper is an Intel Core i5 processor with a 3.3 GHz, 14.0 GB memory on a HP computer running window 10 (64 bit) operating system. This paper uses the TensorFlow framework to build the DBN-based model, in which all DBN-based algorithms are implemented in Python language.

Specific version of the software tools used herein is Tensorflow1.2.1, Python3.5.1.

4.2 Experimental Data and Results

In this experiment, we collect the data according to the method described in Sect. 3.1, and then pre-process the collected data according to the method in Sect. 3.2 of this paper. Thereafter, 4 labeled datasets-Dataset4–Dataset7 are utilized as the test dataset to process validation on previously trained DBN-based model. Figure 3 presents the results about classification accuracy of the model we come to.

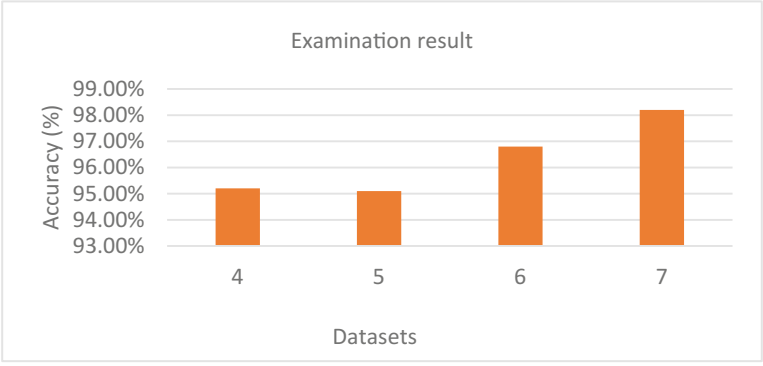

Fig. 3. Examination result

From the Fig. 3 above it may be seen, for different sets of data, the classification precision of the same model is various. This is the result of the difference in distribution of network user categories each dataset contained. It can be learned from the figure that

the DBN-based model has a classification accuracy of over 95% for these four data sets and a maximum of 98%. Experimental results show that DBN-based model has a high accuracy for network user identification.

5 Conclusions

Aiming at the deficiencies of prior art in network users' identification, combining with the current prevalence of deep learning, this paper presents a study on DBN-based network users' identification. Deep learning technology is applied to identify network users and to find the underlying relationships among the behavioral characteristics of network users. Thus, improve the accuracy of network user identification. And utilizes the real traffic data of a college as the research object. The final experimental results show that the DBN-based model can achieve a high classification accuracy of network users.

References

1. Ma, J., Zhou, G., Xu, B., et al.: A microblog user impact analysis method based on the diffusion of topic. Univ. Inf. Eng. **14**(6), 735–742 (2013)
2. Zhou, J.: Network User Behavior Analysis for SDN Firewall. Zhejiang Gongshang University (2017)
3. Ayeldeen, H., Hassanien, A.E., Fahmy, A.A.: Lexical similarity using fuzzy Euclidean distance. In: 2014 International Conference on Engineering and Technology (ICET), pp. 1–6. IEEE, (2014)
4. Celebi, M.E., Kingravi, H.A., Vela, P.A.A.: A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst. Appl. **40**(1), 200–210 (2013)
5. Guan, J., Yao, S., Xu, C., Zhang, H.: Design and implementation of network user behaviors analysis based on hadoop for big data. In: Batten, L., Li, G., Niu, W., Warren, M. (eds.) ATIS 2014. CCIS, vol. 490, pp. 44–55. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45670-5_5
6. Huang, S., Chen, K., Liu, C., et al.: A statistical-feature-based approach to internet traffic classification using machine learning. In: International Conference on Ultra Modern Telecommunications & Workshops, pp 1–6. IEEE (2009)
7. Le, Q.V.: Building high-level features using large scale unsupervised learning. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8595–8598. IEEE (2013)
8. Oravec, M., Podhradsky, P.: Medical image compression by backpropagation neural network and discrete orthogonal transforms. WIT Trans. Biomed. Heal. **4** (1970)
9. Paik, J.H.: A novel TF-IDF weighting scheme for effective ranking, pp. 343–352 (2013)
10. Ruijuan, Z., Jing, C., Mingchuan, Z., et al.: User abnormal behavior analysis based on neural network clustering. J. China Univ. Posts Telecommun. **23**(3), 29–44 (2016)
11. Tan, X., Xi, H.: Hidden semi-Markov model for anomaly detection. Appl. Math. Comput. **205**(2), 562–567 (2008)
12. Hinton, G., Osindero, S.: A fast learning algorithm for deep belief nets. Neural Comput. **18** (7), 1527–1554 (2006)
13. Dong, F.: Studies and utilization on network users' behavior analysis. Xidian University (2005)