Inf Syst Front
DOI 10.1007/s10796-008-9131-2

An SVM-based machine learning method for accurate internet traffic classification

Ruixi Yuan · Zhu Li · Xiaohong Guan · Li Xu

Abstract Accurate and timely traffic classification is critical in network security monitoring and traffic engineering. Traditional methods based on port numbers and protocols have proven to be ineffective in terms of dynamic port allocation and packet encapsulation. The signature matching methods, on the other hand, require a known signature set and processing of packet payload, can only handle the signatures of a limited number of IP packets in real-time. A machine learning method based on SVM (supporting vector machine) is proposed in this paper for accurate Internet traffic classification. The method classifies the Internet traffic into broad application categories according to the network flow parameters obtained from the packet headers. An optimized feature set is obtained via multiple classifier selection methods. Experimental results using traffic from campus backbone show that an accuracy of 99.42% is achieved with the regular biased training and testing samples. An accuracy of 97.17% is achieved when un-biased training and testing samples are used with the same feature set. Furthermore, as all the feature parameters are computable from the packet headers, the proposed method is also applicable to encrypted network traffic.

Keywords Internet traffic · Network traffic classification · Machine learning · Feature selection · SVM

This paper was processed by Ling Li, R. Valerdi and J Warfield.

R. Yuan · Z. Li · X. Guan
Center for Intelligent and Networked Systems, TNLIST Lab,
Tsinghua University,
Beijing 100084, China

X. Guan (✉)
MOE KLINNS Lab and SKLMS Lab, Xi'an Jiaotong University,
Xi'an 710049, China
e-mail: xhguan@tsinghua.edu.cn

L. Xu
College of Economics and Management,
Beijing Jiaotong University,
Beijing 100044, China

L. Xu
Department of Information Technology and Decision Science,
Old Dominion University,
Norfolk, VA 23529, USA

1 Introduction

In recent years, more and more business applications are facilitated by Internet, ranging from e-commerce to e-business (Li et al. 2007a, b). The Internet has become an integral part of business activities of many businesses today (Beheshti et al. 2007; Guo 2007; Wang and Archer 2007). As a result, Internet traffic monitoring and control have attracted an increasing amount of interest in the past few years. In security perspectives, fast identification of malicious traffic will help security control and isolation of attackers. From the QoS perspective, accurate classification of different traffics helps to identify the application utilizing network resources, and facilitate the instrumentation of QoS for different applications. Furthermore, network operators can trace the growth of different applications and provision network accordingly to accommodate the diverse needs of user population.

Traditionally, the identification of network applications can be realized by locating the number of the known ports obtainable from the packet header. However, many new applications do not use known ports. The technique of port tunneling employs known ports (e.g. 80) to tunnel other applications. Therefore, a known port number is no longer a unique sign for a network application.

Precise signature matching is a widely used method in intrusion detection systems (Vigna et al. 2004; Shon and Moon 2007), and the most accurate method for traffic identification. However, these techniques are unable to adapt to new applications with no signatures or newer version of the same application in which the signature has been changed. Developing and maintaining an accurate signature database can be expensive and time-consuming. Furthermore, as a resource intensive operation, precise matching of a large number of signatures within the applications requires the capture and storage of application data across multiple IP packets. In addition, application signature matching is not feasible for network applications using encryption for data protection. For example, the data stream of the popular *Skype* application is encrypted to protect the voice and user data channel.

A fundamental aspect of traffic classification is classification granularity. Some classifications classify the traffic into two groups as "normal" and "abnormal" (Lakhina et al. 2004). Some attempt to identify the exact application types such as "BitTorrent" or "edondey" (Sen et al. 2004). A middle ground can also be found that classifies the traffic into multiple broad based categories, each represents a common group of application (Moore and Zuev 2005a). For example, the "MAIL" category would include "SMTP", "POP", "IMAP", and potentially other email applications.

Pattern recognition, which aims to classify data based on either a priori knowledge or statistical information extracted from raw data, is a powerful tool in data separation in many disciplines (Duan et al. 2007, 2008; Feng et al. 2001; Li and Xu 2001; Li et al. 2007a, b; Li et al. 2008; Luo et al. 2007; Shi et al. 1996, 1999, 2007; Xu 1999, 2006). The patterns to be classified are usually groups of observations (parameters), defining data points in an appropriate multidimensional space. Therefore, pattern recognition methods can be well suited with Internet traffic classification, as long as the traffic classified into categories that exhibit similar characteristics in parameters. Supervised machine learning methods, which use the known training samples to acquire feature parameters and build proper models to classify the unknown samples, is particularly pertinent to traffic classification.

Recently, to overcome the deficiencies of traditional traffic classification methods with port/protocol and IDS-type signature matching, several machine learning techniques were proposed to classify Internet traffic, each with reasonable successes. These related work are described in the next section.

2 Related work

A number of researches has attempted to classify the Internet traffic into exact applications. The flow duration time and average packet size of a flow were used to classify network traffic (Roughan et al. 2004). The nearest neighbor method and linear discriminant analysis are used to train the classifier from known samples. A classification accuracy of 90% has been achieved for seven applications {*domain, ftp-data, https, kazaa, realmedia, telnet,* and *www*}. Haffner reported that three statistical machine learning algorithms as Naive Bayesian, Adaboost and Regulated Maximum Entropy have been used to classify traffic based on the feature vector (n*256 elements) derived from the initial n bytes of application data (Haffner et al. 2005). Each dimension is a Boolean variable and whether it should be 1 or 0 depends on the value of the corresponding bytes. It was found that the Adaboost method yields a high degree of accuracy (99%) with both 64 bytes or 256 bytes of application data for seven distinct applications: {*ftp-control, smtp, pop, imap, http, https,* and *ssh*}. This work is equivalent to construct application signatures using machine learning method; hence a high dimensional feature vector is required.

Early adopted the decision tree method to train a decision tree classifier for {*http, ftp, smtp* and *telnet*} applications with the feature of the probabilities of FIN and PUSH packet, average packet size and RTT of a flow (Early et al. 2003). The unknown flows were then classified with the decision tree; an accuracy of 93% is obtained.

Bernaille proposed the use of few initial packets of a TCP flow to identify the application in early stage (Bernaille et al. 2006). The method uses the k-means algorithm to separate the traffic into clusters. An 80% accuracy is achieved for 7 distinct applications {*edonkey, ftp, http, kazaa, nntp, smtp, ssh, https, and pop3s*}, while the POP3 application was misidentified due to its falling into the categories of NNTP cluster.

Moore et al conducted a series of studies attempting to classify the network traffic into several broad-based groups as shown in Table 1 (Moore and Zuev 2005a). In this type of classification, applications with similar dynamics are classified into the same class. A naive Bayesian estimator is used in the algorithm in which the Bayes formula is used to calculate the posterior probability of a testing sample and select the largest probability class as the classification result.

Table 1 Internet traffic classes

Traffic class	Representative applications
Bulk	ftp
Interactive	ssh, telnet, rlogin
Mail	pop3, smtp, imap
Service	X11, dns
WWW	http, https
P2P	Kazaa, BitTorrent, Gnutella
Multimedia	Voice, video streaming
Game	Half-life
Attack	Worm, virus
Others	Scan, netbios, ntp, tsp

A total of about 200 features of a network flow is used to train the model and a kernel-based function is used to estimate the distribution function (Moore and Zuev 2005b). The total accuracy is about 95% in the dimension of flow number being correctly classified and 84% in the dimension of flow size.

While these methods offer various degrees of successes, there are several limitations:

1) The computation of these algorithms is highly complex. In one algorithm, for a single flow, about 200 features need to be selected (Moore and Zuev 2005b). In another algorithm, due to its high dimensionality, it takes several hours for the algorithm to converge (Haffner et al. 2005). Although a relatively high accuracy is achieved, these methods do not fit into the real-time situation due to their requirement on computation and storage.
2) Models with certain pattern recognition methods such as Bayesian estimating, decision tree, nearest neighbor, may be trapped into local optimization.
3) Accuracy is highly dependent on samples' prior probabilities. The training and testing samples may be biased towards a certain class of traffic. For example, the WWW traffic constitutes the large majority of the sample in (Moore and Zuev 2005a).

In numerous previous studies, the number of training and testing samples in each different application is based upon the actual ratio in the network. While it is reasonable for overall traffic, this can sometimes leads to unusually high classification accuracy. For example, a traffic sample of 95% WWW traffic has at least 95% classification accuracy when all WWW traffic is identified, even though it may misclassify all other traffic classes. Therefore, to obtain the effectiveness of a classification method, it is helpful to study the classification accuracy with unbiased training and testing samples.

To address the above-mentioned problems, we took several steps to improve the speed and accuracy of the machine learning methods for Internet traffic classification:

1) We reduced the number features from a network flow. All of the features can be obtained real- time from packet headers.
2) We used an SVM method, which is a maximum margin classifier and can avoid local optimization.
3) We compared the classification accuracy for both biased and unbiased training samples.
4) We adopted a discriminator selection algorithm to obtain the best combination of features for classification. This optimal set of discriminators not only yields high accuracy, but also offers insight into the factors affecting the classification. It can guide our future work of classifying network flows based on other pattern recognition methods.

As a result, our optimized method yields an accuracy of 97.17% for the unbiased training and testing samples when only 9 feature parameters are used. For regular network traffic (normally biased towards WWW in terms of flow numbers), the same method has an accuracy of 99.42%. This suggests that the discriminator optimization is independent of the traffic mix of the sample, but valid across a broad range of traffic profiles.

The remainder of this paper is organized as follows: Section 3 describes the data set we used for the experiments. Section 4 introduces the classification methodology using the SVM-based method with an RBF kernel function, the performance evaluation method using cross-validation and discriminator selection algorithms. Section 5 presents the experimental results and their analysis. Section 6 concludes the paper and discusses potential future works.

3 Data for experiment

The experiment data were obtained from a backbone router of the campus network of our university. A set of 8-hour traffic data on a Gbps Ethernet link within a one week period was collected. The packets were first separated into unidirectional flow according to the five tuples (*srcIP, desIP, Prot, srcPort, desPort*), and then the unidirectional flows were combined into bi-directional flows from the overlapping time spans of the flows. The first 250 KB payload data from the flow were also stored (if the payload is less than 250 KB, choose all) for offline traffic identification, because the vast majority of application signatures will appear in the initial part of payload. It is generally unnecessary to store all payloads for longer traffic flows.

The collected traffic data were first identified using application signature via precise signature matching for the payload. The signatures were represented using regular expressions. For example, the signature for SMTP protocol is ^220[\x09-\x0d -~]* (e?smtp|simple mail), it can match the strings as,

220 mail.stalker.com ESMTP
CommuniGate Pro 4.1.3
220 mail.vieodata.com ESMTP Merak
6.1.0; Mon, 15 Sep 2003
13:48:11 -0400
220 mail.ut.caldera.com ESMTP
220 persephone.pmail.gen.nz ESMTP
server ready.

Regular expressions of signatures of 90 popular applications were selected to match the collected payload (Sourceforge 2006). The signature matching method identified approximately 70% of the traffic flows and most of

which are TCP flows. The identified flows are listed in Table 2. The reason that only 70% of traffic flows are identified is two folds. First, the signature set is relatively old (Sourceforge 2006), therefore, many newer applications or variants are not able to be identified. Second, some flows are incomplete without complete signature.

Table 2 shows that the majority of the traffic flows are *WWW* and *service* flows. Three classes *Game, Multimedia, Attack* had too few flows in the data, and are therefore excluded from the data set. This is because the Skype voice traffic is encrypted and there is no known attack present during the data collection period. Only one game flow (xboxlive) was identified in the data. Therefore, in subsequent study, we focused on the seven traffic classes that have sufficient number of samples that will make it statistically significant.

For each bidirectional flows, 19 parameters are computed from the packet headers to be the discriminators for the classification algorithms. These parameters are all obtainable in real time from the packet header without storing the packet. For both UDP and TCP flows, these include the known fields in the packet header, as well as the average and variance of the packets sizes. Discriminators 15~19 are applicable to TCP flows only and are set to 0 for UDP flows.

4 Classification method

4.1 SVM

Support Vector Machine (SVM), based on statistical learning theory, is known as one of the best machine learning algorithms for classification purpose and has been successfully applied to many classification problems such as image recognition, text categorization, medical diagnosis, remote sensing, and motion classification (Bazi and Melgani 2006; Bellotti and Crook 2008; Burges 1998; Huang et al. 2008; Liu et al. 2008; Shon and Moon 2007; Yan et al. 2008). SVM method is selected as our classification algorithm due to its ability for simultaneously minimizing the empirical classification error and maximizing the geometric margin classification space. These properties reduce the structural risk of over-learning with limited samples.

Selecting different kernel is an important aspect in the SVM-based classification, commonly used kernel functions include LINEAR, POLY, RBF and SIGMOID. Different kernel functions create different non-linear separation surfaces. An important parameter in SVM is the penalty parameter C, which represents the degree of punishment and has an impact on experiment results.

The effects of different kernel functions and penalty parameters on the classification accuracy have been studied and the results are discussed in Section 5.

Table 2 Data set for network flow experiment

Traffic class	Application	Number of flows
Bulk	ftp, xunlei	14,111
Interactive	telnet, irc, jabber	71
Mail	smtp, pop3	2,245
Service	Whois, dns	16,006
WWW	http, https	48,827
P2P	bittorrent, qq, edonkey, Skype, etc.	10,546
Others	netbios, ntp, tsp, smb, etc	61,832

4.2 Cross validation

In supervised machine learning, if the training samples and the testing samples are the same, the accuracy can be made artificially high. Therefore, it is very important for these two types of samples to be different. We adopt the cross-validation method to evaluate the accuracies of our experiment (Kohavi 1995).

In *n-fold* cross-validation scheme, samples are divided into n subsets of equal size. Sequentially, each subset is tested using the classifier trained on the remaining n-1 subsets. Thus, each instance of the whole training set is tested once and the overall cross-validation accuracy is the average across the entire data set. The prediction accuracy obtained by cross-validation is able to reflect the performance as classifying unknown data more precisely.

In general, the value of n does not affect the cross-validation accuracy much if it is small when compared to the number of samples in the entire data set. Therefore, in the Internet traffic classification experiment, $n=10$.

4.3 Discriminator selection algorithms

In general, different features will have different effects on the classification accuracy. Some features may have greater positive effects on the classification, while others maybe have smaller effects. Moreover, some features may even have negative impacts. Therefore, we must carefully choose the best combination of features for SVM discriminators to optimize classification. The main discriminator selection methods are described in the following.

4.3.1 Optimum searching method

Every possible combination of discriminators is to be evaluated, and the combination that yields the best result is selected. In this study, with 19 features included in the experiment, a total of 524,287 combinations need to be evaluated. Obviously it would be too much to evaluate them all.

Table 3 Network flow features

ID	The flow discriminator
1	Total number of packets in the flow
2	The average packets size of a flow
3	The number of packets sent for the flow
4	The average send packets size of a flow
5	The variance of send packets' size
6	The average receive packets size of a flow
7	The variance of receive packets' size
8	The variance of received packets' size
9	The duration of the flow
10	The protocol (TCP or UDP)
11	The source port of a flow
12	The destination port of a flow
13	The number ratio of send and receive packets
14	The byte ratio of send and receive packets
15	The number of SYN packets
16	The number of RST packets (rst)
17	The number of FIN packets (fin)
18	The average window size (window_size)
19	The variance of window size

4.3.2 Hypo-optimum searching method

This method can produce good results with much less calculation. Two main algorithms are:

a) Sequential forward selection. Begin with zero features chosen, sequentially append one feature that can yield the best classification result to the chosen features. Successively performing this task and finally selecting the combination with the best classification accuracy.
b) Plus-*m*-minus-*r* algorithm. This is the expansion of sequential forward selection algorithm. Begin with zero features chosen, sequentially append m features to chosen ones and pop *r* features from them ($m>r$), select the feature set that yields the best classification result. Successively performing this task and finally selecting the combination with the best classification accuracy. The sequential forward selection can be seen as plus-1-minus-0 algorithm.

Both the sequential forward selection and plus-*m*-minus-*r* algorithms have been evaluated in the experiment. The results are presented in Section 5.

Table 4 Testing different kernel functions

Kernel function	Total accuracy (%)
LINEAR	87.10
POLY	90.83
RBF	93.38
SIGMOID	15.92

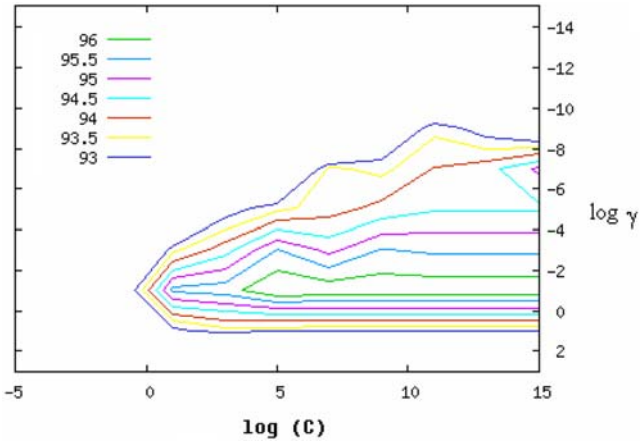

Fig. 1 Effect of penalty parameter and RBF function on classification accuracy

5 Classification results and analysis

5.1 SVM model tuning

The 19 parameters in Table 3 are of different types and can take very different values. To make the discriminators suitable to the SVM algorithm, these parameters are pre-treated using logarithm function so that they all distribute in the same value range between 0 and 1.

To evaluate the effects of kernel functions, four commonly used kernel functions as LINEAR, POLY, RBF, SIGMOID are used. Two hundred samples were selected from each traffic class and all 19 features are used for the discriminators to help the SVM method evaluate their classification accuracy. Cross-validation results as $n=10$ are shown in Table 4.

Clearly, RBF kernel function gives the best classification accuracy. Therefore, RBF kernel function was used in the subsequent experiment.

Figure 1 shows the classification accuracy contour map for different penalty parameter C and the RBF parameter γ.

The figure shows that the classification accuracy is quite stable over a wide range of penalty parameter *C* when the

Table 5 Classification accuracy for each class under the biased-prior-probabilities condition

Traffic class	False negative (%)	False positive (%)
Bulk traffic	1.02	1.79
Interactive	25.49	4.23
WWW	0.20	0.36
Service	0.00	0.45
P2P	1.53	0.84
Mail	13.99	3.10
Other	0.89	0.70

Table 6 Classification accuracy for each class under unbiased-prior-probabilities condition

Traffic class	False negative (%)	False positive (%)
Bulk traffic	6.00	9.00
Interactive	8.68	7.04
WWW	7.00	7.00
Service	5.00	8.00
P2P	5.50	3.00
Mail	5.78	4.62
Other	6.50	7.50

RBF kernel function parameter γ takes the value around 1. Therefore, γ=1/2 and $C=2^{10}$=1,024 were set.

5.2 Comparison between biased and un-biased training samples

As mentioned earlier, in many previous studies, the number of training and testing samples in each different application depends upon the actual ratio in the network. At first glance, this approach seems reasonable. However, since the number of samples for some applications may be much larger than the others, the classification result will be heavily biased towards the classes with more training samples, while the class with few samples may exhibit higher error rate during classification. This bias will not affect the total classification accuracy, and in fact, will actually increase the accuracy, as it can correctly classify more samples. However, this increase of accuracy is at the cost of misclassifying underrepresented applications. For a classification algorithm to perform well, sometimes it is desirable to have relatively uniform accuracy across a range of applications. Therefore, we performed two experiments. First, we choose the training samples according to their actual ratio in the overall data set. Second, we choose same number of sample within each traffic class (200). The classification results are shown in Table 5 and Table 6. Again, all 19 discriminators are used, *C=1024* and RBF kernel function with γ=1/2 was applied.

The experiment results in Table 5 are obtained with the following data set: bulk traffic: 2,237 flows; interactive: 71 flows; WWW: 25,864 flows; service: 7,348 flows; P2P:

Table 7 Discriminator optimization by sequential forward selection method

Selected feature set	Accuracy (%)
12	81.75
12, 7	90.91
12, 7, 18	92.67
12, 7, 18, 11	94.84
12, 7, 18, 11, 14	94.44
12, 7, 18, 11, 14, 13	94.17
12, 7, 18, 11, 14, 13, 1	94.57
12, 7, 18, 11, 14, 13, 1,	94.71
12, 7, 18, 11, 14, 13,,1 10, 2	95.98
12, 7, 18, 11, 14, 13, 1, 10, 2, 6	93.89
12, 7, 18, 11, 14, 13, 1, 10, 2, 6, 8	93.62
12, 7, 18, 11, 14, 13, 1, 10, 2, 6, 8, 3	93.76
12, 7, 18, 11, 14, 13, 1, 10, 2, 6, 8, 3, 15	93.62
12, 7, 18, 11, 14, 13, 1, 10, 2, 6, 8, 3, 15, 17	93.35
12, 7, 18, 11, 14, 13, 1, 10, 2, 6, 8, 3, 15, 17, 16	94.17
12, 7, 18, 11, 14, 13, 1, 10, 2, 6, 8, 3, 15, 17, 16, 9	93.08
12, 7, 18, 11, 14, 13, 1, 10, 2, 6, 8, 3, 15, 17, 16, 9, 19	93.35
12, 7, 18, 11, 14, 13, 1, 10, 2, 6, 8, 3, 15, 17, 16, 9, 19, 4	93.36
12, 7, 18, 11, 14, 13, 1, 10, 1, 5, 7, 2, 14, 16, 16, 9, 19, 4, 5	92.81
Best sequence: 12, 7, 18, 11, 14, 13, 1, 10, 2	95.98

Table 8 Discriminator optimization by plus-2-minus-1 method

Selected feature set	Accuracy (%)
12	81.75
12, 7	90.91
12, 2, 14	93.96
12, 14, 5, 9	95.12
12, 14, 5, 9, 15	96.14
12, 14, 5, 9, 15, 2	96.27
12, 14, 5, 9, 15, 18, 8	96.79
12, 14, 5, 9, 15, 8, 10, 2	96.53
12, 14, 5, 9, 15, 8, 10, 2, 7	97.17
12, 14, 5, 9, 15, 8, 10, 2, 7, 3	96.79
12, 14, 5, 9, 15, 8, 10, 2, 7, 6, 16	96.66
12, 14, 5, 9, 15, 8, 10, 2, 7, 6, 4, 13	96.27
12, 5, 9, 15, 8, 10, 2, 7, 6, 4, 13, 11, 16	96.53
12, 5, 9, 15, 8, 10, 2, 7, 6, 4, 13, 11, 16, 17,	96.40
12, 5, 9, 15, 8, 10, 7, 6, 4, 13, 11, 16, 17, 1, 3,	96.27
12, 5, 9, 15, 8, 10, 6, 4, 13, 11, 16, 17, 1, 3, 19, 14,	96.40
12, 5, 9, 15, 8, 10, 6, 4, 13, 11, 16, 17, 1, 3, 19, 18, 7	96.14
12, 5, 9, 15, 8, 10, 6, 4, 13, 16, 17, 1, 3, 19, 18, 7, 2, 14	96.27
12, 5, 9, 15, 8, 10, 6, 4, 13, 16, 17, 1, 3, 19, 18, 7, 2, 14, 11	96.02
Best sequence: 12, 14, 5, 9, 15, 8, 10, 2, 7	97.17

Table 9 Classification accuracy for each class under the biased-prior-probabilities condition with optimized discriminator set

Traffic class	False negative	False positive
Bulk traffic	1.79	0.94
Interactive	10.86	4.23
WWW	0.31	0.50
Service	0.05	0.03
P2P	2.00	2.82
Mail	9.34	2.44
Other	0.26	0.46

2,021 flows; mail: 902 flows; other services: 8,533 flows. The weighted average across all the traffic class is 99.41% which is a large number. However, the false negative and false positive ratios of each class vary significantly. The classes with more training samples, such as WWW and service, have low false negative ratios (0.20% and 0.00% respectively), while the false negative ratios for the classes with fewer training samples as interactive and mail is 25.49% and 13.99% respectively. Although it is unacceptable for these classes, it actually did not affect the total accuracy.

The experiment result shown in Table 6 is obtained with an unbiased sample set in which the number of samples within all traffic classes is set as 200. The 200 flows are randomly chosen from all data samples. The weighted average across all traffic classes is 92.81%; and Table 6 shows the false negative and false positive ratios of each class.

Although the total accuracy is not as high as the results with biased sample, the false negative and false positive ratios are nearly the same in each class, which reflect the different network characteristics among different applications. Therefore, we used unbiased prior probabilities samples in our subsequent experiments.

5.3 Optimizing the discriminator set

In order to find the best discriminator set from the available 19 parameters for SVM classification, we first used the sequential forward selection method. The result is shown in Table 7. For convenience, we only use the feature ID to represent the features in Table 3. The best discriminator set obtained is {12, 7, 18, 11, 14, 13, 1, 10, 2}, and it yielded a classification accuracy of 95.98%.

The sequential forward selection method has its own shortcomings as if a feature is selected and included in the chosen set, it cannot be subtracted from the set even if it may be unsuitable when other features are added; hence the classification may be trapped into a local optimization. Therefore, we adopt a plus-2-minus-1 strategy for the next experiment. The result is shown in Table 8.

Table 8 shows that some unsuitable features have been subtracted from the set even if such features may be included in previous iterations. It is easy to locate the best sequence as {12, 14, 5, 9, 15, 8, 10, 2, 7}, with an accuracy of 97.17%, increased slightly over 1%.

In order to evaluate whether the optimized discriminator set is also applicable to both biased sample scenario, we used the same feature set {12, 14, 5, 9, 15, 8, 10, 2, 7} to classify the traffic for the samples with biased prior probabilities. The result is shown in Table 9.

The weighted average of classification accuracy across all traffic classes is 99.42%. It is slightly better than the previous result with a total of 19 discriminators. In addition, compared to Table 5, the false negative values of other under represented classes decreased down to a more acceptable level. Although the worst classes are still Interactive and Mail due to fewer training samples, their false negative values are now 10.86% and 9.34% respectively, better than those in Table 5 (25.49% and 13.99%).

6 Summary and future work

The SVM-based method developed in this paper classifies the Internet traffic into broad traffic classes. Each class contains application that shares common network behavior. Experiments were carried out using traffic samples collected from a real campus backbone. The results show that using flow parameters obtained from packet header only can produce highly accurate classification. An accuracy of 99.42% is achieved with the regular biased training and testing samples according to their traffic mix in the real network environment. An accuracy of 97.17% is achieved even in the unbiased sample scenario, where the classification error is evenly distributed across the traffic classes.

The proposed method is also applicable to encrypted network traffic, since it does not rely on the application payload for classification. Furthermore, as all the feature parameters are computable without the storage of multiple packets, the method lends itself well for real-time traffic identification. For the data sets tested, the optimized feature set only contains nine discriminators. The SVM method based on RBF kernel functions is computationally more efficient than the previous methods with similar accuracies.

The fact that the optimized discriminator set is applicable to different traffic mixes is also interesting. We argue that the stability of these discriminators is inherent for the statistical properties of the traffic classes. Thus it could serve to guide our future work for choosing which features to use when classifying new network applications.

One of the disadvantages of SVM-based and other supervised machine learning method is the requirement on a large number of labeled training samples. Moreover, identifying the traffic after the network flow is collected could be too late should security and QoS intervention become necessary in the early stage of the traffic flow. In our future work, we intend to combine the supervised and un-supervised machine learning methods, as well as using feature parameters obtainable early in the traffic flow for fast and accurate Internet traffic classifications.

Acknowledgements The research presented in this paper is supported in part by the NSFC (Grant numbers: 60243001, 60574087, 60605019, 60633020) and 863 High Tech Development Plan (Grant numbers: 2007AA01Z475, 2007AA01Z480, 2007AA01Z464).

References

Bazi, Y., & Melgani, F. (2006). Toward an optimal SVM classification system for hyperspectral remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, *44*(11), 3374–3385.

Beheshti, H., Hultman, M., Jung, M., Opoku, R., & Salehi-Sangari, E. (2007). Electronic supply chain management applications by Swedish SMEs. *Enterprise Information Systems*, *1*(2), 255–268.

Bellotti, T., & Crook, J. (2008). Support vector machines for credit scoring and discovery of significant features. *Expert Systems with Applications*, to appear.

Bernaille, L., Teixeira, R., Akodkenou, I., Soule, A., & Salamatian, K. (2006). Traffic classification on the fly. *Computer Communication Review*, *36*(2), 23–26.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*, 121–167.

Duan, L., Xu, L., Guo, F., Lee, J., & Yan, B. (2007). A local-density based spatial clustering algorithm with noise. *Information Systems*, *32*(7), 978–986.

Duan, L., Xu, L., Liu, Y., & Lee, J. (2008). Cluster-based outlier detection. *Annals of Operations Research*, to appear.

Early, J., Brodley, C., & Rosenberg, C. (2003). Behavioral authentication of server flows. *Proceedings of the 19th Annual Computer Security Applications Conference*, pp. 46–55.

Feng, S., Li, H., & Xu, L. (2001). Knowledge-based systems in China. *Knowledge-Based Systems*, *14*, iii–iv.

Guo, J. (2007). Business-to-business electronic market place selection. *Enterprise Information Systems*, *1*(4), 383–419.

Haffner, P., Sen, S., Spatscheck, O., & Wang, D. (2005). ACAS: Automated construction of application signatures. *Proceeding of ACM SIGCOMM 2005 Workshops: Conference on Computer Communications*, 197–202.

Huang, C., Liao, H., & Chen, M. (2008). Prediction model building and feature selection with support vector machines in breast cancer diagnosis. *Expert Systems with Applications*, *34*, 578–587.

Kohavi, R. (1995). A Study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1137–1143.

Lakhina, A., Crovella, M., & Diot, C. (2004). Characterization of network-wide anomalies in traffic flows. *Proceedings of the 2004 ACM SIGCOMM Internet Measurement Conference*, 201–206.

Li, L., Valerdi, R., & Warfield, J. (2008). Advances in enterprise information systems. *Information Systems Frontiers*, to appear.

Li, L., Warfield, J., Guo, S., Guo, W., & Qi, J. (2007a). Advances in intelligent information processing. *Information Systems*, *32*(7), 941–943.

Li, H., & Xu, L. (2001). Feature space theory-a mathematical foundation for data mining. *Knowledge-Based Systems*, *14*(5–6), 253–257.

Li, W., Zheng, W., & Guan, X. (2007b). Application controlled caching for web servers. *Enterprise Information Systems*, *1*(2), 161–175.

Liu, R., Wang, Y., Baba, T., Masumoto, D., & Nagata, S. (2008). SVM-based active feedback in image retrieval using clustering and unlabeled data. *Pattern Recognition*, *41*, 2645–2655.

Luo, J., Xu, L., Jamont, J. P., Zeng, L., & Shi, Z. (2007). A flood decision support system on agent grid: method and implementation. *Enterprise Information Systems*, *1*(1), 49–68.

Moore, A., & Zuev, D. (2005a). Internet traffic classification using Bayesian analysis techniques. *Performance Evaluation Review*, *33*, 50–60.

Moore, A., & Zuev, D. (2005b). *Discriminators for use in flow-based classification*. Cambridge: Technical Report, Intel Research.

Roughan, M., Sen, S., Spatscheck, O., & Duffield, N. (2004). Class-of-service mapping for QoS: A statistical signature-based approach to IP traffic classification. *Proceedings of the 2004 ACM SIGCOMM Internet Measurement Conference*, 135–148.

Sen, S., Spatscheck, O., & Wang, D. (2004). Accurate, scalable in-network identification of P2P traffic using application signatures. *Thirteenth International World Wide Web Conference Proceedings*, 512–521.

Shi, Z., Huang, Y., He, Q., Xu, L., Liu, S., Qin, L., et al. (2007). MSMiner-a developing platform for OLAP. *Decision Support Systems*, *42*(4), 2016–2028.

Shi, S., Xu, L., & Liu, B. (1996). Application of artificial neural networks to the nonlinear combined forecasts. *Expert Systems*, *13*(3), 195–201.

Shi, S., Xu, L., & Liu, B. (1999). Improving the accuracy of nonlinear combined forecasting using neural networks. *Expert Systems With Applications*, *16*(1), 49–54.

Shon, T., & Moon, J. (2007). A hybrid machine learning approach to network anomaly detection. *Information Sciences*, *177*, 3799–3821.

Sourceforge Application Layer Packet Classifier for Linux (2006). Application Layer Packet Classifier for Linux. Retrieved in 2006, from http://l7-filter.sourceforge.net.

Vigna, G., Robertson, W., & Balzarotti, D. (2004). Testing network-based intrusion detection signatures using mutant exploits. *Proceedings of the 11th ACM Conference on Computer and Communications Security*, 21–30.

Wang, S., & Archer, N. (2007). Electronic marketplace definition and classification: literature review and clarification. *Enterprise Information Systems*, *1*(1), 89–112.

Xu, L. (1999). Artificial intelligence applications in China. *Expert Systems with Applications*, *16*(1), 1–2.

Xu, L. (2006). Advances in intelligent information processing. *Expert Systems*, *23*(5), 249–250.

Yan, Z., Wang, Z., & Xie, H. (2008). The application of mutual information-based feature selection and fuzzy LS-SVM-based classifier in motion classification. *Computer Methods and Programs in Biomedicine*, *90*, 275–284.

Ruixi Yuan received his Ph.D. degree in Electrical Engineering from Texas A&M University in 1991 and his B.S. in Physics from University of Science and Technology of China in 1985. From 1991 to 2004, he was employed in the high tech industry in the USA and conducted research and development in the communication and networking sector. Currently he is a Professor at the Center for Intelligent and Networked Systems, the Department of Automation of Tsinghua University, China. His research interest includes Internet traffic analysis, P2P networking, streaming media, and wireless communications.

Zhu Li received his B.S. and M.S. from the Department of Automation of Tsinghua University, China in 2005 and 2007 respectively.

Xiaohong Guan received his B.S. and M.S. degrees in Control Engineering from Tsinghua University, Beijing, China, in 1982 and 1985, respectively, and his Ph.D. degree in Electrical Engineering from the University of Connecticut in 1993. He was a consulting engineer with PG&E from 1993 to 1995. From 1985 to 1988 and since 1995 he has been with the Systems Engineering Institute at Xi'an Jiaotong University, Xi'an, China, and currently he is the Cheung Kong Professor of Systems Engineering and Director of the National Lab for Manufacturing Systems. He is also the Chair of the Department of Automation and Director of the Center for Intelligent and Networked Systems, Tsinghua University, China. He visited the Division of Engineering and Applied Science, Harvard University from Jan 1999 to Feb 2000. His research interests include computer network security, and economics and security of networked systems.