Traffic Classification Based on Incremental Learning Method

Guanglu Sun, Shaobo Li, Teng Chen, Yangyang Su, and Fei Lang(✉)

School of Computer Science and Technology,
Harbin University of Science and Technology, Harbin, China
langfei@hrbust.edu.cn

Abstract. Machine learning methods become more and more important in traffic classification, because they are able to explore statistical features to identify encrypted traffic and proprietary protocols. Among many machine learning methods, support vector machine is able to achieve state of the art performance in classifying TCP traffic. However, current support vector machine for traffic classification also shows two limitations: (i) unable to support continuously learning, and (ii) high requirements on both memory and CPU. In this paper, incremental Support Vector Machine method is applied to address these two issues. Experimental results show that the incremental Support Vector Machine method decreases the training time, while still sustains the high accuracy of traffic classification.

Keywords: Traffic classification · Incremental learning
Support vector machine

1 Introduction

Internet traffic classification has attracted a lot of research interests in recent years. The ability to identify flows and their relevant protocols is required for many applications, such as security and QoS.

The traditional methods of traffic classification are based on well-known port numbers and deep packets identifications [1]. They become ineffective to deal with unknown protocols, and even variants of known protocols, because of dynamic port numbers, encrypted payloads, etc.

Since 2004, many machine learning models have been introduced to exploit network behaviors and statistical characteristics to address these issues [2, 3]. Two representative methods have shown outstanding performance at that time. Moore's Bayesian method used two types of Bayesian models and feature selection methods based on the Cambridge open data sets [4]. The Support Vector Machine model

This work was partly financially supported through grants from the National Natural Science Foundation of China (No. 60903083 and 61502123), Scientific planning issues of education in Heilongjiang Province (No. GBC1211062), and the research fund for the program of new century excellent talents (No. 1155-ncet-008).

G. Sun and S. Liu (Eds.): ADHIP 2017, LNICST 219, pp. 341–348, 2018.
https://doi.org/10.1007/978-3-319-73317-3_40

(SVM) was applied on three types of well-known data sets, CAIDA, LBNL and UNIBS [5]. SVM obtained an average accuracy over 95%, 2.3% over the best performance of Bayesian methods and other methods on the same data sets [1]. As a result, SVM has become a favored method.

Although SVM is able to achieve impressive performance, it shows two main limitations in practice.

(1) **The lack of ability in continuous learning**. Because SVM has a high training complexity [6], it is difficult to update the classification model in-time when identifying new protocols.
(2) **High requirements on both memory and CPU**. More statistical features help us achieve higher accuracy, but they also consume more memory and CPU resources. The large numbers of traffic and protocols will result in a high-dimensional feature space in a backbone network. It demands us to effectively utilize memory and CPU to process these features in a training model.

In this paper, we propose an incremental method to address the above two limitations, by reducing the learn time for model update and efficiently utilizing memory and CPU resources. Incremental Support Vector Machine (ISVM) is instrumental in practical applications of online learning, which is advantageous when dealing with very large or non-stationary data [7, 8]. ISVM incorporates additional training data without re-training from scratch. Traffic classification based on ISVM is better than traditional SVM not only in accuracy, but in the consumption of system resource. The main contributions of this paper are as follows.

- Incremental SVM is firstly applied to classify Internet traffic.
- The update-time for traffic classification is decreased by adopting the ISVM learning model.
- The continual update of traffic classification model is achieved by using the ISVM method.

The remainder of this paper is organized as follows. In Sect. 2, we discuss existing literature related to our work. In Sect. 3, we show the theoretical details of the ISVM method, and explain how to use ISVM to realize the traffic classification in incremental update module. In Sect. 4, we show empirical results on the open real-world data sets to evaluate the effectiveness of ISVM method in traffic classification. In Sect. 5, we conclude the paper.

2 Related Work

As the increasing deployment of many encrypted protocols, port-based and payload-based methods become less attractive while machine learning based methods gain more attention. McGregor et al. firstly used unsupervised machine learning techniques to cluster traffic flows [9].

In this paper, we mainly focus on supervised learning methods used for traffic classification. The supervised machine learning model is built based on the labeled traffic flows, while statistical patterns are abstracted from the flows as the features.

After the adjustment of estimation parameters in the training phase, the model is then used to classify the new traffic flows. Following the above procedure, a lot of machine learning models were implemented in traffic classification. Williams et al. [10] compared five supervised algorithms including naive Bayes with discretization, naive Bayes with kernel density estimation, C4.5 decision tree, Bayesian network and naive Bayes tree, from the aspects of classification accuracy and computational performance. Finamore et al. [11] presented statistical characterization of payload as features and used SVM to conduct traffic classification. Nguyen et al. [12] trained the machine learning models with a set of sub-flows and investigated different sub-flow selection strategies. The accuracy of their models would be maintained when the traffic mixed up bi-directional flows. Ye and Cho [13] proposed an improved two-step hybrid P2P traffic classification with heuristic rules and REPTree model with different levels of features. Li et al. [14] utilized logistic regression model to classify the flows via non-convex multi-task feature selection. They tried a Capped as the regularizer to learn a set of features in traffic flows. Peng et al. [15] verified that 5–7 packets are the best packet numbers for early stage traffic classification based on 11 well-known supervised learning models.

3 Traffic Classification Based on ISVM Model

We first discuss how traffic classification is transformed into a classical classification problem. Consider a set of flows $T = \{t_1, t_2, \ldots, t_n\}$ and a set of application protocols $P = \{p_1, p_2, \ldots, p_l\}$, each flow belongs to one of application protocols $<t_i, p_j>$. Based on the mapping pairs tagged in a training set, the goal of a machine learning model is to find a discriminative function, by which t^* is classified to protocol p^* correctly.

$$p^* = Func(t^*) \quad p^* \in P,\, t^* \text{ is a pending flow} \tag{1}$$

SVM is a discriminative model which has strong theoretical basis and many empirical successes [6]. We introduce SVM in Sect. 3.1, and then present an incremental learning method for SVM and discuss how to solve the two limitations of traditional SVM model in Sect. 3.2.

3.1 SVM Model

SVM is introduced as a binary classification in batch training. We assume the training data and their labels are given as follows:

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\},\ x_i \in \Re^d,\ \ y_i \in \{+1, -1\}.$$

SVM builds the hyperplane that separates the training data by a maximal margin. The hyperplane is defined by the equation $w \cdot x + b = 0$, where w is a coefficient vector, b is a scalar offset, and the symbol "$\cdot$" denotes the inner product in $\Re^d$, defined as:

$$f(x) = w \cdot x = \sum_{i=1}^{n} w_i x_i \tag{2}$$

Data lying on each side of the hyperplane are respectively labeled as −1 or 1. Through Mercer kernel function $K(x_j, x_k) = \Phi(x_j) \cdot \Phi(x_k)$, e.g. linear, polynomial and RBF kernel, SVM maps the original training data in space X to a higher dimensional space F in order to classify the data that is impossible to be separated in a low dimension space. Using Lagrange interpolation coefficients α_i, Formula (2) is transferred to solve a quadratic programming problem with linear constraints and its dual form with respect to vector $\alpha_i, i = 1\ldots n$. The final discriminative function is:

$$f(x) = sign(w \cdot \Phi(x) + b) \tag{3}$$

Where $w = \sum_{i=1}^{n} \alpha_i y_i \Phi(x_i)$, $b = -\frac{1}{2}\left(\sum_{x_a, x_b \in \{x_i\}} \sum_{i=1}^{n} \alpha_i y_i \Phi(x_a)\Phi(x_b)\right)$.

SVM optimizes the discriminative function with coefficients using all the training data based on sequential minimal optimization techniques. However, not all the samples but support vectors (SV) (whose coefficients are not equal to zero) decide the hyperplane and the discriminative function. SVs absolutely present the class characteristics of the training data, when kernel function and other coefficients are defined.

3.2 Incremental SVM Model

Because traffic is changing over time in a real network, it becomes a challenge for the traditional SVM model to take new and large-scale new traffic into account, and combine them with the previously trained model. With a large amount of non-stationary data, ambiguous traffic, e.g. different traffic distributions varying over time, is hardly integrated by the traditional SVM model. So it is essential to improve the SVM algorithms to avoid completely retraining with huge CPU and memory overheads.

The ISVM model discards the original training data except of the SVs which are acquired by the last training of SVM model. When the additional new training data is joining, ISVM model combines the new data with the existing SVs, then use the combined data to retain SVM in order to get new SVs. Figure 1 shows the procedure of ISVM model.

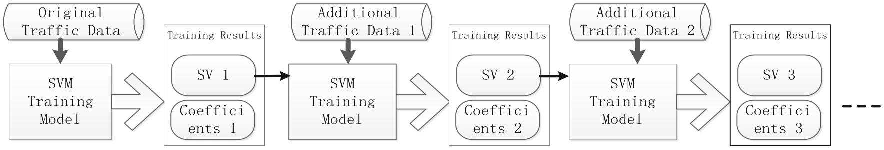

Fig. 1. The sketch map of ISVM learning model

3.3 Incremental SVM Model for Multi-class Traffic Classification

Because a protocol set contains more than two classes, the one-against-all approach is utilized to expand the binary SVM model to multi-class SVM model.

The characteristics of flow t_i are described as a vector of statistical features $F_i = \{f_{i1}, f_{i2}, \ldots, f_{im}\}$, which are numeric or discrete values, e.g. packet length. F_i corresponding to x_i can be denoted as $\{(F_i, p_i)\}$, while p_i corresponds to y_i. Based on the model introduced in Sect. 3.2, we use the multi-class ISVM model for traffic classification with training and test modules.

4 Experimental Results and Discussions

4.1 Data Sets and Evaluation Metrics

The data sets with more than 200 features developed by Moore et al. are used in our experiments [4]. For convince, we tag the data sets from M1 to M10. M1 is divided into ten parts in the data sequence for training, the other 9 data sets are used for test.

The metric of True Precision (TP) is used to evaluate the accuracy of the classification in each model. The results are obtained for the whole system instead of per class. The training time is shown with the style of H(hour), M(minute) and S(second). The number of SVs is the occurrence number of SVs in SVM after the current training process.

4.2 Results and Discussions

4.2.1 The Results of Standard SVM Method

We first present the results based on standard SVM by progressively increasing the training data set. In order to reflect the variation of TP, training time, and the number of SVs, ten parts of training data sets are added to the training module one by one. Table 1 shows the results with the standard SVM method. The result in the 10th column shows all the M1 data is added to the training model. In the M1 row, the result is with closing test, because M1 is training data set. The other rows are with open tests. The Average row is the average TP with M2 to M10 data sets.

In Table 1, along with the increasing of training data, the TP is not always increasing. Because SVM is a discriminative model, its performance does not absolutely depend on the increasing of training data, but on the occurrences of SVs. However, the results are promising considering the increasing trends.

On the other hand, the training time and the number of SVs are growing which increases the complexity and the resources consumption of both CPU and memory. In the 10th column, the scale of training data set is 24863. The corresponding training time is 86 h, 44 min and 51 s. Because the categories and the scale of actual traffic are much more than the experimental data set, the model update with large-scale traffic data is difficult for the traditional SVM model and other learning models.

Table 1. The results based on standard SVM.

	1	2	3	4	5	6	7	8	9	10
Time (H:M:S)	0:11:36	0:26:06	0:40:02	1:10:09	4:31:29	8:33:34	19:48:24	39:13:48	62:57:09	86:44:51
SV	20	86	86	130	263	363	540	657	887	1071
M1(%)	87.8	88.1	87.8	88.6	94.8	97.0	98.1	99.2	99.6	99.8
M2(%)	78.0	77.9	79.3	74.7	76.7	78.0	88.8	83.6	77.2	94.4
M3(%)	72.1	71.7	74.3	70.2	77.7	78.2	90.6	89.0	93.7	96.6
M4(%)	83.7	81.9	83.2	80.4	73.8	77.5	93.3	89.8	86.2	97.9
M5(%)	91.7	90.7	91.6	87.2	92.2	84.0	93.3	92.1	93.5	96.4
M6(%)	79.3	83.0	84.8	78.7	62.6	71.9	79.3	84.9	91.6	98.0
M7(%)	81.4	84.1	87.5	89.8	94.6	94.8	95.9	91.4	97.2	97.8
M8(%)	84.3	77.7	84.5	83.2	74.2	76.7	89.0	77.8	72.9	97.7
M9(%)	79.8	73.6	80.2	80.3	72.9	74.0	87.6	79.0	75.7	96.1
M10(%)	89.8	90.5	90.9	88.5	90.3	90.1	89.0	86.2	63.5	91.5
Average(%)	82.9	81.3	84.8	83.1	81.0	81.9	89.9	84.92	80.3	96.0

4.2.2 The Results of ISVM Method

Secondly, we present the classification results of ISVM model. ISVM model can realize continuous learning and reduce the occupation of CPU and memory effectively. We conduct several experiments based on ISVM model by dividing the training data into different proportion. The training model is the incremental SVM algorithm described in Sect. 3.2. The division of training data set and the style of adding training data are described in Sect. 4.1. The statistics of each class is listed in Table 2.

Table 2. The results based on ISVM.

	1	2	3	4	5	6	7	8	9	10
Time (H:M:S)	0:11:40	+0:01:23	+0:01:20	+0:06:13	+0:26:16	+0:34:19	+0:55:43	+1:07:56	+1:16:09	+1:25:25
SV	20	21	20	41	223	299	526	560	711	768
M1(%)	87.80	88.20	87.80	88.80	91.20	96.50	97.00	99.00	98.00	99.40
M2(%)	78.00	71.90	74.00	82.80	85.20	87.40	90.00	90.40	91.70	81.30
M3(%)	72.10	67.40	69.40	80.90	87.30	91.40	96.10	95.30	95.90	96.90
M4(%)	83.70	76.80	79.60	90.40	90.60	92.00	96.00	96.90	97.80	97.80
M5(%)	91.70	86.80	90.40	92.10	91.70	94.00	93.60	97.30	97.10	98.00
M6(%)	79.30	78.50	77.40	84.00	88.40	82.40	60.30	94.00	95.00	97.90
M7(%)	81.40	78.30	81.00	90.00	92.10	94.20	95.40	96.30	97.30	97.70
M8(%)	84.30	72.10	81.20	85.20	91.80	93.10	95.40	94.80	96.20	94.20
M9(%)	79.80	68.00	76.70	83.90	88.90	87.40	91.90	92.00	93.80	91.30
M10(%)	89.80	81.30	87.20	91.20	87.70	84.50	88.50	92.70	92.50	94.50
Average(%)	82.90	75.20	80.40	87.70	89.50	89.50	91.20	94.10	95.00	94.20

4.2.3 Comparison with Two Methods

Table 3 gives the performance comparison with the above two methods. Column (a) is based on the standard SVM. Column (b) is based on ISVM. The number of SVs significantly impacts the performance of SVM model. More SVs usually mean better TP, while more training data often generate more SVs. However, more training data results in the rapid increasing of computational cost. ISVM method decreases the occupancy of CPU and memory with less training data in each training process.

Table 3. The performance comparison with the two methods.

	(a)	(b)
Time (H:M:S)	86:44:51	6:06:24
SV	1071	768
M2(%)	94.40%	81.30%
M3(%)	96.60%	96.90%
M4(%)	97.90%	97.80%
M5(%)	96.40%	98.00%
M6(%)	98.00%	98.00%
M7(%)	97.80%	97.70%
M8(%)	97.70%	94.20%
M9(%)	96.20%	91.30%
M10(%)	91.50%	94.50%
Average(%)	96.00%	94.20%

5 Conclusions

In this paper, we use incremental pattern for identifying TCP traffic, by using incremental learning SVM. We demonstrate the effectiveness of ISVM model in continuous learning and the reduction of CPU and memory usage.

The experimental results show that our solutions are not only more accurate but also CPU and memory efficient. Incremental learning is advantageous when dealing with very large or non-stationary data. As the original training is completed, the incremental learning method has the ability to learning new data continuously without losing the previously trained model.

References

1. Kim, H., Claffy, K.C., Fomenkov, M.: Internet traffic classification demystified: myths, caveats, and the best practices. In: Proceedings of ACM CoNEXT 2008, Spain, 10–12 December 2008
2. Karagiannis, T., Papagiannaki, K., Faloutsos, M.: BLINC multilevel traffic classification in the dark. In: SIGCOMM 2005, USA, 22–26 August 2005
3. Nguyen, T., Armitage, G.: A survey of techniques for Internet traffic classification using machine learning. IEEE Commun. Surv. Tutor. 1–21 (2008)
4. Moore, A., Zuev, D.: Internet traffic classification using Bayesian analysis techniques. In: ACM SIGMETRICS 2005, Banff, Alberta, Canada, June 2005, pp. 50–60 (2005)
5. Este, A., Gringoli, F., Salgarelli, L.: Support vector machines for TCP traffic classification. Comput. Netw. **53**, 2476–2490 (2009)
6. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995). https://doi.org/10.1007/978-1-4757-2440-0
7. Syed, N., Liu, H., Sung, K.: Incremental learning with support vector machines. In: Proceedings of IJCAI-1999, Sweden, pp. 352–356 (1999)

8. Laskov, P., Gehl, C., Kruger, S., Muller, K.: Incremental support vector learning: analysis, implementation and applications. J. Mach. Learn. Res. **7**, 1909–1936 (2006)
9. McGregor, A., Hall, M., Lorier, P., Brunskill, J.: Flow clustering using machine learning techniques. In: Proceedings of the Passive Active Network Measurement, pp. 205–214 (2004)
10. Williams, N., Zander, S., Armitage, G.: A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. ACM SIGCOMM Comput. Commun. Rev. **36**(5), 5–16 (2006)
11. Finamore, A., Mellia, M., Meo, M., Rossi, D.: KISS: stochastic packet inspection classifier for UDP traffic. IEEE/ACM Trans. Netw. **18**(5), 1505–1515 (2010)
12. Nguyen, T., Armitage, G., Branch, P., Zander, S.: Timely and continuous machine-learning-based classification for interactive IP traffic. IEEE/ACM Trans. Netw. **20**(6), 1880–1894 (2012)
13. Ye, W., Cho, K.: Hybrid P2P traffic classification with heuristic rules and machine learning. Soft. Comput. **18**(9), 1815–1827 (2014)
14. Li, D., Hu, G., Wang, Y., et al.: Network traffic classification via non-convex multi-task feature learning. Neurocomputing **152**, 322–332 (2015)
15. Peng, L., Yang, B., Chen, Y.: Effective packet number for early stage internet traffic identification. Neurocomputing **156**, 252–267 (2015)