Received: 10 April 2015 | Revised: 3 November 2016 | Accepted: 17 November 2016
DOI 10.1002/nem.1962

WILEY

RESEARCH ARTICLE

An accurate traffic classification model based on support vector machines

Jie Cao[1,2] | **Zhiyi Fang**[1] | **Guannan Qu**[1] | **Hongyu Sun**[1,3] | **Dan Zhang**[1]

[1] College of Computer Science and Technology, Jilin University, Changchun, China
[2] School of Information Engineering, Northeast Electric Power University, Jilin, China
[3] College of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore, Maryland, USA

Correspondence
Jie Cao, College of Computer Science and Technology, Jilin University, Changchun, China.
Email: caojiell78@126.com

Summary
Network traffic classification is a fundamental research topic on high-performance network protocol design and network operation management. Compared with other state-of-the-art studies done on the network traffic classification, machine learning (ML) methods are more flexible and intelligent, which can automatically search for and describe useful structural patterns in a supplied traffic dataset. As a typical ML method, support vector machines (SVMs) based on statistical theory has high classification accuracy and stability. However, the performance of SVM classifier can be severely affected by the data scale, feature dimension, and parameters of the classifier. In this paper, a real-time accurate SVM training model named SPP-SVM is proposed. An SPP-SVM is deducted from the scaling dataset and employs principal component analysis (PCA) to extract data features and verify its relevant traffic features obtained from PCA. By employing PCA algorithm to do the dimension extraction, SPP-SVM confirms the critical component features, reduces the redundancy among them, and lowers the original feature dimension so as to reduce the over fitting and increase its generalization effectively. The optimal working parameters of kernel function used in SPP-SVM are derived automatically from improved particle swarm optimization algorithm, which will optimize the global solution and make its inertia weight coefficient adaptive without searching for the parameters in a wide range, traversing all the parameter points in the grid and adjusting steps gradually. The performance of its two- and multi-class classifiers is proved over 2 sets of traffic traces, coming from different topological points on the Internet. Experiments show that the SPP-SVM's two- and multi-class classifiers are superior to the typical supervised ML algorithms and performs significantly better than traditional SVM in classification accuracy, dimension, and elapsed time.

1 | INTRODUCTION

Network traffic classification is defined as a classification of the network flows that are a mixture of various applications with different application protocols.[1] It is the foundation of high-performance network protocol design and network operation management. Four traditional methods,[2–4] which are port numbers, deep packet inspection (DPI),[5,6] protocol analysis, and machine learning techniques, based on current traffic classification research suffer from different practical issues. The port numbers–based method runs faster but is sensitive to ports confusion. The DPI-based method is mature, but it is difficult to obtain mode characteristics. The protocol analysis–based method is more precise but suffers high reverse cost. The machine learning technique has become popular recently since it can automatically search for and describe useful structural patterns in a supplied traffic dataset[7,8] that makes the traffic classification more flexible and intelligent.

Machine learning methods of traffic classification consist of unsupervised methods and supervised methods. The unsupervised learning methods cluster dataset samples according to their similar characteristics instead of prelabeling training data. As a typical clustering algorithm representative of the unsupervised traffic classification, K-means is currently used widely.[9,10] The drawback of the unsupervised traffic classification is that it is hard to construct an application-oriented traffic classifier by using the clustering results without knowing the real traffic classes.[8] The supervised learning methods, such as Naive Bayes,[11] support vector machine (SVM),[12,13] Bayesian networks,[14] k-nearest neighbor (k-NN),[8] C4.5 decision tree,[15] and neural networks,[16] etc, require empirical knowledge (also known as prelabeled training data) to train the classification model and parameters. As the top of the 7 supervised machine learning algorithms reported in 1 study,[17] in which certain pattern recognition methods such as Bayesian networks, C4.5 decision tree, and k-NN, may be trapped into local optimization; SVM maximizes the optimization margin and is capable to solve high-dimensional nonlinear problem. Meanwhile, SVM starts being used in traffic classification to get traffic flow parameters from packet headers[13] and reduce the training set and approximate support vectors.[18] Support vector machine ensemble has been constructed by incorporating bagging, boosting, or cross-validated committee (CVC) characteristics.[19] The optimization algorithm has been proposed to solve multi-class problems.[12]

Even though, there are still some accuracy and real-time traffic classification issues left[18–21] because of the rapid growth of network traffic and the development of backbone network architecture. How to improve the performance of SVM training model in real-time classification with a few samples? What kind of kernel should we use? What kind of functions are valid kernels? How to derive kernel parameters? Etc. All of these above questions are under investigations. This paper will focus on solving multi-class problem, selecting kernel function, reducing dimension of the flow features, and deriving the optimal parameters. The details are as follows:

1. Analyzed applications of two-class SVM approach of traffic classification and integrated SVM "one-against-one" approach to solve multi-class problems when needed,
2. Using principal component analysis (PCA) algorithm for feature extraction and clarifying the relevant traffic features obtained from PCA,
3. Proposed an approach employing improved particle swarm optimization (PSO) algorithm to search for the optimal working parameters of kernel function automatically, and
4. Comparing with traditional SVM and the representative-supervised machine learning algorithm to approve that the proposed model has improved the traffic classification significantly with only few training samples.

In section 2, the related work done on traffic classification has been reviewed. Section 3 describes the proposed methods and framework. Section 4 proposes experimental framework and evaluates the performance of the proposed model. Section 5 discusses the proposed model and its building blocks in depth. Section 6 concludes the work.

2 | RELATED WORK

2.1 | Based on port numbers

Traditional methods based on port numbers traffic classifiers simply inspect TCP or UDP port numbers and identify the application layer protocols according to the Internet Assigned Numbers Authority (IANA) list of well-known ports and registered ports.[9,22] The method was simple and fast in the past. However, it has become obsolete nowadays. The mapping between the ports and the target applications is getting more and more blurred. Thereby port numbers as a classification mechanism has not been applied, and it is difficult to deploy.[23,24]

2.2 | Based on deep packet inspection

Deep packet inspection methods, usually the most accurate, are based on inspection of the packets' payload. They rely on a database of previously known signatures that are associated to application protocols and search each packet for strings that match any of the signatures.[1,5,6] Searching feature string by DPI is generally in the application layer, which is the load of TCP or UDP. Nevertheless, the main drawbacks of DPI techniques are the following: (1) There are more and more nonstandard applications and private protocols without the open and available protocol specification. This makes the feature string vary and hard to find. (2) Protocol syntax or semantic analysis of data needs strong computation power, leading to a great system overhead.[25] So today, DPI is generally used in traffic identification of the specific applications or as a supplementary means of tagging the network dataset.

2.3 | Based on protocol analysis

Based on open protocol regulations, protocol analysis methods analyze the protocols using the following 3 ways for traffic classification: (1) establishing protocol state machines, (2) using fingerprint (protocol traffic features or behavior features) mining, and (3) analyzing flow features and behavior features of unknown protocols using software conversation approach.

However, it is very difficult to resolve and obtain effective features because of the nonstandard applications and encryption protocols, degrading the classification quality.

2.4 | Based on machine learning techniques

The machine learning methods capture and identify the traffic data packets on the basis of calculating the statistical information of the specific application traffic. The methods use various machine learning algorithms, including supervised and unsupervised learning algorithms. Supervised learning builds a classification model from a training set of labeled instances, which is then used to classify unknown instances. Alternatively, unsupervised learning groups instances that have similar characteristics into natural clusters without any prior guidance, and these clusters can be transformed into a classification model.[9] The machine learning method usually includes 3 aspects: statistical feature extraction, classifiers building and training, and new traffic classification. Kim et al[17] evaluated ports-based CoralReef method, host behavior–based BLINC method, and 7 supervised machine learning methods. Support vector machine was the best, followed by neural network, k-NN, Bayes networks, Naive Bayes kernel estimation, Naive Bayes, and C4.5. Williams et al[26] evaluated multi-machine learning methods. According to the overall accuracy, followed in descending order is Bayesian networks, C4.5, AdaBoost C4.5, NBTree, AdaBoost NBT, and k-NN. Alshammari et al[27] used different machine learning algorithms (C4.5, Naive Bayes, and SVM) to identify 13 signatures and 14 attributes of secure shell traffic. From the above research point of view, the machine learning methods can automatically search for and describe useful structural patterns in a supplied traffic dataset, which is helpful to intelligently conduct traffic classification. However, the machine learning methods still face some challenges. Such as the high computational cost, the high-dimensional feature makes the algorithms can not adapt real-time traffic classification and require adjusting various classifiers' parameters frequently etc. Our main focus is to optimize the performance of supervised machine learning algorithms—SVM, so that it can adapt accurate and real-time traffic classification.

3 | PROPOSED METHODS OF SPP-SVM

3.1 | The SPP-SVM model's framework

The SPP-SVM model's framework is shown in Figure 1. The model mainly includes the procedure of scaling, feature extraction, and parameters optimization. We implement the model through the following steps. (1) We processed the original dataset using unbiased samples in our experiments. This guaranteed that the false negative and false positive ratios are nearly the same in each class. These samples reflect the different network characteristics among different applications. (2) The model performance will be affected if the data is not scaled right. To minimize the data scale effect and improve the speed of classification, we rescale the data. (3) To separate training and test sets, we have chosen randomly 50% of unbiased data to form a training set, and the remaining 50% is used as a test set. (4) We use the Andrew Moore datasets,[28] more than 240 features of a single flow, which require high computation complexity. To adapt real-time applications, we extracted the features based on PCA algorithm[29,30] to reduce the feature dimension. (5) The kernel type, kernel parameters, and penalty factor C control the generalization of SVM. However, the computing overhead will increase sharply for optimizing the parameters with the growing dataset. We automatically derive the optimal parameters of the model based on improved PSO, which can increase the accuracy of classification. (6) We use cross-validation method to train and generate the SPP-SVM classifier model. (7) We classify the test set and get the classification results based on the SPP-SVM classifier model. The methods are detailed in sections 3.2 to 3.6.

3.2 | Classifier of SPP-SVM

3.2.1 | Two-class SVM

Classification usually involves with training and testing data, which consist of some data instances. Each instance in the training set contains one "target value" and several "attributes." The goal of SVM is to produce a model to predict

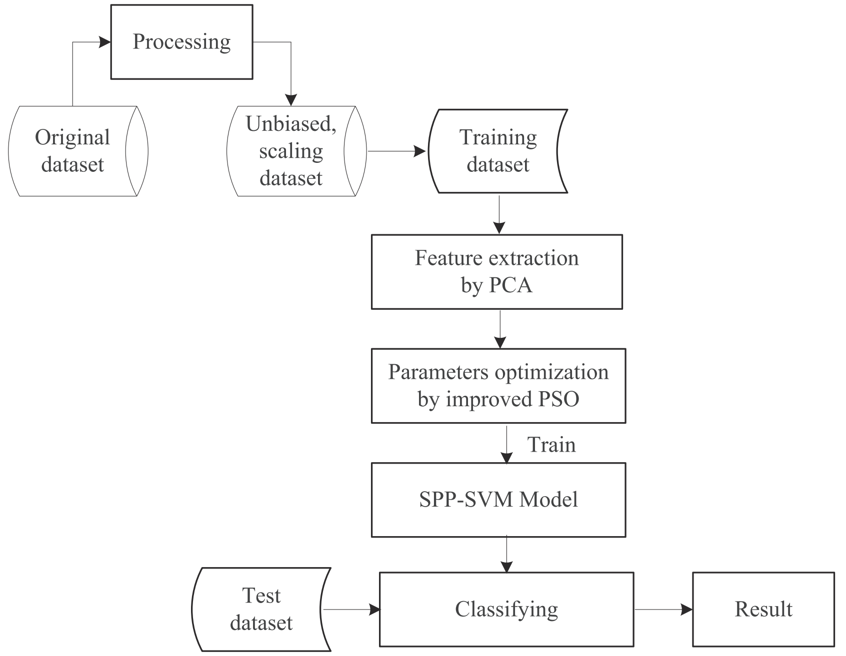


FIGURE 1 Framework of SPP-SVM model. PCA, principal component analysis; PSO, particle swarm optimization

target value of data instances in the testing set, which has only the attributes.[31]

Given a training set of instance-label pairs (x_i, y_i), $i=1,\cdots,l$, where $x_i \in R^n$ and $y \in \{1,-1\}^l$, the function $\phi(x)$ is defined as the mapping of the space to a high-dimensional feature space Z, w is the weight vector, and b is the threshold. The optimal classification plane is generalized, and the relaxation factor $\xi_i \ge 0$ is introduced, which makes the sample satisfy Equation 1. The SVMs require the solution of the following optimization problem as in Equation 2.

$$y_i[w \cdot x_i + \mathrm{b}] \ge 1-\xi_i, \quad i = 1, \cdots, l, \tag{1}$$

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C\sum_{i=1}^{l}\xi_i$$
$$s.t.\begin{cases} y_i\left[w^T\phi(x_i) + \mathrm{b}\right] \ge 1-\xi_i, & i = 1, \cdots, l \\ \xi_i \ge 0, & i = 1, \cdots, l \end{cases}. \tag{2}$$

To solve the original problem, we introduced the dual problem of Equation 2, such as in Equation 3. The inner product kernel function is instead of $K(x_i, x_j) = \phi(x_i)\phi(x_j)$. The final classification discriminant function is $f(x) = \sum_{i=1}^{l} \alpha_i y_j K(x_i \cdot x) + b$. Among them, α_i is the Lagrange multiplier, and the corresponding training sample point is the support vector when α_i is not 0. K is a kernel function to solve nonlinear problems.[32,33] C is a penalty coefficient, which is used to control the error during the training.

$$\min \sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j \phi(x_i)\phi\left(x_j\right) - \sum_{i=1}^{l}\alpha_i$$
$$s.t.\begin{cases} \sum_{i=1}^{l} y_i \alpha_i = 0 \\ 0 \le \alpha_i \le C, \quad i = 1, \cdots l \end{cases}. \tag{3}$$

3.2.2 | Multi-class SVM

We integrate SVM one-against-one approach to solve multi-class problems. About k-class data, one-against-one–based SVM classifier builds $k(k-1)/2$classifiers. Each classifier only trains two-class data. The data of i- and j-classes build the classifier c_{ij}, which is required to solve the quadratic optimization problems shown as in Equation 4 below.

$$\begin{cases} \min_{w^{i,j}, b^{i,j}, \xi^{i,j}} \frac{1}{2}\left(w^{i,j}\right)^T w^{i,j} + C\sum_{l}^{n} {\xi_i}^{i,j} \\ \left(w^{i,j}\right)^T \phi(x_i) + b^{i,j} \ge 1 - {\xi_i}^{i,j}, y_l = i \\ \left(w^{i,j}\right)^T \phi(x_i) + b^{i,j} \le -1 + {\xi_i}^{i,j}, y_l = j \\ {\xi_i}^{i,j} \ge 0 \end{cases}. \tag{4}$$

After the $k(k-1)/2$ classifiers are built, we used the "voting" strategy to predict the unknown sample. Set the initial value of the votes to 0, for example x, subsequently make judgment on $k(k-1)/2$ numbers of the decision functions—$sign((w^{i,j})^T\phi(x) + b^{i,j})$. If classifier c_{ij} determines x to belong to category i, then 1 vote is added to category i. If the classifier determines x to belong to category j, then 1 vote is added to category j. The higher votes will determine which category sample x belongs to, after $k(k-1)/2$ numbers of classifiers are verified. The same rule applies to "one-against-multiple." If more than 1 categories get high votes, one of the categories will be picked randomly, or the decision will be rejected. In reality, solving the problem is also to solve the dual-problem of the original quadratic optimization problem. Every variable of each dual-problem is a sample size of 2 types of categories. Therefore, if the number of input sample size is "n" and totally "k" types of categories, then the entire problem is to solve $k(k-1)/2$ quadratic optimization problems. On average, each problem has 2n/k variables.

3.3 | Kernel function

Kernel function can map the linearly inseparable samples from the lower dimensional space to the higher one. After mapping, the samples are linear and separable. Then in the high-dimensional space, a classification plane can be built to separate the 2 types of samples evenly. The function that meets the conditions of the Mercer function can be considered as the kernel function of SVM. The selection of kernel function will affect the classification results of SVM. There are 4 main kernel functions at present[19]:

3.3.1 | Linear: $K(x_i, x_j) = {x_i}^{\mathrm{T}}, x_j$

When the samples are separable in a lower dimensional space, classification can use the linear kernel function directly. But in lower dimensional space, most samples are linearly inseparable.

3.3.2 | Polynomial: $K(x_i, x_j) = (\gamma {x_i}^{\mathrm{T}} x_j + \mathrm{r}]^d, \gamma > 0$

Parameter d represents the number of dimensions of a kernel function. Polynomial kernel function is part of the global kernel functions. Its locality is poor. Even sample points from a distance can impact the classifier.

3.3.3 | Radial basis function (RBF): $K(x_i, x_j) = \exp(-\gamma\, x_i - x_j)^2, \gamma > 0$

Radial basis function kernel has a good classification effect on the near sample points. The locality performance is perfect, while the generalization ability is weakened accordingly when the parameter γ is increased.

3.3.4 | Sigmoid: $K(x_i, x_j) = \tanh(\gamma {x_i}^{\mathrm{T}} x_j + \mathrm{r})$

The sigmoid kernel function needs to satisfy certain conditions when it is used. The kernel has good global convergence and is equivalent to two-layer neural network.

Above 4 main kernel functions, the advantages of the RBF kernel are the following: (1) Radial basis function can be applied in a wider scope. It is not restricted to the number of samples and the dimension of features. (2) The RBF can map a sample to a higher dimension space, and linear kernel function is a special case of RBF. That is, linear kernel function can be replaced by RBF. (3) Compared with the polynomial kernel function, RBF needs to determine fewer parameters, which can affect complexity of functions. In addition, when the order of the polynomial is high, the element value of kernel matrix will tend to infinity or to infinitesimal. Radial basis function can reduce the difficulties of numerical computation. For the reasons given above and the result in experiments 4.2, we choose RBF as SVM kernel function to get better classification performance.

3.4 | Scaling

Attribute data are easily affected by the data scaling. Support vector machines model effectiveness will be affected if the data is not scaled right. To minimize the data scale effect and improve the speed of classification, we rescaled the data. Data scaling means to convert attribute data into a smaller scale range using certain algorithms. It can ensure convenient data processing and speed up the program convergence.

To scale data mapping attributes to a smaller range according to the certain rules, we use "min-max" scaling to linearly scale each attribute to the range of [−1,1], namely, $X_{ij}{}' = \frac{X_{ij} - \min X_{ij}}{\max X_{ij} - \min X_{ij}}$ ($\min X_{ij} < X_{ij} < \max X_{ij}$), which $X_{ij}{}' = -1$, if $X_{ij} = \min X_{ij}$, $X_{ij}{}' = 1$; if $X_{ij} = \max X_{ij}$, X_{ij} is the attribute value for the jth sample from the ith index, $X_{ij}{}'$ is the new attribute value for the jth sample from the ith index, $\min X_{ij}$ is the smallest attribute value of all samples that belong to the ith index, and $\max X_{ij}$ is the largest attribute value of all samples belonging to the ith index. Scaling is the first step of applying SPP-SVM, and the original data will be changed from here. We use the following methods to scale the original data:

1. We scale the original data with each dimension instead of each sample. Because the dimension of each sample is different, scaling each sample will make the sample's attribute of lower magnitude to be 0 when the magnitude was in great disparity. This will cause the loss of the original information. Since the magnitude is the same for the same dimension, we scale the data with each dimension. This can avoid the situation where lowest magnitude of each sample's attributes is always 0.
2. We put the training set and the test set together to scale. If the training set is scaled first (in each dimension), then the scaling mapping is recorded. In this mapping record, the maximum data value of a dimension is n. When a test set is scaled by this mapping, this will lead to a hypothesis that the maximum data value of this test set dimension is not more than n. This hypothesis is not reasonable. So to avoid this problem, we put the training set and the test set together to scale. And the maximum and minimum data values for each dimension are searched from the training set and test set.

As we know from the above, (1) scaling can avoid too large range of some feature values and too small range of others, and (2) scaling effectively avoids the difficulty of numerical calculation because of calculating inner product to compute the kernel function.

3.5 | Feature extraction

Principle component analysis[20,34] is a feature extraction technique,[35] which rotates the original eigenvector coordinate system and selects the maximal variance vector to build a new coordinate system. The original feature sets are mapped to the lower dimensional space to obtain the essential feature of the original samples.

Assume that among the original samples $X = [x_1, \ldots, x_d]^{\mathrm{T}} = [x_{(1)}, \ldots, x_{(\mathrm{n})}]$, the eigenvectors are $X_i = [x_{1\mathrm{i}}, \ldots, x_{ni}]$ ($i = 1, 2, \ldots \mathrm{D}$), and the samples $X_{(i)} = [x_{i1}, \ldots, x_{id}]^{\mathrm{T}}$ ($i = 1, 2, \ldots, n$). If $Y_d = a_{D1}X_1 + a_{D2}X_2 + \ldots a_{DD}X_D$ ($a_{i1}{}^2 + a_{i2}{}^2 + \ldots + a_{iD}{}^2 = 1$), then Y_d is the dth principle component, and its variance is at the maximum on the Y_d direction. Each principle component is independent. To meet the principle component requirements, we need to solve the following optimization problem as in Equation 5:

$$\max a_1{}^T \sum a_1 \quad s.t. \quad a_1{}^T a_1 = 1. \tag{5}$$

After PCA algorithm dimension reduction, the principle component features are statistically irrelevant. The inter-feature redundancy is reduced. Therefore, PCA can solve the curse of dimensionality effectively, reduce over fitting, and increase the generalization of the SPP-SVM model.

3.6 | Parameters optimization for kernel function

The parameters of kernel function have a very important effect on the performance of the SVM classifier. At present, the selection of optimal parameters based on SVM is usually achieved from a large number of experiments. On the basis of SPP-SVM model, we propose an automatic approach to search for the optimal working parameters of kernel function automatically using an improved PSO[36] algorithm. The optimal C and γ are automatically calculated. This method does not need to traverse all the parameter points, and the inertia weight coefficient is adaptive. The method extracts a certain number population from random solutions and ultimately produces SPP-SVM optimal parameters according to the specific rules of operation, shown as the following:

1. Particle swarm initialization. The dimension of the solution space is defined as D. Randomly generate number of n particles, which are defined as $X=\{x_1, \dots, x_i, \dots, x_n\}$. The position vector of the particles is $x_i=(x_{i1}, \dots, x_{ik}, \dots, x_{iD})$, and the velocity vector of the particles is $v_i=(v_{i1}, \dots, v_{ik}, \dots, v_{iD})$.
2. Set the inertia maximum weight as $\omega_{\max}$, minimum weight as $\omega_{\min}$, learning factors as c_1 and c_2, maximum velocity of particles as $v_{\max}$, current evolutional generation as m, and maximum evolutional generation is $m_{\max}$. Initialize individual and global extreme before the first iteration.
3. Calculate accuracy fitness value $f(x_i)$ of each particle in the particle swarm.
4. Compare the particle's individual fitness value with the individual extreme fitness value. If the individual fitness value is better than the individual extreme fitness value, replace the individual extreme fitness value with the individual fitness value.
5. Compare the particle's individual fitness value with the global extreme fitness value in the particle swarm. If the individual fitness value is better than the global extreme fitness value, replace the global extreme fitness value with the individual fitness value.
6. Recalculate the particle position vector x_i and the velocity vector v_i in the particle swarm.
7. If the iteration number of algorithm reaches the max evolutional generation $m_{\max}$, then end the algorithm and output the global optimal solution. Otherwise, the algorithm will jump back to Step 3 and continue the next iteration.

The inertia weight is the most important parameter in the parameters which affect the performance of the PSO algorithm.[37] A larger ω can improve the global search ability, and a smaller ω can improve the local search ability of the algorithm.[38,39] To balance the global search ability and local search ability of PSO, we adopt the nonlinear dynamic inertia weight coefficient ω (Equation 6). We set objective function value (fitness value) of the particle as f, average objective function value of all the particles as f_{avg}, and minimum objective value of all the particles as $f_{\min}$. Parameter ω changes with the objective function value of the particle, so ω is adaptive. When the objective value of each particle tends to be uniform or locally optimal, the inertia weight is increased; when the objective value of each particle is scattered, the inertia weight is decreased. Meanwhile, the objective function value is better than the average objective function value of the particle, which corresponds to the ω is smaller, and the particle is retained. The objective function value is poorer than the average objective function value of the particle, which corresponds to the ω is larger, and makes the particle closer to the better search area, and vice versa.

$$\omega = \begin{cases} \omega_{\min} - \dfrac{(\omega_{\max}-\omega_{\min})\times(f-f_{\min})}{f_{\text{avg}}-f_{\min}}, f \le f_{\text{avg}} \\ \omega_{\max},\ f > f_{\text{avg}} \end{cases}. \quad (6)$$

4 | EXPERIMENTAL PERFORMANCE EVALUATION

4.1 | Datasets

In this paper, we used the Andrew Moore[28] datasets, which consisted of 10 separate subdatasets each from a different period of the 24-hour day. The day trace was split into 10 blocks of approximately 1680 seconds. Each subdataset was represented by a data text file that included tens of thousands of data lines. Each line represented a traffic flow. The information is derived from packet header information.

To reduce the imbalance of the data, we deleted the traffic flows of games and interactive, which the samples were very few. And we extracted samples randomly within 3000 from every subset to build the new datasets. The datasets included 24 897 samples as in Table 1. The generation process of the training set in subsequent experiments is as follows:

The traffic class has been derived using a content-based analysis. The content-based classification process is described in 1 study.[28] We used the LibSVM[40] and converted original data from arff to csv data format. Then, we labeled the data. We labeled 10 traffic classes from WWW to services with integers from 1 to10. For example, WWW's label is 1 in training set. Finally, to separate training set and test set, we have chosen randomly 50% of unbiased data to form a training set, and the remaining 50% is used as a test set.

4.2 | Kernel selection

We transformed the data format into libsvm, took half of the sample datasets as training set, and the rest as test set. We compared the traffic classification performance of 4 different kernel functions based on SVM. Result shows in Table 2. Linear and polynomial kernel's performance are better, but

TABLE 1 Datasets

Traffic class	WWW	Mail	FTP-control	FTP-pasv	Attack	P2P	Database	FTP-data	Multimedia	Services
Representative applications	HTTP and HTTPS	Pop2/3, smtp, and imap	FTP	FTP	worm and virus	Kazaa, BitTorrent, and Gnutella	Postgres, sqlnet, oracle, and ingres	FTP	Voice and video streaming	X11, dns, ident, and ntp
Samples of flows	2999	2999	2990	2989	1793	2391	2943	2997	576	2220

TABLE 2 Classification accuracy (%) of 4 different kernels based on SVM classifiers

Classifier	Two-class SVM WWW	Mail	FTP-control	FTP-pasv	Attack	P2P	Database	FTP-data	Multimedia	Services	Multi-class SVM
Linear	99.9036	87.4839	12.01	30.1816	88.6166	52.0566	44.2561	67.0389	64.8538	77.3618	87.5803
Polynomial	99.7269	95.5174	75.0321	12.0019	86.8011	84.8249	11.8172	99.4457	95.9672	99.0681	73.9557
RBF	87.9579	88.1748	88.0141	87.9981	92.794	90.4804	88.3194	88.3355	97.6944	91.5087	12.4518
Sigmoid	87.9499	82.222	87.982	87.9981	92.794	87.5884	88.1668	87.9579	97.6864	91.0749	12.0341

Abbreviations: RBF, Radial basis function; SVM, support vector machine.

their two-class performance is quite unstable, and some flows' classification accuracy is lower than 15%. Radial basis function kernel's[41] classification accuracy is higher than sigmoid kernel's, and its two-class performance is very stable. Furthermore, the linear kernel cannot handle the case when the relation between class labels and attributes is nonlinear. The polynomial kernel has more hyperparameters than the RBF kernel. The sigmoid kernel behaves like RBF for certain parameters.

All analyses show that the RBF kernel nonlinearly maps the samples into a higher dimensional space. It is suitable for various conditions, and the required parameters are less. Therefore, we use RBF kernel.

4.3 | Scaling

The average accuracy of two-class SVM classifier with RBF kernel is more than 85%, while the accuracy of multi-class is only 12.45%. Scaling of datasets improves the performance of the classifier. We train the datasets with RBF kernel, which parameters are default ($C = 1$ and $\gamma = 0.004$). After scaling, the attribute values mapping is $[-1, 1]$. All traffic flows' accuracy of two-class is more than 94%, and the accuracy of multi-class is improved from 12.45% to 77.76%. Elapsed time is significantly shorten by 83.9% (CPU frequency is 2.27 GHz). Table 3 shows the classifier performance on the original datasets and the scaling datasets.

4.4 | Feature extraction

After scaling, the performance of two-class classifier is better, while the performance of multi-class classifier is not ideal. We propose the PCA method to reduce dimension. We set the threshold (the percentage of original features) at 90%. The two-class classification performance on scaling datasets (24 897 samples) and unbiased small scaling datasets (5000 samples) are shown in Figure 2. The multi-class classification performance on scaling datasets and unbiased small scaling datasets (5000 samples) are shown in Figure 3.

Dimension decreased greatly of two-class after feature extraction on the scaling and unbiased small scaling datasets. All of the classification accuracy of each threshold is over 95%, except P2P which is over 92%. At this time, the accuracy is close to the original dimension. The average accuracy of unbiased small scaling datasets is only 1.5% lesser than the average accuracy of scaling datasets, which is shown in Figure 2. So, the unbiased small scaling datasets can be used to replace the scaling datasets for classification prediction.

The accuracy of multi-class on unbiased small scaling datasets is noticeably higher than on scaling datasets obviously, which is shown in Figure 3. And the performance is the best when threshold is set as 94%. At this time, the feature dimension is reduced to 20 dimension, which is 89.8% lower than the original one. Accuracy reaches 86.48%, which is 8.72% higher than on the scaling datasets. Elapsed time is 92.8% shorter than it on the scaling datasets. Feature extraction on the unbiased small scaling datasets is more suitable for real-time traffic classification.

We further clarified which are the relevant traffic features we have obtained by PCA. That is, we get a feature subset, which features have maximum relevance to the class and minimum redundancy between them. We have identified a key feature subset through the Correlation-Based Feature Selection (CFS) algorithm and the genetic search strategy. Correlation-Based Feature Selection algorithm evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred. We used the default parameters. The population size is 20, the probability of crossover is .6, the number of generations is 20, the probability of mutation is .033, and the random number seed is 1. Then, we obtained an 8 dimension feature subset. The result of feature selection 10-fold cross-validation is shown in Table 4. In the 20 dimension

TABLE 3 Classifier performance on original and scaling data

Classifier	Two-class WWW	Mail	FTP-control	FTP-pasv	Attack	P2P	Database	FTP-data	Multimedia	Services	Multi-class
Accuracy (%) on original data	87.9579	88.1748	88.0141	87.9981	92.794	90.4804	88.3194	88.3355	97.6944	91.5087	12.4518
Accuracy (%) on scaling data	98.7789	98.9878	98.9798	99.6546	97.5016	94.0312	99.036	99.8233	99.2449	98.6102	77.7555

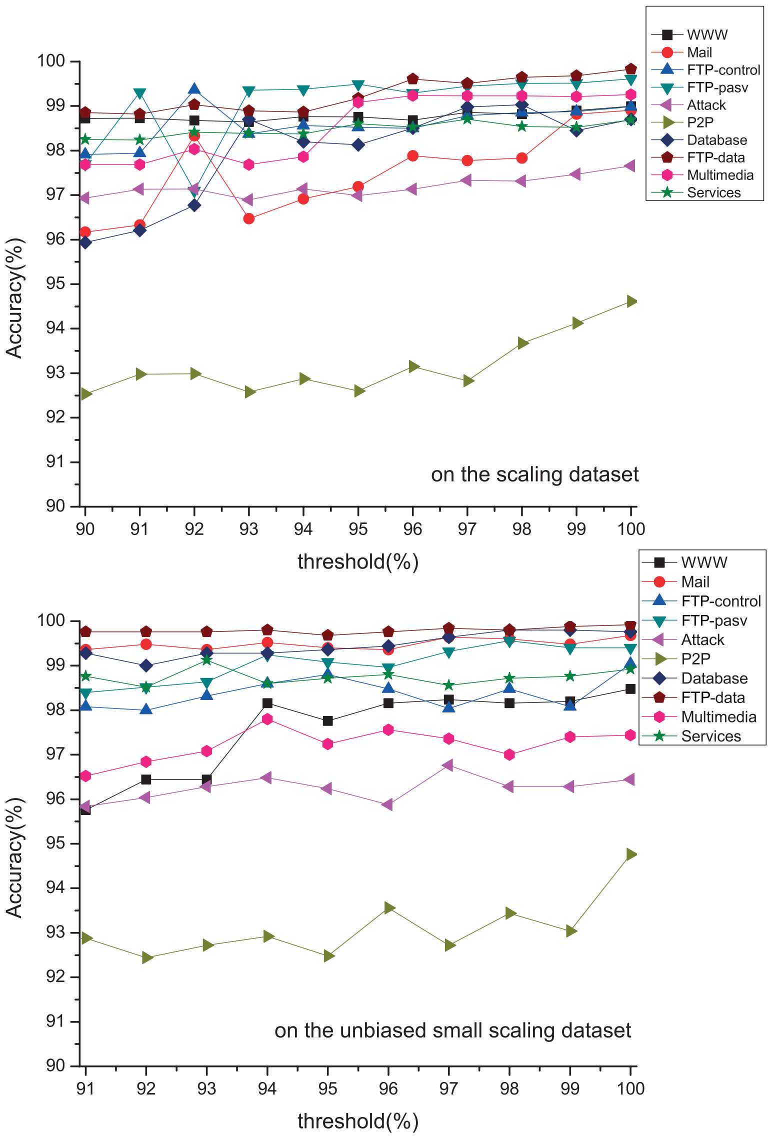


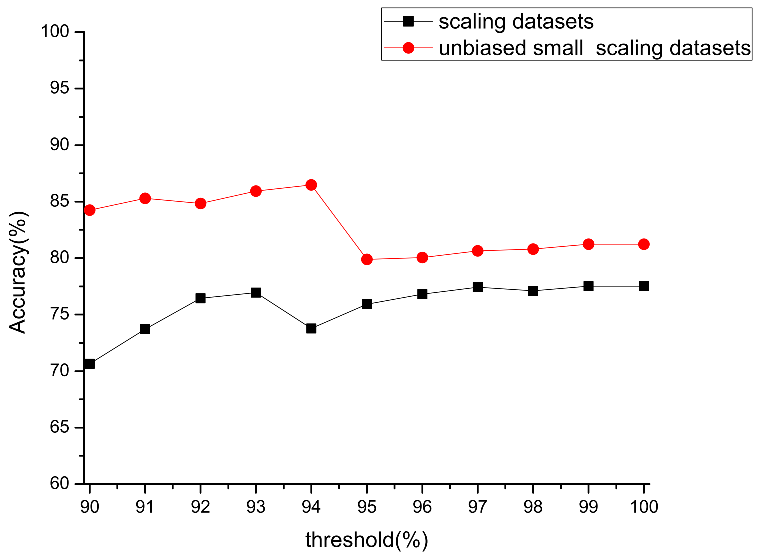
FIGURE 2 Accuracy (%) of two-class classifier using PCA feature extraction on the scaling datasets and unbiased small scaling datasets

features we obtained from PCA, 1, 2, 3, 4, 5, 7, 8, and 11 dimension features have maximum relevance to class and minimum redundancy among them. They are the key feature subset of PCA feature extraction. At this time, classification accuracy based on 8 dimension features subset reaches 83.84%, which is very close by PCA. And the feature dimension is 60% lower than the dimension by PCA feature extraction only. Feature extraction and feature selection can be performed under the specific situation during traffic classification. Feature extraction by PCA can get higher accuracy of classification. Feature selection by CFS after PCA can obtain the key feature subset and the lower feature dimension. Through CFS feature selection, we further clarified which are the relevant traffic features we have obtained by PCA.

4.5 | Parameters optimization

The SVM training model usually can not get the best training result of the default parameters. We derived the optimal working parameters C and γ of kernel function based on improved PSO algorithm. The classifier's performance is better when threshold is set as 94% for the PCA feature extraction. The experiment shows that when threshold is at 94%, the classification performance is also ideal during the parameters optimization by PSO. Partical swarm optimization's initial parameters: $c1$ is 1.5, $c2$ is 1.7, termination iteration is 200, and population is 20. The result of parameters optimization by PSO on threshold 90%-100% is shown in Table 5. The SPP-SVM model includes the procedure of scaling, feature extraction based on PCA, and parameters optimization based on improved PSO.

According to Figure 3 and Table 5, performance is the best when threshold is at 94% of SPP-SVM. As a result, the number of dimensions is 20, the best c is 40.1731, and the best γ is 0.01. To prevent over fitting, we inspected the optimal parameters c and γ. The smaller the value of c, the smaller the penalty of the experience error. That is, the smaller the complexity of the learning machine, the greater the risk value, and vice versa. If c is ∞, it means that all the constraints must be satisfied, which means that the training samples must be classified accurately. When c exceeds a certain value, the complexity of the model reaches the



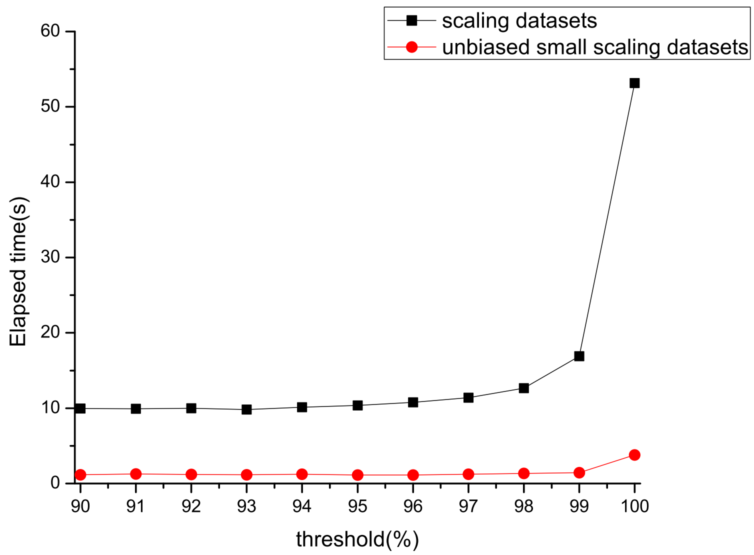


FIGURE 3 Elapsed time and accuracy (%) of multi-class classifier using PCA feature extraction on the scaling datasets and unbiased small scaling datasets

TABLE 4 Feature selection 10-fold cross-validation (stratified)

Feature dimension by PCA	1	2	3	4	5	6	7	8	9	10
Number of folds (%)	9 (90%)	10 (100%)	10 (100%)	10 (100%)	10 (100%)	0 (0%)	9 (90%)	7 (70%)	1 (10%)	0 (0%)
Feature dimension by PCA	11	12	13	14	15	16	17	18	19	20
Number of folds (%)	10 (100%)	1 (10%)	0 (0%)	1 (10%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)

Abbreviation: PCA, principal component analysis.

TABLE 5 Result of parameters optimization by PSO on threshold 90%-100%

Threshold (%)	90	91	92	93	94	95	96	97	98	99	100
Dimension	14	15	16	18	20	23	26	31	39	54	196
Accuracy (%)	91.6	91.76	91.96	91.96	92.12	91.76	91.72	92	92.16	91.92	92.52
c	69.8035	48.39	61.1767	60.7706	40.1731	25.1315	27.4772	30.2797	71.5069	31.9239	22.4321
γ	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01

Abbreviation: PSO, particle swarm optimization.

maximum value of the feature subspace. Then, the model is over fitting. To verify the SPP-SVM model, we amplify and reduce the value of *c*. When *c* is between 1 to 100, the number of boundary support vectors is monotone decreasing but non-0. When *c* exceeds 100, the number of boundary support vectors decreases rapidly to 0 ($c = 200$). That is, when *c* exceeds 100, all the training samples are classified accurately, resulting in over fitting. Therefore, when $c = 40.1731$ and $\gamma = 0.01$, it does not have an over fitting, and they are the best parameters combination. Meanwhile, the best multi-class accuracy is 92.12%, the best average two-class accuracy is over 97%, and the shortest elapsed time is less than 1 second, which is shown in Table 6. Even with small samples, SPP-SVM model can get higher accuracy and lower elapsed time than traditional SVM model.

4.6 | Performance comparison

Performance comparison of SPP-SVM's each stage is shown in Figure 4, where S-SVM is the scaling stage; SP-SVM is the scaling and feature extraction stage (threshold = 94%); and SPP-SVM is the scaling, feature extraction, and parameters optimization stage.

The performance of two- and multi-classes are improved visibly on S-SVM.

Accuracy is close to the original dimension on SP-SVM, but dimension is lower and elapsed time is shorter.

Accuracy is the best, and the elapsed time is shortest on SPP-SVM.

Performance comparison of SPP-SVM with different sample sizes is shown in Figure 5. The result shows that, only with

TABLE 6 Accuracy (%) and elapsed time(s) of two- and multi-class SPP-SVM (threshold = 94%, dimension = 20, c = 40.1731, and γ = 0.01)

Traffic class	WWW	Mail	FTP-control	FTP-pasv	Attack	P2P	Database	FTP-data	Multimedia	Services	Multi-class
Accuracy (%)	99.84	99.92	99.76	99.64	98.6	97.6	99.92	99.92	97.36	99.96	92.12
Elapsed time(s)	1.740768	0.710184	0.589454	0.763306	0.625813	0.711868	0.641928	0.551276	0.677295	0.742169	0.772112

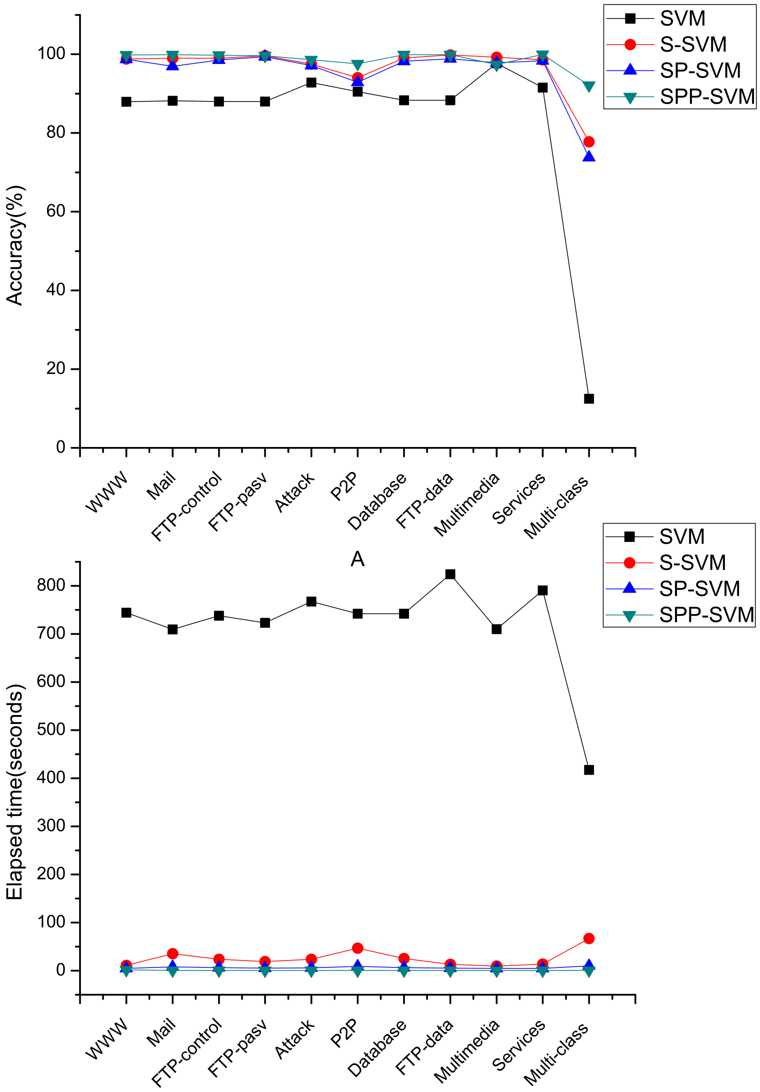


FIGURE 4 Accuracy (%) and elapsed time(s) of SPP-SVM each stage. SVM, support vector machine

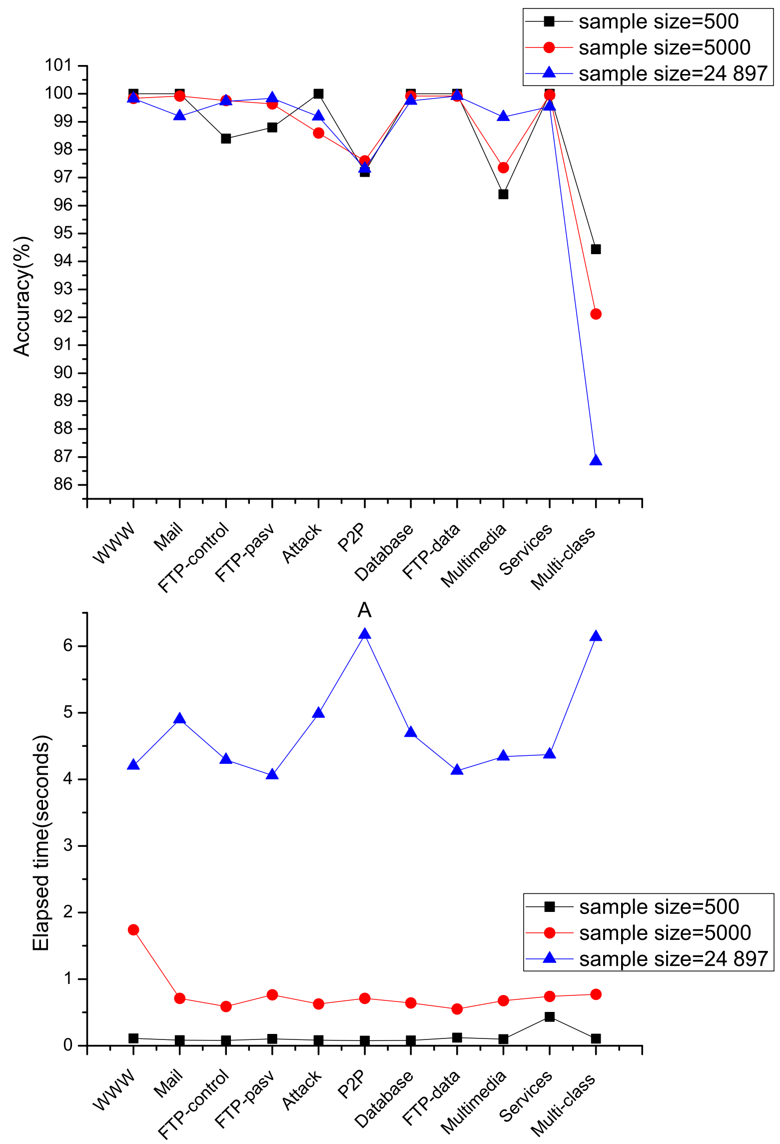


FIGURE 5 Accuracy (%) and elapsed time(s) of SPP-SVM with different sample size

a few hundred samples, SPP-SVM model can get higher overall accuracy and less elapsed time than with larger samples.

We also compare the classification performance of SPP-SVM to Naïve Bayes, k-NN, and RBF Network, respectively, with weka.[42] The comparison about classifiers above is shown in Figure 6. The performance of SPP-SVM is the best among the above classifiers, especially on the small datasets. Only with a few hundred samples, SPP-SVM's two-class average accuracy is up to 99.08%, which is 10.8% higher than Naïve Bayes's, 11.64% higher than RBF Network's, and 7.64% higher than k-NN's. SPP-SVM's multi-class accuracy is 94.44%, which is 6.44% higher than Naive Bayes's, 15.69% higher than RBF network's, 12.14% higher than k-NN's, and its average elapsed time is no more than 1 second. SPP-SVM is an effective model, which can classify the traffic flows in real time accurately.

We also inspected the classification performance of SPP-SVM on the other datasets. We applied the SPP-SVM model on the CAIDA dataset.[43] The CAIDA Internet traffic was collected from a different period of the 3 days. In parallel to DPI classification labels, 61 variables that are useful for nonpayload traffic classification methods. We deleted the traffic flows of unknown. And to reduce the imbalance of the data, we deleted the traffic flows, which the samples were very few. The dataset is shown in Table 7. And we extracted samples randomly within 50 from every subset (total of 650 samples) to build the small unbiased new training dataset. We applied the SPP-SVM model to scale data, extract features, and optimize parameters. Classification performance is the best when threshold is at 96%. As a result, the dimension is 16, the best c is 60.01, and the best γ is 0.01 on SPP-SVM. Meanwhile, best average accuracy is over 95.23%, which is shown in Table 8. SPP-SVM can also achieve good classification performance on other datasets.

5 | DISCUSSION

In this section, we provide some discussions on the model performance and related approaches.

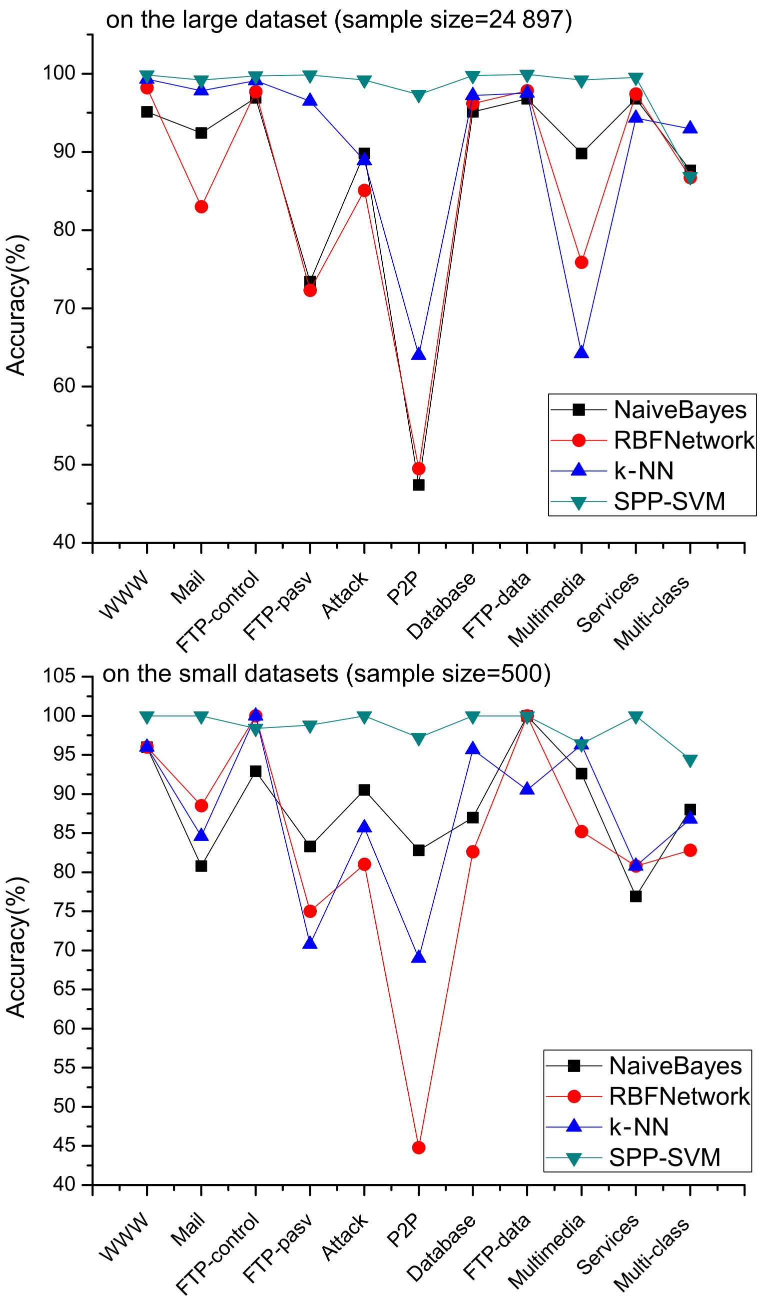


FIGURE 6 Accuracy (%) of Naïve Bayes, RBF network, k-NN, and SPP-SVM on the large dataset (sample size = 24 897) and small dataset (sample size = 500), respectively

TABLE 7 CAIDA dataset

Traffic class	HTTP	DNS	ICMP	SSL	MSN	BitTorrent	NTP	NetBIOS	Mail_POP
Samples of flows	33667	22814	1318	2671	190	61	81	58	64
Traffic class	Windowsmedia	Gnutella	Oscar	DirectDownloadLink					
Samples of flows	46	108	126	45					

Abbreviation: CAIDA, Cooperative Association for Internet Data Analysis.

TABLE 8 Accuracy (%) on SPP-SVM (threshold = 96%, dimension = 16, c = 60.01, and $\gamma = 0.01$)

Traffic class	HTTP	DNS	ICMP	SSL	MSN	BitTorrent	NTP	NetBIOS	Mail_POP
Accuracy (%)	92.5	99.375	99.9	92.1875	93.125	97.19	96.5625	95.625	92.50
Traffic class	Windowsmedia	Gnutella	Oscar	DirectDownloadLink					
Accuracy (%)	99.6875	92.188	94.0625	93.125					

TABLE 9 Related approaches

Approach of SVM	Training Set	Feature Selection	Parameters Optimization	Classification Performance
SPP-SVM (proposed in this paper)	Unbiased, a few hundred samples	PCA	Improved PSO	98.6%
Este et al[12]	Unbiased, a few hundred samples	No	Grid-search	98.8%
Yuan et al[13]	Unbiased, 1400 samples	Sequential forward	Grid-search	97.17%
Kim et al[17]	Unbiased, 1000 flows	CFS	No	More than 95%
Khan L et al[18]	Biased, randomly	No	Experience	69.8%, shortest training time
Sena et al[32]	Unbiased, 15% flows	No	Grid-search	More than 95%

Abbreviation: CFS, content filtering service; PCA, principal component analysis; PSO, particle swarm optimization; SVM, support vector machine.

5.1 | Performance

On SPP-SVM, we first map the attributes into smaller intervals, avoid the dimensional effects, and accelerate the convergence of the programs. After scaling, the average accuracy of two-class is 8.34% higher and the accuracy of multi-class is 65.3% higher than the traditional SVM. The elapsed time of two-class is 97.06% shorter, and the elapsed time of multi-class is 83.98% shorter than the traditional SVM. We extract features based on PCA. When threshold is set at 94%, dimension is 20, average accuracy of two-class is 7.57% higher than its of traditional SVM, and accuracy of multi-class is 61.32% higher than its of traditional SVM. The performance is close to the scaling datasets without feature extraction, and the original datasets can be replaced by the unbiased small samples datasets for training. Finally, we derived the optimal parameters C and γ based on PSO automatically.

On SPP-SVM model, average accuracy of two-class is 98.6%, which is 15.54% higher than that of the traditional SVM model. Accuracy of multi-class is 92.12%, which is 79.6% higher than that of the traditional SVM model. Dimension is 20, which is 89.8% lower than that of the traditional SVM model. Average elapsed time is less than 1 second, which is 99% shorter than that of the traditional SVM model.

Compared with others' typical supervised algorithms, the SPP-SVM's average accuracy of two-class is 11.25% higher than that of Naïve Bayes, 13.29% higher than that of RBF Network, and 8.45% higher than that of k-NN. Accuracy of multi-class is equivalent to that of k-NN, 4.49% higher than that of Naïve Bayes, and 5.42% higher than that of RBF Network.

5.2 | Related approaches

Table 9 compares the related approaches under 4 properties, ie, the training set, the feature selection, parameters optimization, and the performance of classification. The proposed approach, SPP-SVM model, has the advantages over other related approaches. Feature extraction by PCA can not only solve the curse of dimensionality effectively but also avoid deleting more information. In Table 9, the related approaches mostly used specific feature selection approaches. These approaches are mainly to find the smallest contribution feature and remove the feature. However, in practice, many features depend on each other or depend on the underlying unknown variables. A feature can be represented by a combination of multiple types of information. Removing such a feature would remove more information. Parameters optimization based on improved PSO is a heuristic-algorithm. It does not need to find the parameters in a wider range, it does not need to adjust the step gradually, and it does not need to traverse all the parameters points in the grids. From this perspective, the SPP-SVM model is much more effective.

6 | CONCLUSION

In this paper, we have proposed a new accurate network traffic classification model which is based on SVMs. A comprehensive analysis on the model framework has been done. The model is capable of solving the curse of dimensionality, reducing computation complexity, and searching for the optimal working parameters of kernel function automatically. Numbers of experiment results from 2 traffic datasets prove that the model can get the superior performance with hundreds of training samples and is more suitable for real-time traffic classification.

Besides, the proposed model can be applied to traffic classification related feature extraction and parameters optimization. Further research and improvement on SVM multi-classifier are ongoing.

REFERENCES

1. Gomes JV, Inácio PRM, Pereira M, et al. Detection and classification of peer-to-peer traffic: a survey. *ACM Computing Surveys*. 2013;45(3):1–40.
2. Dainotti A, Pesca e A. Issues and future directions in traffic classification. *IEEE Network*. 2012;26(1):35–40.
3. Zavala Y, De GS, et al. Internet traffic classification using multifractal analysis approach. *Simulation Series*. 2012;44(3):38–42.
4. Wujian Y, Kyungsan C. Hybrid P2P traffic classification with heuristic rules and machine learning. *Soft Computing*. 2014;18(9):1815–1827.
5. Yu W, Yang X, et al. Generating regular expression signatures for network traffic classification in trusted network management. *Journal of Network and Computer Applications*. 2012;35(3):992–1000.
6. Tongaonkar A, Torres R, et al. Towards self adaptive network traffic classification. *Computer Communications*. 2015;56:35–46.
7. Jin Y, Duffield N, et al. A modular machine learning system for flow-level traffic classification in large networks. *ACM Transactions on Knowledge Discovery from Data*. 2012;6(1):1–34.
8. Zhang J, Xiang Y, et al. Network traffic classification using correlation information. *IEEE Transactions on Parallel and Distributed Systems*. 2013;24(1):104–117.

9. Yu W, Yang X, Yu SZ. An automatic application signature construction system for unknown traffic. *Concurrency and Computation Practice and Experience*. 2010;22(13):1927–1944.

10. Finamore A, Mellia M, Meo M. Mining unclassified traffic using automatic clustering techniques. *Lecture Notes in Computer Science*. 2011,6613: 150–163.

11. Wang Y, Yu SZ. Machine learned real-time traffic classifiers. *Proceedings 2nd International Symposium on Intelligent Information Technology Application*. 2008;3:449–454.

12. Este A, Gringoli F, Salgarelli L. Support vector machines for TCP traffic classification. *Computer Networks*. 2009;53(14):2476–2490.

13. Yuan R, Li Z, et al. An SVM-based machine learning method for accurate internet traffic classification. *Information Systems Frontiers*. 2010;12(2): 149–156.

14. Bashar A, Parr G, et al. Application of Bayesian networks for autonomic network management. *Journal of Network System Manage*. 2014;22(2):174–207.

15. Monemi A, Zarei R, et al. Online NetFPGA decision tree statistical traffic classifier. *Computer Communications*. 2013;36(12):1329–1340.

16. Auld T, Moore AW, Gull SF. Bayesian neural networks for internet traffic classification. *IEEE Trans Neural Netw*. 2007;18(1):223–239.

17. Kim H, claffy kc, Fomenkov M, Barman D, Faloutsos M, Lee K. Internet traffic classification demystified: myths, caveats and the best practices. ACM CoNEXT 2008, December 10–12, 2008, Madrid SPAIN.

18. Khan L, Awad M, et al. A new intrusion detection system using support vector machines and hierarchical clustering. *VLDB Journal*. 2007;16(4): 507–521.

19. Chen S, Wang W, et al. Construct support vector machine ensemble to detect traffic incident. *Expert Systems with Applications*. 2009;36(8):10976–10986.

20. Tellenbach B, Burkhart M, Schatzmann D, et al. Accurate network anomaly classification with generalized entropy metrics. *Computer Networks*. 2011;55(15):3485–3502.

21. Zhu L, Yuan R, et al. Accurate classification of the internet traffic based on the SVM method. IEEE International Conference on. Communications, 2007;1373–78.

22. Internet Assigned Numbers Authority. Available from: http://www.iana.org [15 November 2009].

23. Karagiannis T, Broido A, Brownlee N, Claffy K. Is P2P dying or just hiding? Proceedings of the 47th Annual IEEEGlobecom 2004, Dallas, TX, U.S.A., November/December 2004.

24. Sen S, Spatscheck O, Wang D. Accurate, scalable in-network identification of P2P traffic using application signatures. WWW2004, New York, NY, U.S.A., May 2004.

25. Ohm P, Sicker DC, Grunwald D. Legal issues surrounding monitoring during network research. In Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC'07). ACM, New York, NY, 2007;141–148.

26. Williams N, Zander S, Armitage GJ. A preliminary performance classification of five machine learning algorithms for practical IP traffic flow classification. *Computer Communication Review*. 2006;36(5):5–16.

27. Riyad Alshammari A, Zincir-Heywood N, Farrag AA. Performance comparison of four rule sets: an example for encrypted traffic classification. Proceeding of CONGRESS'09 Proceedings of the 2009World Congress on Privacy, Security, Trust and the management of e-business: 21–28.

28. Moore AW, Zuev D, et al. Discriminators for use in flow-based. Technical Report RR-05-13, Department of Computer Science, Queen Mary, University of London, 2005;1–16.

29. Jiyong O, Nojun K. Generalized mean for robust principal component analysis. *Pattern Recognition*. 2016;54:116–127.

30. Gianni D'A, Salvatore R. Feature extraction and soft computing methods for aerospace structure defect classification. *Measurement: Journal of the International Measurement Confederation*. 2016;85:192–209.

31. AlSabah M, Bauer K, Goldberg I. Enhancing Tor's performance using realtime traffic classification. Proceedings of the ACM Conference on Computer and Communications. Security, 2012;73–84.

32. Sena GG, Belzarena P. Statistical traffic classification by boosting support vector machines. Proceedings of the 7th Latin American Networking Conference, 2012; 9–18.

33. Kuang R, Gu J, Cai H, et al. Improved prediction of malaria degradomes by supervised learning with SVM and profile kernel. *Genetica*. 2009;136(1): 189–209.

34. Yassin NI, Salem NM, Adawy MI. Entropy based video watermarking scheme using wavelet transform and principle component analysis. International Conference on. Engineering and Technology, 2012.

35. Fahad A, Tari Z, Khalil I, et al. An optimal and stable feature selection approach for traffic classification based on multi-criterion fusion. *Future Generation Computer Systems*. 2014;36:156–169.

36. Tayal VK, Lather JS. Reduced order H_∞ TCSC controller & PSO optimized fuzzy PSS design in mitigating small signal oscillations in a wide range. *International Journal of Electrical Power and Energy Systems*. 2015;68:123–131.

37. Wang Y, Lv J, Zhu L, Ma Y. CALYPSO: a method for crystal structure prediction. *Comput Phys Commun*. 2012;183(10):2063–2070.

38. Abbas ED, Patrick S. Effect of the dynamic topology on the performance of PSO-2S algorithm for continuous optimization. *Lecture Notes in Computer Science*. 2015;9432:60–64.

39. Xiaodong L, Yao X. Cooperatively coevolving particle swarms for large scale optimization. *IEEE Transactions on Evolutionary Computation*. 2012;16(2): 210–224.

40. Chang C-C, Lin C-J. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*. 2011;2(3):

41. Vatankhah M, Asadpour V, et al. Perceptual pain classification using ANFIS adapted RBF kernel support vector machine for therapeutic usage. *Applied Soft Computing Journal*. 2013;13(5):2537–2546.

42. Weka 3, data mining software in java, Available from: http://www.cs.waikato.ac.nz/ml/weka/. Accessed 2014.

43. The Cooperative Association for Internet Data Analysis (CAIDA), Available from: http://www.caida.org. Accessed 2014.

How to cite this article: Cao J, Fang Z, Qu G, Sun H, Zhang D. An accurate traffic classification model based on support vector machines. *Int J Network Mgmt*. 2017;27:e1962. doi: 10.1002/nem.1962

AUTHOR BIOGRAPHIES

Jie Cao received the MS degree in information engineering from Northeast Electric Power University, Jilin, China, in 2007. She is currently a lecturer of information engineering in Northeast Electric Power University and now she is a PhD candidate in Computer science and Technology of Jilin University. Her research interests include the areas of computer network, computer system architecture, and data mining.

Zhiyi Fang received the PhD degree in computer science from Jilin University, Changchun, China, in 1998, where he is currently a professor of computer science. He was a senior visiting scholar of the University of Queensland, Australia, from 1995 to 1996, and the University of California, Santa Barbara, from 2000 to 2001. He is a member of China Software Industry Association (CSIA) and a member of Open System Committee of China Computer Federation (CCF). He has published SCI/EI retrieval International Conference/ journal articles more than 40 articles. He is a professor of

Computer Science and Technology Institute, Jilin University, doctoral tutor. His keynote courses are distributed systems, parallel computing system, mobile communication, and wireless networks.

Guannan Qu recieved the BE degree in computer science and technology from Jinan University, Guangzhou, China, in 2003, and the MS and PhD degrees in computer science from Jilin University, Changchun, China, in 2007 and 2010, respectively. She is currently a lecturer of computer science and technology in Jilin University. She was financially supported by China Scholarship Council as a visiting scholar in the University of Texas at Dallas, from 2008 to 2010. Her research interests include the areas of computer system architecture, performance analysis, and quality-of-service issues in high-speed networks. She is a member of the IEEE.

Hongyu Sun received the master degree in Computer Science and technology from Jilin University, Changchun, China in 2012, and now she is a PhD candidate in Computer science and Technology of Jilin University. From Jan. 2015 to now, she is a visiting scholar in the University of Maryland, Baltimore County. Her research interests include Internet of Thing (IoT), wireless communications, and 5G technologies.

Dan Zhang received the master degree in Software Engineering from Jilin University, Changchun, China in 2006; and now she is a PhD candidate in Computer Science and Technology of Jilin University: her research interest includes localizations, routing of wireless sensor networks.