This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2011 proceedings

# An Internet Traffic Classification Method Based on Semi-Supervised Support Vector Machine

Xiang Li, Feng Qi, Dan Xu, Xue-song Qiu
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China, 100876
{lxxuanyuan, qifeng, xudanbupt, xsqiu}@bupt.edu.cn

***Abstract*—Identifying and classifying different network applications is very important for trend analysis, dynamic access control, network security and traffic engineering, while traffic classification is able to classify applications effectively. Current popular methods of traffic classification mainly include machine learning algorithm based on supervised or unsupervised. In practical applications, the above methods have high complexity or low accuracy degree, so we propose a semi-supervised support vector machine method only based on flow statistics to identify and classify network applications. In this method, SVM, "constant" flow and co-training algorithm are the key to obtain a classifier rapidly. The classifier got by this method has three advantages contrast to the previous classical methods: 1) high classification degree; 2) high generalization performance; 3) rapid computational performance. As a proof of the concept, we implement the classification algorithm based on open-resource, and show the characteristics and feasibility of our method in the campus and resident network.**

***Keywords-Internet traffic; Network traffic classification; Machine learning; Semi-Supervised; SVM***

## I. INTRODUCTION

A variety of network applications are running on Internet currently and new applications are still emerging. So how to classify applications accurately and identify new applications has an important significance for network administrators, researchers, service providers and users. Every application has its own traffic behavior, so traffic classification methods are studied to classify and identify applications [1, 2].

The traffic classification method based on machine learning only builds on statistical characteristics of flow and is easy to expand and maintain. So it has attracted wide attention from scholars [3, 4, and 5]. The current study of traffic classification based on machine learning has focused on supervised learning form and unsupervised learning form. The method based on supervised learning form manually classifies labels and then models all samples, which not only has a huge workload, but also depends on the understanding of the samples. What's more, it is unable to identify unknown applications. The method based on unsupervised learning form can find the structural knowledge hidden in training cases through learning the training samples without labels. Although it is able to find unknown applications, it has lower classification accuracy and more difficult training processes.

This paper presents a traffic classification method based on semi-supervised support vector machine (SVM). Our method makes full use of high accuracy and robustness of SVM [4], as well as improves and enhances the semi-supervised learning form. The semi-supervised learning form [6] not only improves the speed and accuracy of classifier through a few expensive labeled flows and a large number of cheap unlabeled flows, but also classifies unknown applications and changed known applications.

Many supervised classifiers have high accuracy in the training sets, but poor performance and weak adaptation in the testing sets. Our approach in this paper solves the above problems successfully through "constant" labeled flows, the co-training style [7] of semi-supervised learning and SVM. What's more, it reduces the workload of manual classification and improves the efficiency of classification. In a word, the major advantages of our approach include high classification degree, high generalization performance and rapid computational performance.

The remainder of this paper is organized as follows: related work is presented in section II. The algorithm and method is described in section III, and semi-supervised SVM architecture for traffic classification is proposed in section IV. The two sections are crucial contents in this paper. Then section V introduces data sets collection and pretreatment. The experiments and testing results are presented in section VI. In the end, section VII concludes the paper and forecasts the future work.

## II. RELEATED WORK

**Supervised machine learning.** Supervised learning method has become one of the most concerned traffic classification methods. [8] made use of Nearest Neighbor and Linear Discriminate Analysis to map different applications to different QoS levels. [9] studied network applications identification based on Naive Bayes. [10] identified application protocols through simple fingerprint statistics. [3] compared the performance of several supervised learning algorithms.

**Unsupervised machine learning.** Unsupervised learning method only uses unlabeled data, while labeled data is used to test its learning performance as testing data. [8] constructed a flexible traffic generator through a clustering method based on flow communication mode. [11] used Expectation Maximization to classify flows into different applications. [12] used Sequential Forward Selection and Autoclass clustering method to identify applications. [13] evaluated and compared

978-1-61284-233-2/11/$26.00 ©2011 IEEE

the clustering algorithms, including KMeans, DBSCA and AutoClass.

**Semi-supervised machine learning.** Semi-supervised learning method is an emerging learning mechanism in recent years, which just starts in the network traffic classification. [14] firstly applied semi-supervised in network applications identification, but the author only used a kind of clustering algorithm, lack of comparing with other algorithms. Meanwhile, [14] clustered flows through unsupervised method, and used unlabeled flows to map clusters into applications, which was very different from our method.

## III. ALGORITHM AND METHOD

### A. SVM

A support vector machine constructs a hyperplane or set of hyper planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks [15]. SVM method has ability to simultaneously minimize the empirical classification error and maximize the geometric margin classification space. These properties reduce the structural risk of over-learning with limited samples. The key of SVM is kernel function. In this paper, we use radial basis function (RBF) kernel function selected by experiences to get classifiers. Another important parameter in SVM is penalty factor C stating degree of punishment, which has a great impact on the experimental results.

### B. Co-training

Co-training algorithm [6] is a semi-supervised learning technology, which uses two kinds of independent complete feature sets to describe objects based on multi-view. In the process of co-training, every classifier selects and marks several samples with higher degree of confidence from unlabeled samples, and then adds labeled samples to labeled training sets of another classifier, which help another classifier update with these new labeled samples. The process of co-training iterates continuously, until meeting pre-conditions. Under ideal condition, co-training requires two views are independent and every feature set can get a strong classifier. In the paper, we use the feature selection algorithm based on two evaluation metrics to obtain two feature sets, which satisfies the input requirement of the co-training semi-supervised algorithm in a large extent.

### C. Feature Selection Algorithm

Feature selection is to remove irrelevant or redundant features from candidate feature sets and select an optimal feature subset under the condition of guarantying or not reducing classification accuracy. In order to meet the independent requirement of two views according to co-training, we use the filter mode‘s feature evaluation metrics of Consistency-Based Feature (CBF) and Information Gain (IG) [16, 17] to obtain a streamlined feature set. In order to get a streamlined data set from the original large data sets and maintain the integrity of the original data sets, this paper uses Sequential Forward Selection (SFS) algorithm to search feature subsets. SFS algorithm is a greedy selection process, which starts from an empty feature subset, and adds a feature every time, and the feature added has the greatest evaluation metrics.

### D. “Constant” Labeled Flows Selection Algorithm

We introduce “constant” labeled flows selection algorithm (CLFS) in this section. The “constant” flow is required a strong adaptability for improving the generalization performance of traffic classification model. In addition, the original kernel parameter initγ and penalty factor C are got through experience.

**Algorithm CLFS: “Constant” Labeled Flows Selection**

**Input:** Total training set A (labeled), punishment C and γ parameters of RBF, minγ, maxγ, initγ
**Output:**”constant” flows B
**begin**
  B=A;
  γ=initγ;
  **while** ($\boldsymbol{\gamma} \in [\mathbf{min\gamma}, \mathbf{max\gamma}]$) **do**
    gain classifier from B by SVM algorithm;
    **for** ($\mathbf{x} \in \mathbf{B}$) **do**
      **if** (x is not support vector with C)
      remove the **x** from B;
    **end if**
    **end for**
    $\boldsymbol{\gamma} = \boldsymbol{\gamma} \pm \mathbf{0.03}$;
  **end while**
  **return** B;
**end**

Algorithm CLFS uses SVM based on RBF kernel function to select the most representative flow from lots of labeled flows. In every step of the cycle, we remove flows which are not support vectors from the original training flows set according to kernel parameters and penalty factor. When satisfying our pre-condition, the process terminates, and the remaining flows are “constant” flows.

## IV. SEMI-SUPERVISED SVM ARCHITECTURE FOR TRAFFIC CLASSIFICATION

### A. Classification Object

From the resource utilization and QoS requirement perspective, network applications are usually divided into a few categories referring to [4, 9]. A typical classification is based on the application characteristics. The Table I shows 10 categories and their example including unknown application. We will research more accurate traffic classification on the application layer in the future.

TABLE I
INTERNET TRAFFIC CATEGORIES

| Class | Representative Application/Protocal |
|---|---|
| WWW | http,https |
| FTP | ftp |
| DNS | dns |
| Mail | smtp,pop3,imap |
| Multimedia | voice,video streaming |
| Interactive | ssh,telnet,rlogin |
| Chat | qq,msn,yahoo |
| P2P | Kazaa,Bittorrent,Gnutella,Thunder,uTorrent |
| Game | WoW,WarCraft,Half-life |
| Unknown | |

### B. Experiment process

In this section, we discuss the experiment process of traffic classifier based on semi-supervised SVM. Our method is different from traditional semi-supervised application and shown as follows:

Step 1: “Constant” Flow Selection. The first step of training the classifier is to create a set of labeled flows and use the algorithm CLFS. Based on semi- supervised theory, the aim of

using unlabeled flows is to adapt to inherent characteristics of the traffic under the current environment, and then improve the generalization ability of the classifier.

Step 2: Classification. Depending on two feature subsets (CBF and IG), Step 2 uses the SVM to train two classifiers on labeled flow. The two classifiers are taken for the input of Step 3 co-training algorithm, which effectively enhances the accuracy of the final classifier.

Step 3: Co-training. The input of co-training semi-supervised classification algorithm is "constant" labeled flows and a large number of unlabeled flows obtained from the environment to be classified. Through the two classifiers obtained in Step 2, we adopt the co-training algorithm referred in Section III to carry on semi-supervised training. When the algorithm ends, the final traffic classifier is acquired.

## V. DATA SET

### A. Description of Data Sets

TABLE II
DATA SET FOR NETWORK FLOW EXPERIMENT (CAMPUS TRACES)

| Traffic Class | Bytes | Number of Packets | Number of Flows |
|---|---|---|---|
| WWW | 2.92GB | 7,538,462 | 63,406 |
| FTP | 6.33GB | 12,198,334 | 9,847 |
| DNS | 0.85GB | 2,913,896 | 22,485 |
| Mail | 0.48GB | 1,371,355 | 13,049 |
| Multimedia | 4.73GB | 7,080,415 | 3,520 |
| Interactive | 0.01GB | 11,207 | 227 |
| Chat | 1.14GB | 2,730,304 | 26,741 |
| P2P | 14.7GB | 31,023,819 | 31,397 |
| Game | 1.36GB | 3,890,237 | 21,482 |
| Unknown | 2.31GB | 6,893,572 | 35,123 |
| Total | 34.47GB | 75,651,601 | 227,277 |

TABLE III
DATA SET FOR NETWORK FLOW EXPERIMENT (RESIDENT TRACES)

| Traffic Class | Bytes | Number of Packets | Number of Flows |
|---|---|---|---|
| WWW | 1.87GB | 5,139,406 | 43,850 |
| FTP | 0.83GB | 1,851,098 | 1,495 |
| DNS | 0.79GB | 2,485,204 | 23,107 |
| Mail | 0.22GB | 713,097 | 9,041 |
| Multimedia | 5.38GB | 9,034,315 | 4,830 |
| Interactive | 0.01GB | 20,406 | 482 |
| Chat | 1.06GB | 2,487,309 | 27,132 |
| P2P | 19.4GB | 48,371,014 | 44,129 |
| Game | 3.48GB | 7,481,592 | 51,970 |
| Unknown | 1.31GB | 24,590,23 | 18,593 |
| Total | 34.35GB | 80,042,467 | 224,629 |

Data sets used are described in this section. In order to facilitate our work, we use Jpcap open-source toolkit based on winpcap/libpcap [18] to collect data in the network. Because five-tuple array (including source address, source port number, protocol, destination address, destination port number) can determine the unique flow, we consider the same five-tuple array during the close interval as the same flow. The data packets are firstly divided into uni-directional flows according to five-tuple array, and then uni-directional flows are combined into bi-directional flows. Although the method of flows statistics is used, we intercept the complete information of packets, because we need to use the application layer information to determine the categories of flows in later analysis training. Table II and III respectively demonstrates two kinds of network data sets traced.

Depending on different subnets traced, all the data collected are classified into the campus network and the resident network. The campus network is a special network for universities, which concentrates lots of educational resources, while the resident network is an ordinary commercial network. The different applications of the two networks will satisfy generalization performance tests of different networks later.

Because of our limited disk space, we take an hour in a day to collect flow data of the campus and resident network from backbone network link in a week. We develop a simple filter to filter data packet. The filter can check the payload and filter out not TCP/UDP data packets. So we only capture and analysis TCP/UDP flows in our experiment.

### B. Flow Feature Definitions

The flow features defined preferably have distinction and low cost, which can obtain maximum interval with minimal cost. At the same time, flow features selection is also restricted by the actual IP network resources. In our selection, the bottom acquisition based on libpcap packets can obtain more data.

We have selected 30 flow features including 8 bi-directional and 11 unidirectional flow features, which are based on 248 bi-directional flow features in [19]. 30 flow features are shown in Table IV, which we refer to as the full feature set:

TABLE IV
THE FULL FLOW FEATURES

| |
|---|
| bidirectional flow features |
| The protocol (TCP or UDP)<br>The flow duration<br>Total number of packets in the flow<br>The average packets size of a flow<br>The version<br>The variance of window size<br>The number ratio of send and receive packets<br>The byte ratio of send and receive packets |
| unidirectional flow features(send or arrival) |
| Port<br>Flow volume in bytes and packets<br>Packet length (minimum, mean, maximum and variance)<br>Inter-arrival time between packets (minimum, mean, maximum and variance) |

Our features are simple and well understood because they do not need payload. They represent a reasonable benchmark feature set, and more complex features might be added in the future.

## VI. TRAFFIC CLASSIFICATION EXPERIMENT BASED ON SEMI-SUPERVISED SVM

### A. Experiment Setup

This section mainly introduces the setup of traffic classification experiment and preliminary data processing. The algorithm in this experiment is implemented by procedures based on WEKA 3.7 [20]. SVM in this experiment uses the LibSVM [21] open-source package.

TABLE V
THE COMPOSITION OF "CONSTANT" FLOWS

| Traffic class | Number of flows |
|---|---|
| WWW | 826 |
| FTP | 247 |
| DNS | 268 |
| Mail | 229 |
| Multimedia | 94 |
| Interactive | 107 |
| Chat | 355 |
| P2P | 507 |
| Game | 381 |
| Unknown | 0 |
| Total | 3014 |

*1) "Constant" flow selection model:* We choose 3014 flows as "constant" labeled flows in the semi-supervised method, as shown in Table V. To ensure stability of the final classification of the campus network, the full feature set is used in the process of "constant" flows selection. Fig. 1 shows the influence of the penalty parameter C and the factor γ in RBF kernel function for classification accuracy in training set. From the figure, we find the classifier can easily reach 100% classification accuracy in a large scale, due to many flows with non-standard behavior have been excluded. Because classification accuracy is stable in certain scale, in the later experiment the factor γ is set 1/4 and C is set 1024.

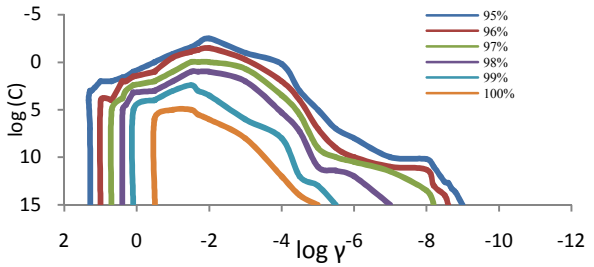


Fig. 1. Impact of penalty parameter and RBF function on classification accuracy.

*2) Feature selection model:* Table VI shows the feature selection result, which can be seen elements in the feature sets are very different. It satisfies the training requirement of co-training semi-supervised algorithm. Fig. 2 shows the comparison of the mean accuracy of the two feature subsets with the full feature set. We use supervised SVM algorithm to choose 30000 flows to train classifiers, and access the mean accuracy based on 10 fold cross-validation under the same condition. From the Fig. 3 we can see the accuracy of CBF and IG is respectively 95.7% and 94.9%, which is lower compared to the full feature set. And with the number of features in the feature set significantly reduces, for example the number of CBF and IG is respectively 11 and 14, which reduces redundancy and improves the classification rate.

TABLE VI
THE FEATURE SUBSETS ACCORDING TO CBF AND IG METHOD

| | |
|---|---|
| CBF subset | protocol, duration, averpacknum, arport, flowbyte, seminpkl, semeanpkl, arvarpkl, seminibp, arvaribp |
| IG subset | version, duration, varofws, nrofsrp, brofsrp, arport, flowpack, armeanpkl, armaxpkl, arvarpkl, semeanibp, sevaribp |

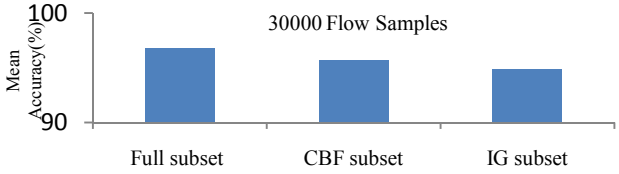


Fig. 2. Accuracy of algorithms using CBF subset, IG Subset and Full features.

## B. Accuracy

The evaluating metrics of classifier mentioned include accuracy, precision and recall. This section compares the three metrics between our classifier and other classical classification methods under the same conditions.

In the comparison of the overall accuracy, we compare semi-supervised SVM method with supervised SVM [15], unsupervised clustering [12] and the classifiers based on the above feature subsets (CBF and IG). Fig. 3 shows the growth of the overall accuracy with the size of the training set of the above methods. The semi-supervised method itself has 3014 labeled flows, so the X-coordinate represents the sum of labeled flows and unlabeled flows. In addition, CBF and IG express the classifiers based on the two feature subsets before handled by co-training semi-supervised algorithm.

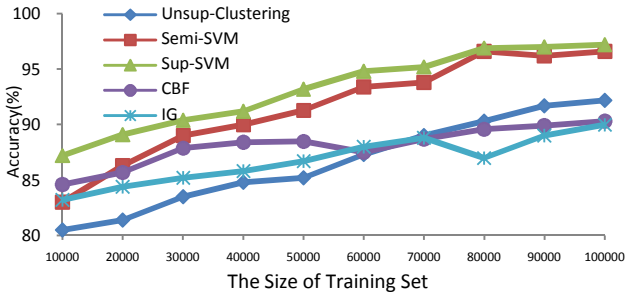


Fig. 3. Influence of the size of training set on classification accuracy.

From Fig. 3 we can see that the accuracy of classifiers based on CBF and IG is approximately equal. The classification accuracy has greatly improved through co-training; at the same time, the accuracy of semi-supervised SVM (semi-SVM) and supervised SVM (sup-SVM) is clearly higher than that of unsupervised clustering algorithm. The unsupervised method doesn't refer to labeled flows, so it has higher computing performance but lower classification accuracy. The supervised method has highest accuracy and is relative stable, but the cost of a lot of labeled flows is very large. In contrast, the accuracy of the semi-supervised SVM method is close to the supervised SVM method after 8000 training flows.

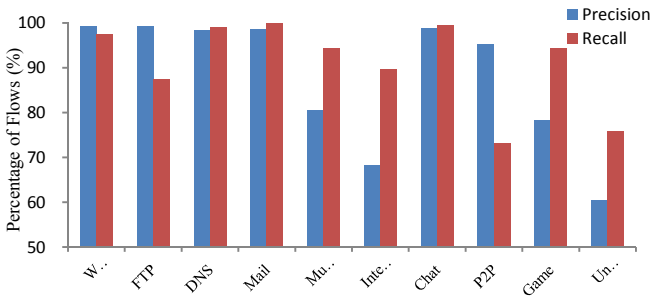


Fig. 4. Precision and Recall of Per-application Classification.

Fig. 4 shows the precision and recall of per-application using our method. We can see our method is able to reach more than 60% precision and more than 70% recall in unknown applications. The precisions of Multimedia, Interactive and Game are lower than all other applications, the reason of which is the three applications respectively have a lot of different protocols. Meanwhile, some traffic of these applications is encrypted.

## C. Generalization

In the comparison of generalization, we compare the classifier accuracy based on semi-supervised SVM, supervised SVM and unsupervised SVM in different testing sets. Because the unsupervised method do not need labeled flows and the semi-supervised method do not need additional new labeled flows, unsupervised clustering and semi-supervised clustering train the classifiers again in the resident network, while supervised clustering uses the original classifier. Fig. 5 and Fig. 6 show the performance of the three methods under two different networks.

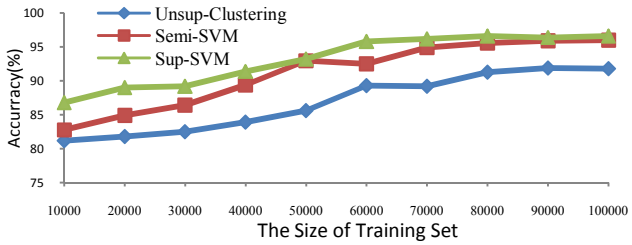


Fig. 5. The Size of Training Set-Accuracy in Campus Network.

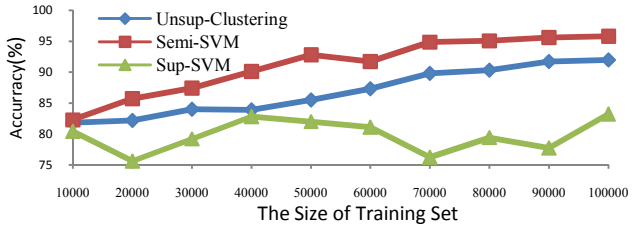


Fig. 6. The Size of Training Set-Accuracy in Resident Network.

The proportion of applications used by the resident network is very different from that of the campus network. Compared to Fig. 5, Fig. 6 shows the accuracy of semi-supervised SVM and unsupervised clustering is basically equal in the resident network, while the accuracy of supervised SVM decreases greatly due to the non-adaptation of training samples. In this experiment, we can conclude that the semi-supervised SVM method has a good performance under different environments, which shows a smaller structural risk and higher generalization performance.

### *D. Computational Performance*

Table VII shows the levels of time cost and mean accuracy degree of classifiers obtained by three different methods. The performance of Clustering algorithm is better than Semi-SVM and Sup-SVM, so its training speed is higher. From the result of the above experiment, we found that the mean accuracy of Clustering algorithm is lower than others. While Sup-SVM algorithm needs a lot of labeled flows corresponding to specific environment, so its preparing work needs to take a lot of time. The advantage of traffic classification based on semi-supervised SVM is rapid training time like unsupervised and high accuracy like supervised.

TABLE VII
THE COMPARISON OF COMPUTATIONAL PERFORMANCE

| | Training speed | Preparing work | Overhead time | Mean Accuracy |
|---|---|---|---|---|
| Semi-SVM | Medium | Once | Medium | High |
| Sup-SVM | Medium | Large | High | High |
| Clustering | High | No | Low | Medium |

## VII. CONCLUSIONS AND FUTURE WORK

The paper proposed and evaluated a semi-supervised SVM method only based on flow statistics, which is used to analyze various applications' traffic. The key advantage of semi-supervised SVM is to use a few labeled flows and lots of unlabeled flows to train a classifier with high accuracy and strong generalization performance. In the paper, "constant" flows enhance the training speed and performance of the classifier, and co-training algorithm based on the view of CBF and IG features selection improves the accuracy of the classifier. Our experimental results show the good performance of the classifier based on semi-supervised SVM.

Co-training study requires the data set have both sufficient and redundant views, so the focus of our future work is to study and evaluate other better classification methods in the semi-supervised learning mode. Meanwhile, it's also an important future direction how to re-train classifiers and evaluate sampling technology to extend the life of classifiers.

## VIII. ACKNOWLEDGEMENT

This work was supported in part by the 973 project of China (2007CB310703), Funds for Creative Research Groups of China (60821001), NSFC (60973108), Fok Ying Tung Education Foundation (111069) and NCET-10-0240

## REFERENCES

[1] CAIDA : research : traffic-analysis : classification-overview. http://www.caida.org/research/traffic-analysis/classification-overview/.

[2] S. Sen, J. Wang, "Analyzing Peer-to-Peer Traffic across Large Networks," IEEE/ACM Transaction Networking, 2004.

[3] N. Williams, S. Zander, G. Armitrage, "A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification," Computer Communication Review, 2006.

[4] H. Kim, K. Claffy, M. Fomenkov, "Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices," In CoNEXT'08.

[5] T. Nguyen and G. Armitage, "A Survey of Techniques for Internet Traffic Classification using Machine Learning," IEEE Communications Surveys and Tutorials, 2008.

[6] A. Blum, T. Mitchell, "Combining labeled and unlabeled data with co-training," COLT' 98.

[7] O. Chapelle, B. Schölkopf, A. Zien, eds. Semi-Supervised Learning, Cambridge, MA: MIT Press, 2006.

[8] F. Hernández-Campos, F. D. Smith, "Statistical Clustering of Internet Communications Patterns," Computing Science and Statistics 2003.

[9] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in Proc. ACM SIGMETRICS, June 2005

[10] M. Crotti, M. Dusi, F. Gringoli, "Traffic classification through simple statistical fingerprinting," SIGCOMM Comput. Commun., 2007.

[11] A. McGregor, M. Hall, P. Lorier, "Flow clustering using machine learning techniques," PAM2004.

[12] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," LCN 2005

[13] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in MineNet '06: Proceedings of the 2006 SIGCOMM workshop on Mining network data.

[14] J. Erman, A. Mahanti, M. Arlitt, "Offline/Realtime Traffic Classification Using Semi-Supervised Learning," In IFIP Performance, October 2007.

[15] R. Yuan, Z. Li, "An SVM-based machine learning method for accurate internet traffic classification," Information Systems Frontiers 2008.7.

[16] H. Liu, L. Yu, "Towards integrating feature selection algorithms for classification and clustering," IEEE Trans. on Knowledge and Data Engineering, 2005.

[17] L. Yu, H. Liu, "Efficient feature selection via analysis of relevance and redundancy", Journal of Machine Learning Research, 2004, 5.

[18] Jpcap. A Java library for capturing and sending network packets. http://netresearch.ics.uci.edu /kfujii/jpcap/doc/download.html

[19] A. W. Moore and D. Zuev, "Discriminators for use in flow-based classification," Technical report, Intel Research, Cambridge, 2005.

[20] WEKA: Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/.

[21] LibSVM. A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/