International Journal of Grid Distribution Computing
Vol. 8, No. 3, (2015), pp. 309-322
http://dx.doi.org/10.14257/ijgdc.2015.8.3.29

A New Network Traffic Classification Method Based on Classifier Integration

Zhang Luoshi[1], Xue Yibo [2, 3] and Bao Yuanyuan[2]

[1] *School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China*
[2] *Research Inst. of Info. & Tech., Tsinghua University, Beijing 100084, China*
[3]*Tsinghua National Lab for Information Sci. & Tech., Beijing 100084, China*
luoshi.zh@gmail.com, yiboxue@tsinghua.edu.cn, by51800@163.com

Abstract

With development of scale, diversity and complexity of network traffic, the drawbacks of traditional machine learning methods on traffic classification is gradually exposed, especially the false positive problem in large-scale real network traffic classification is particularly serious. In this paper, aiming at reducing the false positive rate of network traffic classification, an effective network traffic classification method --- CMM method. CMM method contains three steps, including dividing the training set into clusters, forming sub-classifiers, and classifier integration in accordance with the principle of minimization and maximization. In this paper, we firstly demonstrate the effectiveness of this method in reducing the false positive rate. Secondly, we conduct experiments in large-scale national backbone network, such as the SSL protocol classification and experimental results verify the effectiveness of this method in large-scale the actual network traffic classification.

Keywords: *Traffic Classification, Precision, False Positive, Classifier Integration, Machine Learning*

1. Introduction

Network traffic classification is the premise and basis for network management, optimization and security. Accurate network traffic classification can help rationally allocate network resources, optimize network scheduling, enhance network security and prevent loss of private data. This is a prerequisite for the healthy, safe and reliable operation of network. However, with the rapid development of network technology and the continuous expansion of the network size, the traditional network traffic classification technology has faced major challenges. On the one hand, the widespread use of random port technology has led to the failure of network traffic classification methods based on fixed port. At present, they can only identify less than 30% of network traffic [1]. On the other hand, in consideration of data security and privacy protection, encryption technology has been widely used. It has resulted in failure of network traffic classification methods based on payload. In addition, the continuous use of fuzzy technology such as protocol multiplexing and traffic obfuscation has further reduced the identification ability of traditional network traffic classification technology.

To effectively solve the new problems of network traffic classification, machine learning methods have been introduced in this field. Compared to the traditional ones, the network traffic classification methods based on machine learning take the network flow as the classification unit to extract the information at packet-level (such as packet payload length and arrival interval between packets) and flow-level

ISSN: 2005-4262 IJGDC

(such as the number of all packets and the average payload length of packets) as the classification feature and use Naive Bayes, SVM, C4.5 and other machine learning algorithms for network traffic classification. Such methods do not depend on the contents of the payload of packet or the port number. They can solve the current problems of network traffic classification such as port multiplexing and traffic encryption and have become a research focus.

However, most existing network traffic classification methods based on machine learning have a premise that the flow number of various protocol in the training set is basically balanced and the classifier trained accordingly has better classification effect on the test data with balanced class in the laboratory environment. But in the real-world network environment, various class of network traffic are of great difference in the proportion, that is, some class of protocol are far fewer than others, with a serious imbalance. The class with fewer samples are called little class or rare class in this paper, while those with a large number of samples are called large class. In the above-mentioned real-world network environment, if the model trained by the data of balanced class was used to classify the network traffic of little class, a lot of large class will be mistakenly identified as little class. That is to say, a serious problem of false positives will occur and greatly affect the accuracy of traffic classification. Therefore, in the context of massive and complex real network traffic, how to design a high-accuracy method to classify the traffic of little class has great academic significance and high practical value.

This paper analyzed the causes of low traffic classification accuracy of little class in the large-scale network environment, designed a new network traffic classification method to effectively increase the classification accuracy and experimentally validated it in the actual network traffic classification. The experimental results validated the effectiveness of this method in the classification of actual network traffic of little class.

The main contributions of this paper are as follows:

- It analyzed the main causes of the decrease in classification accuracy of traffic of little class based on traditional machine learning methods;
- It designed a network traffic classification method based on classifier ensemble that could effectively improve the classification accuracy of traffic of little class – CMM (Cluster-Min-Max);
- It experimentally validated CMM method in the context of actual network traffic, and the experimental results verified the effectiveness of CMM method in the classification of massive network traffic.

This paper was organized as follows: Section 2 described the traditional network traffic classification technology and the research status of that based on machine learning; Section 3 analyzed the reasons for the decrease in accuracy and proposed CMM method, a classification method of large-scale network traffic; Section 4 experimentally validated the accuracy, recall and false positive of CMM method in the context of actual network traffic; Section 5 summarized and discussed the next research work.

2. Related Works

Traditional network traffic classification technologies mainly include port-based ones and payload-based ones. The port-based network traffic classification technology is based on the port number assigned by IANA for each network application to classify the traffic, such as port 443 assigned to HTTPS protocol. However, with the widespread use of the technology such as random port, port multiplexing and port hopping, the existing port-based network traffic classification

technology can only identify less than 30% of network traffic [1], with the identification accuracy of only 50-70% [2].

Payload-based network traffic classification technology builds signature through the pre-analysis of packet payload characteristics of network application, uses regular expressions or string matching to determine the presence of such signature in the network traffic, and based on this determines the protocol of network flow. However, with the extensive use of encryption protocol, protocol multiplexing and feature obfuscation, the payload characteristics are hidden so that payload-based network traffic classification technology has gradually lost its effect [3].

In order to effectively address the failures of port-based and payload-based network traffic classification technologies, machine learning methods have been introduced in the field. In 2003, Early *et al.* [4] used the statistical features of network flow such as average packet payload length, average interval time between packets and TCP header flag to distinguish the applications of HTTP, SMTP, FTP, SSH and TELNET. Since then, network traffic classification methods based on machine learning have gradually become a research focus, and a large number of research results have been achieved [5].

To further improve the accuracy of network traffic classification methods based on machine learning, Moor *et al.* summarized 248 class of features in the network flow that could be used for machine learning methods [6]. Williams *et al.* made an effective evaluation on the performance and accuracy of five different machine learning algorithms in terms of network traffic classification [7].

Yang Baohua *et al.* [8] proposed SMILER and used the semi-supervised machine learning algorithm as well as the payload length of the first 5 packets to classify network traffic and achieved high accuracy. Xi Liang et.al analyzed research status of immunity-based intrusion detection system (IIDS) and promoted the conversion of the theoretical fruits to applications and stimulated the deeper developments of artificial immune systems [9].

Based on active learning and SVM algorithms, Wang Yipeng *et al.* [10] achieved the classification of unknown network protocol traffic only depending on the payload information in untreated network traffic. They reduced the number of learning samples and at the same time ensured the classification accuracy. Dong Hui *et al.* proposed a new method based on link homophily to classify traffic in application layer without the payloads and properties, and achieved above 80% accuracy [11].

However, the existing classification algorithms mostly consider the balanced data set, that is, the same number of sample data is obtained from each protocol class to constitute the training set. Meanwhile, for the collection of training samples, only the random under-sampling method can be used to obtain partial traffic due to environment and time constraints. Therefore, flow features cannot be all effectively covered. As put into the real-world network environment, they show a sharp decline in their identification effect, have a high misclassification rate and take on significant false positives. To this end, Zhang Hongli *et al.* [12] compared the performance of C4.5, SVM, NBK and Bagging algorithms for the classification of network traffic with imbalanced class. Although Bagging algorithm and C4.5 decision tree algorithm have relatively better performance in handling little class of network protocol, they still cannot solve the problem of imbalanced class. Nguyen [13] and Dainotti [14] *et al.* also argued that the existing network traffic classification technology based on machine learning mainly faced serious challenges of imbalanced class and had poor classification effect on little class and almost invalid usability[15-17].

3. CMM Method

3.1. Description of Problem

What is researched in this paper is the classification of little class of network traffic in the real network environment. The network traffic of little class to be identified was called positive example traffic or positive example set. The remaining traffic of other class was called negative example traffic or negative example set.

There are extremely imbalanced between positive example traffic (traffic of little class) and negative example traffic (traffic of other class) in the actual network traffic. As required by traditional machine learning methods, balanced positive example traffic and negative example traffic are taken as the training sample set, and the classifier obtained from such training sample set is applied to the actual network traffic classification. As a result, a lot of negative examples will be misidentified as positive examples so that a serious problem of false positives will occur. False positives mean that the negative example data is misidentified as the positive example, also known as Class I errors. False positives can be clearly represented by hybrid matrix, as shown in Table 1.

Table 1. Confusion Matrix of the Classifier

	Classified as positive	Classified as negative
Positive	TP	FN
Negative	FP	TN

Seen from Table 1, false positives indicate the number of negative example data misidentified as positive example, which can be represented by FP. False positives may cause a decrease in classification accuracy. The precision P of the classifier is the ratio of those correctly classified as positive examples and those identified as positive examples. See Formula 1.

$$P = \frac{TP}{FP + TP} \tag{1}$$

Because the number of flow in large class are far more than little class, the classification accuracy of the little-class traffic is greatly affected by false positives. In addition, the problem of false positives tends to be perceived by users. Symantec's mistake in 2007 [18] indicated the serious impact of false positives. Therefore, the goal of classification in this paper is to design a network traffic classifier that is applicable to the real network environment and has high accuracy so as to reduce the probability of false positives and improve the classification accuracy.

3.2. Design Philosophy

The data set that had consistent distribution with the traffic dataset in the actual network environment was selected as the training set, namely imbalanced positive and negative examples. This paper proposed an effective network traffic classification method in the real-world network environment – CMM. Its design idea is as follows.

Suppose that Xi represents a flow, Y represents protocol class and can have a value of 1 (positive example traffic) or -1 (negative example traffic). This paper is to obtain a classification function f through training set learning so as to correctly classify positive example traffic and negative example traffic. Since this paper

focuses on the problem of low accuracy caused by false positives, it aims to generate a classifier f with a low probability of false positives, i.e. the minimum probability $P[f(X)=1 \wedge Y=-1]$. This probability can be further represented as shown in Formula 2.

$$\begin{aligned} & P[f(X)=1 \wedge Y=-1] \\ &= P[f(X)=1]-P[f(X)=1 \wedge Y=1] \\ &= P[f(X)=1]-P[Y=1]+P[f(X)=1 \mid Y=1]P[Y=1] \end{aligned} \tag{2}$$

Because $P[Y=1]$ is a constant, Formula 3 can be minimized so as to minimize the probability of false positives.

$$P[f(X)=1]+P[f(X)=1 \mid Y=1]P[Y=1] \tag{3}$$

If $P[f(X)=1 \mid Y=1]$ can be reduced to very small, then the learning process is the same as to minimize $P[f(X)=1]$. $P[f(X)=1 \mid Y=1]$ is the recall ratio of positive example traffic, which is generally also an important measure, $P[f(X)=1 \mid Y=1]$ cannot be infinitely small. Given that $P[f(X)=1 \mid Y=1]$ is required $\geq\beta$, the minimization of Formula 3 is the same as that of $P[f(X)=1]$ when $P[f(X)=1 \mid Y=1]$ is maintained $\geq\beta$. Therefore, to minimize the probability of false positives $P[f(X)=1]$ in the case that the recall ratio of positive example traffic is not less than β is to minimize the number of positive example traffic in the data set in the same case.

Based on the above analysis, the minimization of probability of false positives is converted to that of the number of positive example traffic in the case that the recall ratio of positive example traffic is maintained not less than β. Based on this goal, the strict standard was used to determine positive examples. This goal was ensured through the division between positive example set and negative example set in the training set as well as the repeated validation of the negative example subset to the positive example subset. In this paper, the method included the following two steps:

(1) Division of training set and generation of sub-classifier: the positive examples set and the negative example set in the training set were clustered respectively to form m positive example sets P_1, P_2, ..., P_m, and n negative example sets N_1, N_2, ..., N_n; m positive example sets and n negative example sets were intersected as the positive example sets and negative example sets in m×n sub-training sets, in which m×n classifiers were trained and generated;

(2) Classifier integration: in order to minimize the number of positive example traffic, as all negative example subsets were used to validate positive example subsets, the minimum value of all classification results was obtained according to the minimization principle; at the same time, because all positive example subsets were not intersected, as the traffic was finally determined as positive example traffic, the maximum value of all classification results was obtained according to the maximization principle to achieve the sub-classifier integration to get the final classification result.

3.3. Division of Training Set and Generation of Sub-Classifier

If the original problem is that the traffic of one protocol is identified in the environment of large-scale traffic and this particular class is c, then the traffic of such class is called positive example traffic and the rest of traffic of other class is collectively referred to as negative example traffic. The classification problem of such traffic is a binary classification problem. In order to minimize the number of

positive example traffic in the data set in the case of an ensured recall ratio of positive example traffic, the clustering method was first used to divide the positive example set and the negative example set in the training set. T was used to represent the training sample set of this binary classification problem, as shown in (4).

$$T = \{(X_l, y_l)\}_{l=1}^{L} \tag{4}$$

Where L is the number of training samples, X_l is the input vector and y_l is the class variable, as shown in (5).

$$y_l = \begin{cases} 1 & X_l \text{ belong to c} \\ -1 & X_l \text{ not belong to c} \end{cases} \tag{5}$$

Training sample set T can also be expressed by (6).

$$T = \{(X_l^{(1)}, 1)\}_{l=1}^{L_1} \cup \{(X_l^{(-1)}, -1)\}_{l=1}^{L_2} \tag{6}$$

Where $X_l^{(1)}$、$X_l^{(-1)}$ respectively represent the input variable that belongs to class c and that does not belong to class c, and χ_1 and χ_{-1} respectively represent the set of input variables of the two class, as shown in (7).

$$\chi_1 = \{X_l^{(1)}\}_{l=1}^{L_1}, \quad \chi_{-1} = \{X_l^{(-1)}\}_{l=1}^{L_2} \tag{7}$$

With the clustering method, suppose that χ_1 can be divided into $N_1 (1 \le N_1 \le L_1)$ subsets, the form is shown in (8).

$$\chi_{1,j} = \{X_l^{(1,j)}\}_{l=1}^{L_1^{(j)}}, \ j = 1, 2, \ldots, N_1 \tag{8}$$

Where $\chi_{1,j}$ represents subset j in the traffic of class c, and $L_2^{(j)}$ represents the number of samples in subset $\chi_{1,j}$. Similarly, suppose that χ_{-1} can be divided into $N_2 (1 \le N_2 \le L_2)$ subsets, the form is shown in (9).

$$\chi_{-1,j} = \{X_l^{(-1,j)}\}_{l=1}^{L_2^{(j)}}, \ j = 1, 2, \ldots, N_2 \tag{9}$$

Where $\chi_{-1,j}$ represents subset j in the traffic of other class, and $L_2^{(j)}$ represents the number of samples in subset $\chi_{-1,j}$. On the basis of the divided training sample set, the original binary classification problem can be decomposed into $L_1 \times L_2$ sub-binary classification problems. The training sample set of each sub-binary classification problem can be expressed as (10).

$$\begin{aligned} &T_{ij} = \{(X_l^{(1,i)}, 1)\}_{l=1}^{L_1^{(i)}} \cup \{(X_l^{(-1,j)}, -1)\}_{l=1}^{L_2^{(j)}} \\ &i = 1, 2, \ldots, N_1, \ j = 1, 2, \ldots, N_2 \end{aligned} \tag{10}$$

Where $X_l^{(1,i)} \in \chi_{1,i}, X_l^{(-1,j)} \in \chi_{-1,j}$ respectively represent the input variable that belongs to class c and that does not belong to class c, and $\sum_{i=1}^{N_1} L_1^{(i)} = L_1, \sum_{j=1}^{N_2} L_2^{(j)} = L_2$.

$N_1 \times N_2$ Classifiers could be trained and obtained in each training sample subset above. See Figure 1 for the process to divide the training set and generate classifiers. The positive example set was divided into $N_1 (1 \le N_1 \le L_1)$ subsets, and the

negative example set was divided into $N_2 (1 \le N_2 \le L_2)$ subsets. They respectively constituted the training set of sub-binary classification problems, where $N_1 \times N_2$ sub-classifiers were generated, f_{ij}, $i = 1, 2, ..., N_1$, $j = 1, 2, ..., N_2$.

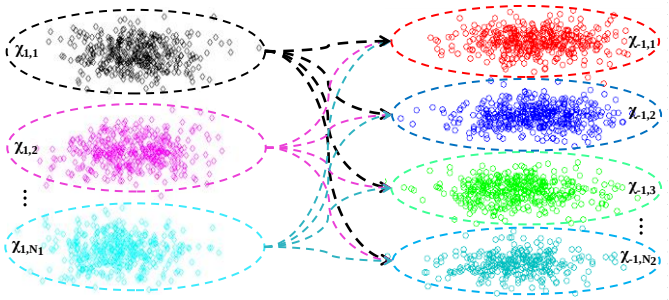


Figure1. The Division of Training Set and Generation of Classifiers

3.4. Classifier Integration

In order to minimize the number of positive example traffic, the positive example set should first be validated by all negative example sets. Therefore, the minimum value of all classification results was obtained in line with the minimization principle.

Theorem 1: (The minimization principle [19]). Suppose that a large-scale binary classification problem is decomposed into P small sub-binary classification problems B_i, i = 1,2, ..., P, and all of them have the same positive example training set and different negative example training sets. If the P sub-binary classification problems have been properly learned, then the minimum value of result of them is the right training result of all training samples in B.

Since any of the positive example sets is not intersected with others, when the traffic is finally determined as positive example traffic, the maximum value of all classification results needs to be obtained in line with the maximization principle.

Theorem 2: (The maximization principle [19]). Suppose that a large-scale binary classification problem is decomposed into P small sub-binary classification problems B_i, i = 1,2, ..., P, and all of them have the same negative example training set and different positive example training sets. If the P sub-binary classification problems have been properly learned, then the maximum value of result of them is the right training result of all training samples in B.

Based on the above analysis, the classifier integration of all sub-binary classifiers should be carried out in line with the minimization principle and the maximization principle. See Figure 2 for the ensemble idea.

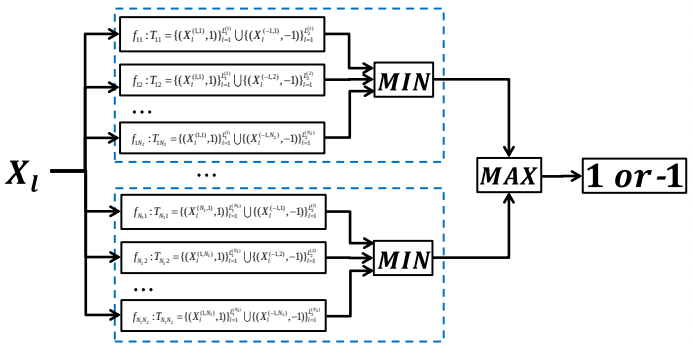


Figure 2. Classifier Integration

For the large-scale binary classification problem, first the positive example training sets and the negative example training sets were divided to form $N_1 \times N_2$ training sample sub-sets, where $N_1 \times N_2$ sub-binary classifiers could be trained and obtained, f_{ij}, $i = 1, 2, ..., N_1$, $j = 1, 2, ..., N_2$, that is, the large-scale binary classification problem was decomposed into $N_1 \times N_2$ sub-binary classification problems. The integration of the N_2-power minimum and the 1-power maximum of $N_1 \times N_2$ sub-binary classifiers f_{ij} was carried out. Given that the result classified by f_{ij} sub-binary classifiers was $output_{ij}$, $output_i$ was the output classification result after processing with the minimization principle, and *output* was the final classification result of the original binary classification problems. $output_i$ could be represented by (11) and *output* by (12).

$$output_i = MIN(output_{i1}, output_{i2}, ..., output_{iN_2}) = \min_{j=1}^{N_2}(output_{ij}) \tag{11}$$

$$output = \max_{i=1}^{N_1}(output_i) \tag{12}$$

Based on the minimization principle and the maximization principle, the N_2-power minimum and the 1-power maximum of $N_1 \times N_2$ sub-binary classifiers f_{ij} were the final classification result of original sub-binary classification problems, that is, the traffic was classified as the positive (output = 1) or the negative (output = -1).

3.5. CMM Traffic Classification Method

CMM traffic classification method consists of classifier training and traffic classification, as shown in Figure 3.

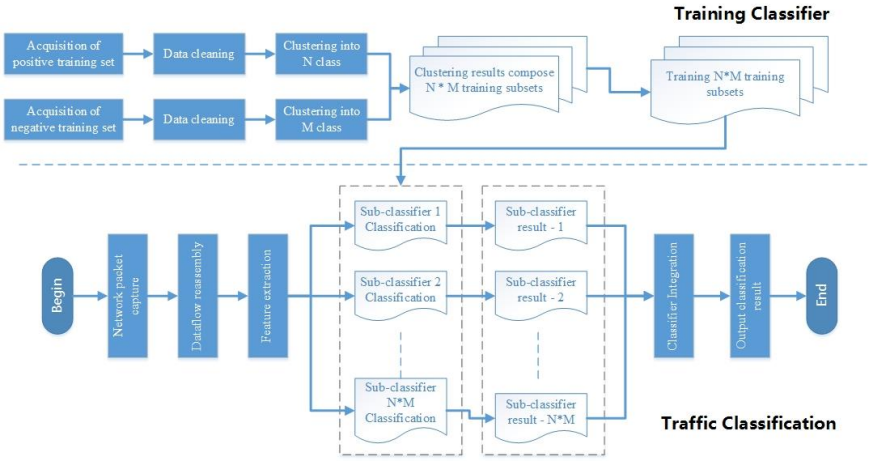


Figure 3. The Processing Flow of CMM Method

The classifier training contains five steps:

(1) Acquisition of training set. In the environment of controlled network and operating system, the marked class of traffic was obtained with artificial repetition as positive examples of the training set. The background traffic excluding positive example traffic was taken as negative examples of the training set.

(2) Data cleaning. In order to reduce the training cost and improve the precision of subsequent clustering, repetitive data sets were removed from the training set.

(3) The clustering algorithm is used to respectively cluster the positive and negative examples of the training set, the positive examples are divided into N subsets and the negative examples are divided into M subsets.

(4) The subsets generated by clustering the positive and negative examples are combined with each other and N*M training subsets are formed.

(5) The classification algorithm is used to train N*M sub-classifiers.

In the process of traffic classification, the network packets are firstly captured in the network, and the network packets with the same five-tuple form a network flow, then features are extracted from the network flow to form the feature vectors, which are used for detection. The feature vectors are classified by N*M sub- classifiers generated in the training process. N*M classification results are obtained and are then integrated by classifiers combination algorithm mentioned above, obtaining the final classification results and the results are finally exported.

4. Experiment and Analysis

4.1. Experimental Data

In order to validate the effect of the proposed CMM method on the classification of little-class traffic in the large-scale network traffic environment, 1-hour network traffic (800G) collected at the gateway of a national backbone network was taken as the experimental subject. In consideration of the feasibility of algorithm performance evaluation, the universal encryption protocol with a fixed port number or significant payload characteristics - SSL protocol (443) was selected as the target protocol of classification, i.e. positive example traffic.

For feature selection, the most commonly used features include the packet length (length of the packet payload with IP header and TCP header removed) and the time interval between two consecutive packets. This paper selected the packet payload length i.e. the payload length of the first 5 consecutive packets in the same network traffic as the classification attribute.

4.2. Measurements and Metrics

The accuracy evaluation of the protocol identification method based on machine learning mainly depended on the following four indicators:

(1) Precision: the proportion of the correct samples in those correctly classified as a class in a given test data set, which is used to measure the model's precision capability, commonly represented with P.

(2) False Positive Rate: negative example data identified as positive examples, also called Class I errors and often represented with FPR.

(3) Recall: the proportion of the correctly classified samples in a particular class of correct samples in a given test data set, which is used to measure the model's recall capability, commonly represented with R.

(4) F-Measure shows the weighted average of precision and recall and comprehensively measures the identification accuracy, as follows:

$$F = \frac{2PR}{P + R} \tag{13}$$

These four metrics effectively evaluate the identification effect of different machine learning algorithms and different features and better show the usability of algorithms.

4.3. Results and Analysis

In order to identify SSL traffic in the 1-hour network traffic at the gateway of the national backbone network, the payload length of the first 5 consecutive packets was taken as traffic classification features. SSL traffic was positive example traffic and the rest was negative example traffic. All traffic was cleaned and filtrated, and the final data set contained a total of 101,196 flows. After matching of significant features, only 407 flows of positive example were obtained from all the data, while the number of negative example flow was 100,789.

To validate the effectiveness of CMM method in the classification of large-scale network traffic, three experiments were used to test the SSL traffic in the above large-scale network traffic. The first experiment used algorithms of J48, Naïve Bayes and Bayes Net to respectively classify SSL traffic in the case of distributed original data. The second selected the negative examples equal to positive examples as negative example traffic to form the training set with all the positive example sets. It used algorithms of J48, Naïve Bayes and Bayes Net to respectively classify SSL traffic. The third used the proposed CMM method for training and SSL traffic classification in the case of distributed original data. Naïve Bayes EM algorithm was used to divide the training set into positive example training set and negative example training set. In sub-binary classification, algorithms of J48, Naïve Bayes and Bayes Net were respectively used for the classifier training of training sample subsets. For classifier integration, the ensemble of classification results was carried out in accordance with the minimization principle and the maximization principle described in Section 3.5 to obtain the classification result of the original network traffic protocol. See Figures 4, 5 and 6 for the results of the above three experiments.

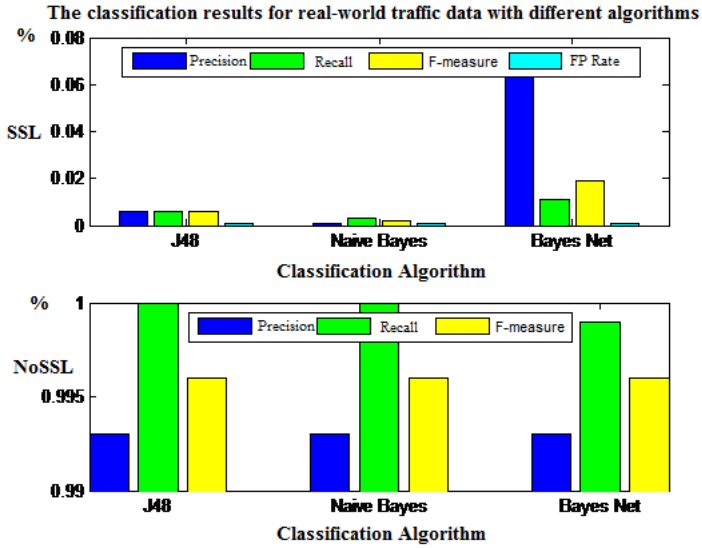


Figure 4. The Classification Results for Real-World Traffic with Different Algorithms

As can be seen from Figure 4, the first experiment had great difference between positive example data and negative example data. As J48, Naïve Bayes, Bayes Net and other traditional machine learning algorithms were directly used in the original (real-world) data set for SSL traffic classification, the accuracy or recall rate of classification was 0. The features of little-class traffic were ignored so that little-class traffic could not be fully identified and F-Measure was 0. In this case, the

classification accuracy of little-class traffic was approximately 0, basically unable to classify the little-class traffic.

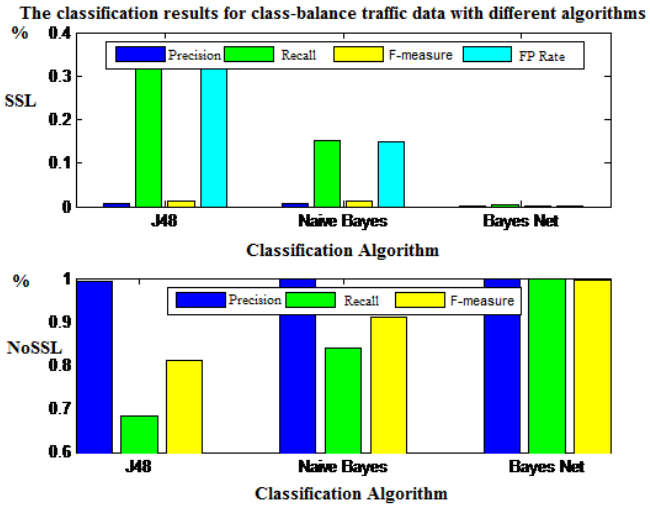


Figure 5. The Classification Results for Class-Balance Traffic with Different Algorithms

As can be seen from Figure 5, the second experiment selected negative examples equal to positive example sets as negative examples traffic to constitute the training set with all the positive example sets and used algorithms of J48, Naïve Bayes and Bayes Net to classify SSL traffic. Because this method achieved a man-made balance between positive and negative examples and highlighted the features of positive example traffic, it increased the recall rate of positive example traffic. But it reduced identification accuracy. A large number of negative examples were identified as positive ones. This resulted in a high false positive rate, up to 0.316. For large class are far more than little class, the number of false positive traffic of the false positive rate 0.316 was great. This led to low classification accuracy of such method. F-Measure value of 0.014 indicated that this method was invalid for SSL traffic classification.

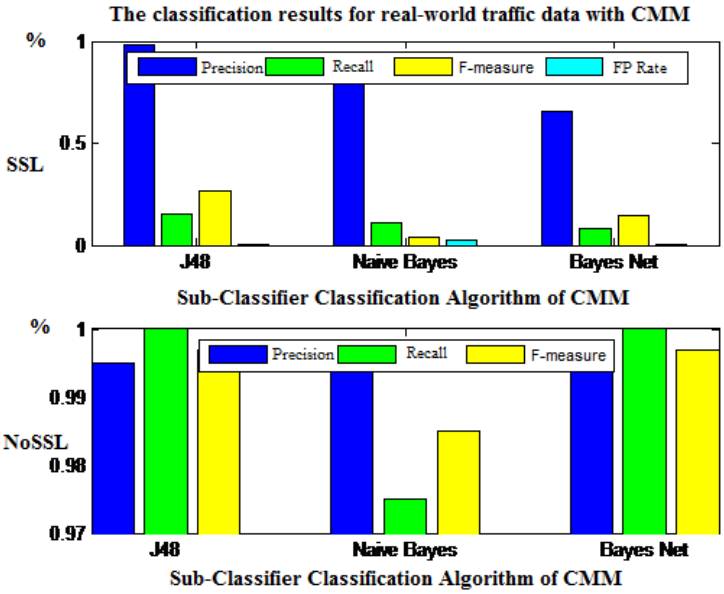


Figure 6. The Classification Results for Real-World Traffic with CMM

The third experiment used the proposed CMM method for SSL traffic classification in the case of distributed original (real-world) data. As can be seen from Figure 6, when the sub-classifier used J48 algorithm, the accuracy of SSL traffic classification was 0.979, the recall rate 0.156, the false positive rate 0 and F-Measure was 0.269. This indicated that the method had much higher effect in the classification of little-class traffic in the large-scale traffic environment than that of the first and the second experiments. To validate CMM method's independence of the sub-classifier's classification algorithm, Naïve Bayes algorithm and Bayes Net algorithm were additionally selected as the classification algorithms of the sub-classifier. See Figure 6 for the classification effect. The algorithms of the sub-classifier had a little impact on the classification effect of CMM method. But they were effective in the classification of little-class traffic. For example, in the above two cases, the classification accuracy was respectively 0.825 and 0.658, higher than the accuracy of the first 2 experiments, and F-Measure was 0.189 and 0.146. These experiments indicated that the proposed CMM method had a greatly improved accuracy in classification of little-class traffic and at the same time had a higher recall rate and higher classification effect.

5. Conclusion

Network traffic classification is the basis for network security and management. However, with the extensive use of the encryption protocol and the Private Protocol, the traditional traffic classification methods appear to be inadequate in the face of new problems. In order to effectively solve the problems, the traffic classification methods based on machine learning have become a research focus and a lot of research results have been achieved. Due to the development trend of large-scale, diverse and complex network traffic, the traditional machine learning methods have gradually exposed their drawbacks in traffic classification. In particular, the problem of false positives was serious in the classification of large-scale network traffic. They have had a sharp decline in accuracy and even failure. Therefore, this paper proposed an effective network traffic classification method - CMM method to address the problem of low accuracy in large-scale network traffic classification. This method consists of division of training set, generation of sub-classifier and sub-classified integration. The effectiveness of this method in terms of accuracy improvement was proved theoretically. And in the experiments, the real network traffic from the national backbone network was used for the classification of SSL protocols. The experimental results validated the effectiveness of this method in the classification of large-scale network traffic.

In the future research, the impact of different clustering methods and classifier integration methods on the proposed method will be further analyzed, and the resampling method will be tried to solve the problem of classification of little-class traffic and promote the usability of machine learning methods in the field of network traffic classification so as to overcome the problem of industrialization of network traffic classification methods based on machine learning.

6. Acknowledgements

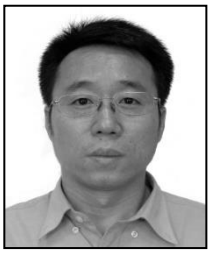
This work was supported by the National Key Technology R&D Program of China under Grant No.2012BAH46B04.

References

[1] A. Madhukar and C. A. Williamson, "longitudinal study of P2P traffic classification", Proceedings of the 14th IEEE International Symposium on Modeling", Analysis, and Simulation, **(2006)** September, pp. 11-14, Monterey, CA, USA.
[2] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications. Proceedings of 6th International Workshop", PAM 2005, **(2005)** March 31 - April 1, Boston, MA, USA.
[3] Q. Liu, Z. Liu, M. Huang, "Study on Internet Traffic Classification Using Machine Learning", Computer Science, **(2010)**, pp. 12, 37
[4] J. Early, C. Brodley and C. Rosenberg, "Behavioral authentication of server flows", Proceedings of the 19th Annual Computer Security Applications Conference, **(2003)** December, pp. 8-12, Las Vegas, NV, USA.
[5] T. Auld, A. W. Moore and S. F. Gull, "Bayesian neural networks for Internet traffic classification", IEEE Trans. Neural Networks, vol. 1, no. 223, **(2007)**.
[6] A. W. Moore, and D. Zuev, "Internet Traffic Classification Using Bayesian Analysis Techniques", Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, **(2005)** June, pp. 6-10. New York, NY, USA.
[7] N. Williams, S. Zander and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification", ACM SIGCOMM Computer Communication Review. Vol. 5, no. 36, **(2006)**.
[8] B. Yang, G. Hou, L. Ruan, Y. Xue and J. Li, "SMILER towards practical online traffic classification", Proceedings of the 2011 ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems, **(2011)** October, pp. 3-4, Brooklyn, NY, USA.
[9] D. Hui, S. Guanglu, L. Dandan and X. Feng, "Application Layer Traffic Classification Based on Link Homophily", Journal of Harbin University of Science and Technology, vol. 4, no. 18, **(2013)**.
[10] W. YiPeng, Y. XiaoChun, Z. YongZheng and L. ShuHao, "Network protocol identification based on active learning and SVM algorithm" Journal of Communications, vol. 10, **(2003)**.
[11] X. Liang and Z. Fengbin, "Recent Advances and Prospects of Immunity-based Intrusion Detection Systems", Journal of Harbin University of Science and Technology, vol. 2, no. 19, **(2014)**.
[12] Z. Hong-Li, L. Gang, "Machine Learning Algorithms for Classifying the Imbalanced Protocol Flows", Evaluation and Comparison. Journal of Software, vol. 6, no. 27, **(2012)**.
[13] T. T. T. Nguyen, "A Novel Approach for Practical", Real-Time, Machine Learning Based IP Traffic Classification", Swinburne University of Technology Melbourne **(2009)**, Australia.
[14] A. Dainotti, A. Pescape and K. C. Claffy, "Issues and future directions in traffic classification", Network, IEEE, vol. 1, no. 26, **(2012)**.
[15] A. Dainotti, A. Pescapé and C. Sansone, "Early classification of network traffic through multi-classification", Traffic Monitoring and Analysis, vol. 6613 **(2011).**
[16] A. Dainotti, A. Pescapé, and H. Kim, "Traffic classification through joint distributions of packet-level statistics", Proceedings of Global Telecommunications Conference (GLOBECOM 2011), **(2011)** December, pp. 5-9, Houston, TX, USA.
[17] C. G. Yin, S. Q. Li and Q. Li, "Network traffic classification via HMM under the guidance of syntactic structure", Computer Networks, vol. 6, no. 56, **(2012)**.
[18] http://netsecurity.51cto.com/art/200807/78902.htm
[19] B. L. Lu, M. Ichikawa, "Emergence of Learning: An Approach to Coping with NP-complete Problems in Learning", Proceedings of the International Joint Conference on Neural Networks, **(2000)** July, pp. 24-27, Como, Italy.

Authors

Zhang Luoshi, was born in 1983. He is a Ph.D. candidate at School of Computer Science and Technology, Harbin University of Science and Technology. His research interests include networking security, protocol identification and traffic management, *etc.*



Xue Yibo, was born in 1967. He received his M.S. and Bachelor degree in computer science and engineering at Harbin Institute of Technology in 1992 and 1989 respectively, and Ph.D. degree in computer architecture at Institute of Computing Technology, Chinese Academy of Sciences in 1995. Now he is a professor at Research Institute of Information Technology, Tsinghua University. He is a senior member of CCF and a member of IEEE/ACM. His research interests include computer network and information security, parallel processing and distributed system. He has published more than 100 papers in journals and conferences and applied for more than 30 patents

Bao Yuanyuan, was born in 1984. Tsinghua University, Assistant Professor, Research Area: social networks, data mining, information diffusion.