# Density-based Adaptive Wavelet Kernel SVM Model for P2P Traffic Classification

Xinlu Zong[1], Chunzhi Wang[2] and Hui Xu[3]

*School of Computer Science and Technology, Hubei University of Technology Wuhan 430068, China*

[1]*zongxinlu@126.com,* [2] *chunzhiwang@vip.163.coml,* [3] *xuhui_1004@126.coml*

## *Abstract*

*In this paper an adaptive wavelet kernel based on density SVM approach for P2P traffic classification is presented. The model combines the multi-scale learning ability of wavelet kernel and the advantages of support vector machine. Mexican hat wavelet function is used to build SVM kernel function. The wavelet kernel function is tuned adaptively according to the density of samples around support vectors for several times during the training process. The experimental results show that the presented model can improve classification accuracy while reducing the number of support vectors and has better performance for solving P2P traffic classification.*

*Keywords: Traffic classification, Peer-to-peer, Wavelet, SVM*

## 1. Introduction

Over the past few years, peer-to-peer (P2P) technologies, which are considered as new Internet technology, have become attracting widespread attention of researchers. Recent studies have shown that P2P has been applied to many different aspects, such as multimedia fine sharing, live video streaming and so on. P2P is an Internet application that allows a group of users to share their files and computing resources. It has been observed that P2P traffic has become the majority of today's Internet traffic [1]. Therefore, accurate P2P traffic identification is crucial to network activities [2-4].

Traditional identification techniques can be classified into three catalogues, including port matching [5] and protocol decoding or packet payload analysis [6]. However, with the growth of the Internet traffic, in terms of number and type of applications, traditional identification techniques are no longer effective. Current P2P applications tend to disguise their traffic by using arbitrary port numbers in order to break firewalls and network management applications. Identification methods based on payload analysis, which searches packet payloads for the signatures of known applications is ineffective for the traffic which is encrypted.

Due to the ineffectiveness associated with the traditional methods, several identification methods based on data mining techniques have been proposed in recent years to identify the Internet traffic using its statistical characteristics [7]. This kind of approaches is able to identify the encrypted traffic. In general, statistic-based methods include three steps: abstracting the traffic as a feature vector, estimating the statistical distribution of the feature vector, and recognizing a sample. For example, Erman *et al.,* [8] proposed a classification model and evaluated K-Means and DBSCAN algorithms and compared them to AutoClass algorithm. Iliofotou *et al.,* [9] used Traffic Dispersion Graphs to remedy limitations of

traditional classification methods. This graph-based classification framework provided a systematic way to classify traffic by using information from the network-wide behavior and flow-level characteristics of Internet applications. Yin *et al.*, [10] presented a classification method for network traffic classification. A hidden Markov was used to model for a type of traffic. But this method is a supervised learning and relies on predefined classes and class-labeled training data sets.

Recent years, many researchers have focused their concerns on methods based on machine learning [11]. Machine Learning is aimed to generate general classifiers based on network features from available data and then classify unknown data using the results from learning.

Currently, support vector machine (SVM) has provided a new way to solve P2P traffic identification, because of its superiority in high generalization performance, globally optimal and dealing with curse of dimensionality. Many research works on network traffic classification based on SVM have been reported in recent years [12-14]. For example, Yuan *et al.*, [15] used SVM to optimize network flow features via multiple classifier selection methods. Este *et al.*, [16] introduced a classification technique based on SVM. The mechanism is based on a flow representation that expresses the statistical properties of an application protocol. In their method, an optimization procedure was applied to derive the ideal SVM parameters for the data used.

As a machine learning method on finite samples, SVM has good performance in generalization ability and robustness. However, SVM classifies sample data upon only one scale and fails to solve classification problems on multi-scale samples.

Actually, the P2P traffic in real network environment obviously shows the characteristics of self-similarity, mutability and multi-scale. Due to these complex characteristics, the traditional models have not identified network traffic effectively. The approaches combined several identification methods based on traffic features can improve recognition rate to some extent, but the cost of computation is rather expensive.

In this paper, an adaptive wavelet kernel based on density (AWKBD) SVM model for P2P traffic identification is proposed to combine the multi-scale learning method of wavelet analysis and the advantages of support vector machine. Mexican hat wavelet function is utilized to build SVM kernel function. During the training process, the kernel function is tuned dynamically according to the density of samples around support vectors. Then the tuned kernel function is used to retrain samples. The experimental results show that the presented model is suitable for solving P2P traffic classification problem. The AEKBD SVM model can improve classification accuracy while reducing the number of support vectors.

## 2. Basic Concepts of SVM Classification and Wavelet Theory

### 2.1. Support Vector Machine

SVM is a learning algorithm developed by Vapnik [17], which is based on the small sample statistical theory and the maximum class interval thought. SVM has shown prominent superiority, especially in handling noise, large datasets and large input spaces and widely been applied in pattern recognition. As a supervised method, SVM has on two stages: training and evaluation. During the training process, the algorithm acquires the information about the classes. The goal of the algorithm is to estimate the boundaries between the classes in the training sets. And during the evaluation process, the testing set is examine by the trained classification mechanism to associates its members to the corresponding classes.

In the binary classification problem the objective of SVM is to find the optimal separating hyper-plane between data points of two classes.

For a two-class problem, given a training set $(x_i, y_i)$, $i = 1, 2, ..., n$, where $x_i \in R^n$ and $y_i \in \{+1, -1\}$, SVM finds an optimal separating hyper-plane with the maximum margin by solving the following constrained optimization problem:

$$\text{Min } \frac{1}{2} w^T w \qquad (1)$$

Subject to:

$$y_i[(w \cdot x_i) + b] - 1 \geq 0, i = 1, 2, ..., n \qquad (2)$$

The above optimization problem represents the minimization of a quadratic program under linear constraints. A convenient way to solve the constrained minimization problems is to transform the problem to its dual using Lagrange optimizing method, then solve the quadratic programming problem:

$$\text{Max } \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i,j=1}^{n} a_i a_j y_i y_j (x_i \cdot x_j) \qquad (3)$$

Subject to:

$$\sum_{i=1}^{n} a_i y_i = 0 \quad a_i \geq 0, i = 1, 2, ..., n \qquad (4)$$

Where $a_i$ is the Lagrange multiplier corresponding to each sample. Generally, most $a_i$ of the samples are zero, and the samples corresponding non-zero $a_i$ are called support vectors. The decision function is defined by the following equation:

$$f(x) = \text{sgn}(\sum_{i=1}^{n} a_i^* y_i (x_i \cdot x) + b^*) \qquad (5)$$

The above concepts can also be extended to the non-separable problems. In terms of these slack variables, the main goal of SVM is to find an optimal separating hyper-plane, to maximize the separation margin, and to minimize the empirical classification error. Thus, the problem of seeking the optimal separating hyper-plane has the expression as follows:

$$\text{Min } \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \qquad (6)$$

Subject to:

$$y_i(w^T x_i - b) \geq 1 - \xi_i$$
$$\xi_i \geq 0 \quad i = 1, 2, ..., n \qquad (7)$$

where $C$ is a penalty parameter on the training error, and $\xi_i$ is the non-negative slack variables.

The above quadratic optimization problem can be solved by transforming it into Lagrange function:

$$L = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} a_i[1 - \xi_i - y_i(w^T x_i - b)] - \sum_{i=1}^{n} \theta_i \xi_i \qquad (8)$$

where $a_i$, $\theta_i$ denote Lagrange multipliers; $a_i \geq 0$ and $\theta_i \geq 0$.

Thus, the dual optimization problem can be transformed as follows:

$$\text{Max } \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j y_i y_j k(x_i, x_j) \qquad (9)$$

$$\text{Subject to: } \sum_{i=1}^{n} a_i y_i = 0, \; 0 \le a_i \le C, \; i = 1, 2, ..., n \tag{10}$$

The decision function can be expressed as the following form:

$$f(x) = \text{sgn}(\sum_{i=1}^{n} a_i^* y_i k(x_i \cdot x) + b^*) \tag{11}$$

The commonly used kernel functions for SVM are the following three kinds: polynomial kernel, radial basis functions (RBF) kernel and Sigmoid kernel. These functions are computed in a high dimensional space and have a nonlinear character. They are mathematically defined in Eqs. (12)-(14).

Polynomial kernel:

$$k(x_i, x_j) = [(x_i \cdot x_j) + c]^q \tag{12}$$

Radial basis function RBF kernel:

$$k(x_i, x_j) = \exp\left[ -\frac{|x_i - x_j|^2}{\sigma^2} \right] \tag{13}$$

Sigmoid kernel:

$$k(x_i, x_j) = \tanh(v(x_i \cdot x_j) + c) \tag{14}$$

## 2.2. Wavelet Kernel Function

Wavelet analysis [18] is one kind of time/frequency analytical methods. Wavelet analysis attempts to solve these problems by decomposing a time-series into time/frequency space simultaneously. One gets information on both the amplitude of any "periodic" signals within the series, and how this amplitude varies with time. Broadly speaking, the wavelet transform can provide economical and informative mathematical representations of many different objects of interest. Such representations can be obtained relatively quickly and easily through fast algorithms. As a result, wavelets are used widely, not only by mathematicians in areas such as functional and numerical analysis, but also by researchers in the natural sciences such as physics, chemistry and biology, and in applied disciplines such as computer science, engineering and econometrics.

The support vector's kernel function can be described as not only the product of point, such as $K(x, x') = K(x \cdot x')$, but also the horizontal floating function, such as $K(x, x') = K(x - x')$. In fact, if a function satisfied condition of Mercer [19], it is the allowable support vector kernel function.

Lemma 1. The symmetry function $K(x, x')$ is the kernel function of SVM if and only if: for all function $g \ne 0$ which satisfies the condition of $\int_{R^n} g^2(\xi) d\xi < \infty$, we need satisfy the condition as follows:

$$\iint K(x, x') g(x) g(x) dx dx' \ge 0, \quad x, x' \in \mathbf{R}^n \tag{15}$$

All constructed SVM kernel functions must satisfy this theorem.
The condition of horizontal floating kernel function is given in Lemma 2 [20]:

Lemma 2. The horizontal floating function is allowable support vector's kernel function if and only if the Fourier transform of $K(x)$ need satisfy the condition follows:

$$F[K(x)] = (2\pi)^{-n/2} \int_{R^n} \exp(-jwx)K(x)dx \geq 0, \quad x \in R^n \tag{16}$$

If a function satisfies conditions of Lemma 1 and Lemma 2 simultaneously, it is allowable SVM's horizontal floating kernel function.

If the wavelet function $\psi(x)$ satisfies the conditions: $\psi(x) \in L^2(R) \cap L^1(R)$, and $\overset{\wedge}{\psi}(x) = 0, \overset{\wedge}{\psi}$ is the Fourier transform of function $\psi(x)$. The wavelet function group can be defined as:

$$\psi(x) = (a)^{\frac{1}{2}} \psi(\frac{x-b}{a}), \quad x \in R \tag{17}$$

where $a$ is the scaling factor, $a$ is the horizontal floating factor, and $\psi(x)$ is called the "mother wavelet".

Horizontal floating wavelet kernel function which satisfies Mercer condition is built as follows:

$$K(x, x^{'}) = k(x - x^{'}) = \prod_{i=1}^{n} \psi(\frac{x_i - x_i^{'}}{a_i}) \tag{18}$$

where $a_i$ is the scaling parameter of wavelet, $a_i > 0$.

It can be proven that kernel functions constructed by Mexican hat mother wavelet function $h(x) = (1 - x^2)\exp(-\frac{x^2}{2})$ can satisfy the conditions of allowable support vector's kernel function. The built wavelet SVM function is as below:

$$K(x, x^{'}) = \prod_{i=1}^{n} (1 - \frac{\|x - x^{'}\|^2}{a_i})\exp(-\frac{\|x - x^{'}\|^2}{2a_i^2}) \tag{19}$$

## 3. Adaptive Wavelet SVM for P2P Classification

### 3.1. Adaptive Kernel Function

Traditional kernel methods for classification problems usually tune kernel parameters manually and can not achieve better effectiveness. Among pattern recognition problems, distance between two classes need be enlarged as far as possible in order to separate the two different classes much better. It means that the local region nearby separating hyper-plane should be magnified while keeping other regions with little changes. Adaptive kernel, as a computing form of kernel function, is able to get the most befitting kernel parameters by calculating training data repeatedly, as Figure 1 shows.
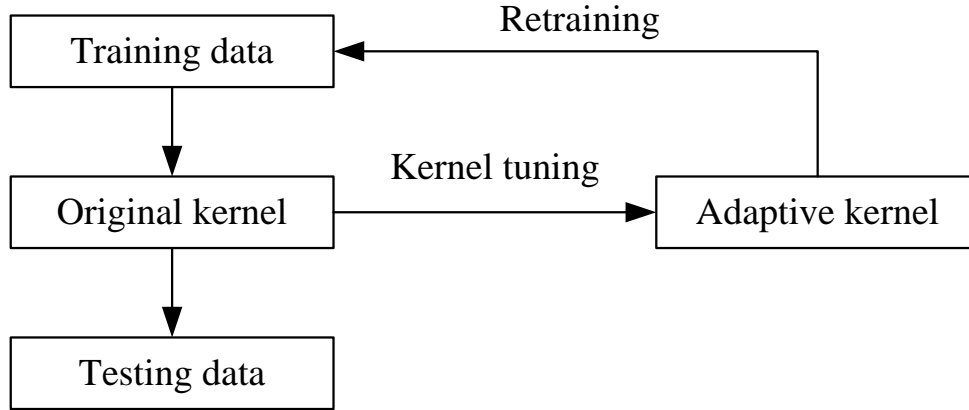
**Figure 1. Adaptive SVM Training**

Considering the fact that support vectors always appear in the vicinity of separating hyper-plane, more attention should be paid to the regions surrounding these support vectors. Adaptive kernel function can be used to stress the importance of each support vector on the basis of original kernel:

$$k^*(x,x^{'}) = c(x)x(x^{'})k(x,x^{'}) \qquad (20)$$

Actually, the extent of attention to each support vector is different due to the different density of samples around each support vector. A proper way is to attach greater importance to the regions around support vectors which are concentrated while less importance to those regions which are rather sparse.

The density of samples around the $i^{th}$ support vector is defined as:

$$d_i = \frac{\sum_a \|x_a - x_i\|}{K} \qquad (21)$$

where $K$ is the number of the nearest samples, and $a$ is the set of nearest samples.

Accordingly, we construct $c(x)$ function as below:

$$c(x) = \sum_{i \in SV} a_i e^{\frac{-\|x-x_i\|^2}{d_i^2}} \qquad (22)$$

where $a_i$ is the Lagrange multiplier of the $i^{th}$ support vector.

Here, the wavelet kernel function given in Eq. (19) was selected as original SVM kernel function. The proposed adaptive wavelet kernel function is described as:

$$k^*(x,x^{'}) = \sum_{i \in SV} a_i \exp(-\frac{\|x-x_i\|^2}{d_i^2}) \sum_{i \in SV} a_i \exp(-\frac{\|x^{'}-x_i\|^2}{d_i^2}) \prod_{i=1}^{n} (1-\frac{\|x-x^{'}\|^2}{a_i}) \exp(-\frac{\|x-x^{'}\|^2}{2a_i^2}) \qquad (23)$$

### 3.2. Adaptive SVM Training Algorithm

Adaptive SVM training algorithm based on adaptive wavelet kernel function is given below:

Step 1: Initialization

Step 1.1 Set parameters of SVM, including penalty parameter $C$, $\sigma$, $gen\_\max$

Step 1.2 Set $gen = 0$

Step 2: Train original SVM

Step 2.1 Train SVM on training set using wavelet kernel function given in Eq. (19)

Step 2.2 Get each preliminary support vector with Lagrange multiplier $a_i > 0$

Step 3: Repeat until $gen > gen\_\max$

Step 3.1 Calculate density of each support vector by Eq. (21)

Step 3.2 Adjust kernel function according to Eq. (23)

Step 3.3 Train the new adaptive SVM on training set

Step 3.4 Update support vectors and learning parameters

Step 3.5 $gen = gen + 1$

Step 4: Test and return

Step 4.1 Test on testing set by using the adaptive kernel function of iteration $gen$

Step 4.2 Output classification results

## 4. Experimental Results

Experiments were implemented to test the effectiveness of the proposed approach. The proposed AWKBD SVM approach was implemented in a MATLAB software environment, and run on a personal computer with Intel CPU 2.40GHz and 4 GB of RAM.

Parameters of the adaptive wavelet kernel SVM method were set as follows: scaling factor $a=2.5$, penalty parameter $C=2.5$, $\sigma=1$. There kernel functions—RBF kernel, adaptive RBF kernel and adaptive wavelet kernel based on density, were applied to experiments to analyze the performances of different approaches.

**Table 1. Results of Three Methods**

| Kernel | Accuracy (%) | Training Time (s) | Testing Time (s) | SVs |
|---|---|---|---|---|
| RBF | 91.43 | 1.825 | 0.188 | 121 |
| Adaptive RBF | 94.27 | 1.607 | 0.234 | 116 |
| Adaptive wavelet | 96.00 | 2.403 | 0.624 | 83 |

From the experiment results presented in Table 1, it can be clearly seen that AWKBD approach has a better performance than using traditional RBF and Adaptive RBF kernel functions. The classification accuracy of AWKBD, on the one hand, has been improved up to 96%, which is nearly 4% higher than that of traditional RBF kernel SVM. On the other hand, the number of support vectors is obviously decreased with three different kernel functions and

the number of support vectors (SVs) of AWKBD are reduce by 31% comparing with RBF kernel. It is shown in Table 1 that the training and testing time of AWKBD are longer than RBF kernel because of the adjustment to original kernel. Considering the better performance in classification accuracy and number of support vectors, it is a trade-off and can be acceptable.

**Table 2. Results with Different Penalty Parameters**

| Kernel | C | Margin | Accuracy (%) | SVs | Training Time (s) | Errors |
|---|---|---|---|---|---|---|
| RBF | 1 | 0.15267 | 91.43 | 123 | 1.997 | 15 |
| | 2 | 0.1243 | 91.43 | 121 | 1.731 | 2 |
| | 4 | 0.10353 | 92.00 | 124 | 1.654 | 1 |
| | 10 | 0.069657 | 89.14 | 121 | 1.450 | 0 |
| | 100 | 0.069657 | 89.14 | 121 | 3.650 | 0 |
| Adaptive RBF | 1 | 0.23419 | 95.43 | 119 | 2.309 | 4 |
| | 2 | 0.17843 | 94.29 | 116 | 1.576 | 1 |
| | 4 | 0.14142 | 96.00 | 102 | 2.137 | 0 |
| | 10 | 0.14324 | 96.00 | 100 | 2.605 | 0 |
| | 100 | 0.14324 | 96.00 | 100 | 2.418 | 0 |
| Adaptive wavelet | 1 | 0.28549 | 94.88 | 95 | 2.480 | 2 |
| | 2 | 0.26472 | 96.00 | 83 | 2.293 | 1 |
| | 4 | 0.20623 | 95.43 | 95 | 2.075 | 0 |
| | 10 | 0.22314 | 96.00 | 78 | 2.355 | 0 |
| | 100 | 0.22541 | 96.00 | 79 | 2.995 | 0 |

Table 2 lists the results under different values of penalty parameter. It is shown than methods based on adaptive kernel has fewer support vectors and points that classified incorrectly. As $C$ increases, the number of misclassified samples (Errors) of three methods can be reduced to 0 because the punishment to misclassified samples is increased. But it is not mean that classification performance is better with bigger values of $C$. When $C$ is set to 100, the generalization effect has not become better, and it costs more training time. Therefore, only when $C$ is set to an appropriate value, the number of support vectors is decreased and the margin between two classes is moderate. Then the better learning ability and generalization effect can be achieved.

The varying curves of the number of support vectors are depicted in Figure 2. As it is shown, the number of vectors of proposed AWKBD is always lower than that of adaptive RBF kernel owing to the multi-scale learning ability of wavelet kernel function. At each iteration, the AWKBD approach adaptively magnifies regions around support vectors trained from last iteration. As a result, the points which were considered as support vectors are no longer in the vicinity of separating hyper-plane of tuned kernel.

Figure 3 shows the curves of performances of AWBKD versus scaling factor of wavelet kernel function. It is observed that classification accuracy rate, number of support vectors and separating margin width reduces sharply, while the number of misclassified samples increases when scaling factor is set 3 to 5. Thus, scaling factor is not suggested to set in this range. Comprehensively analyzing the varying curves in Figure 3, it is concluded that AWBKD approach has the best performances when scaling factor is in the range of 2 to 3.
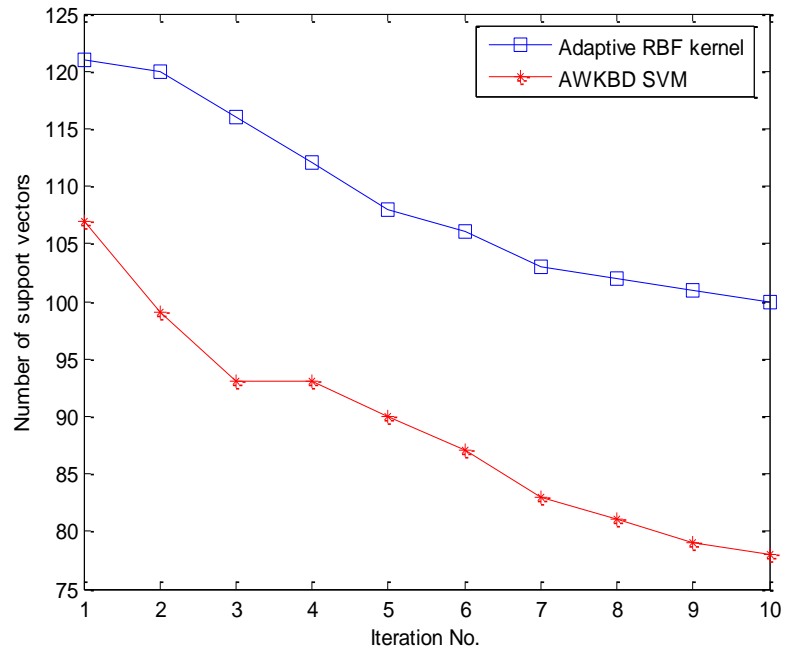
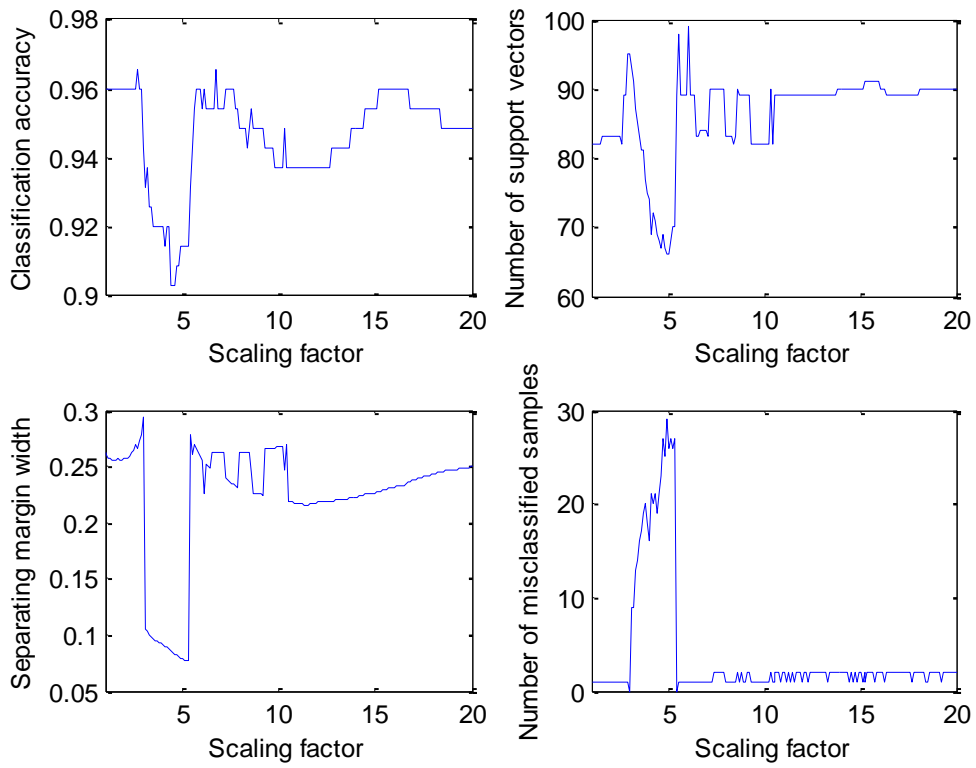**Figure 2. Cures of Varying Number of Support Vectors**



**Figure 3. Performances of AWBKD with Different Scaling Factors**

## 5. Conclusion

P2P traffic classification is a significant task in current Internet network. In this paper, an adaptive wavelet kernel based on density SVM approach for P2P traffic identification is proposed to combine the multi-scale learning method of wavelet analysis and the advantages of support vector machine. Mexican hat wavelet function is utilized to build SVM kernel function. During the training process, the kernel function is tuned dynamically according to the density of samples around support vectors. Then the tuned kernel function is used to retrain samples. The experimental results show that the presented model can improve classification accuracy while reducing the number of support vectors.

However, the AWKBD approach is more time-consuming, further research will focus on reduction of time complexity. Another challenging subject for future study is to apply adaptive kernel to other intelligent algorithms.

## Acknowledgements

## References

[1]  S. Iekel-Johnson Labovitz, D. McPherson, J. Oberheide and F. Jahanian, "Internet inter-domain traffic", Proceedings of the ACM SIGCOMM 2010 Conference on SIGCOMM, New York: ACM, **(2010)**, pp. 75-86.

[2]  K. Xu, M. Zhang, M. Ye, D. M. Chiu and J. Wu, "Identify P2P traffic by inspecting data transfer behavior", Computer Communications, vol. 33, no. 10, **(2010)**, pp. 1141-1150.

[3]  P. Bermolen, M. Mellia, M. A. Meo, D. Rossi and S. Valenti, "Abacus: Accurate behavioral classification of P2P-TV traffic", Computer Networks, vol. 55, no. 6, **(2011)**, pp. 1394-1411.

[4]  W. Hong, "A Novel Method for P2P Traffic Identification", Procedia Engineering, vol. 23, **(2011)**, pp. 204-209.

[5]  A. Gerber, J. Houle, H. Nguyen, M. Roughan and S. Sen, "P2P, the gorilla in the cable", Proceedings of National Cable & Telecommunications Association Conference, Chicago, **(2003)**, pp. 8-11.

[6]  S. Sen, O. Spatscheck and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures", Proceedings of the 13th International Conference on World Wide Web, New York, NY: ACM, **(2004)**, pp. 512-521.

[7]  H. Huang and J. Zhu, "Kernel based non-linear feature extraction methods for speech recognition", Proceedings of the sixth International Conference on Intelligent System Design and Applications, vol. 2, **(2006)**, pp. 749-754.

[8]  J. Erman, M. Arlitt and A. Mahanti, "Traffic classification using clustering algorithms", Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data, ACM, Pisa, Italy, **(2006)**, pp. 281-286.

[9]  M. Iliofotou, H. Kim, M. Faloutsos, M. Mitzenmacher, P. Pappu and G. Varghese, "Graption: A graph-based P2P traffic classification framework for the internet backbone", Computer Networks, vol. 55, no. 8, **(2011)**, pp. 1909-1920.

[10] C. G. Yin, S. Q. Li and Q. Li, "Network traffic classification via HMM under the guidance of syntactic structure", Computer Networks, vol. 56, no. 6, **(2012)**, pp. 1814-1825.

[11] X. Xie, J. W. K. Ho, C. Murphy, G. Kaiser, B. Xu and T. Y. Chen, "Testing and validating machine learning classifiers by metamorphic testing", Journal of Systems and Software, vol. 84, no. 4, **(2011)**, pp. 544-558.

[12] N. M. Khan, R. Ksantini, I. S. Ahmad and B. Boufama, "A novel SVM+NDA model for classification with an application to face recognition", Pattern Recognition, vol. 45, no. 1, **(2012)**, pp. 66-79.

[13] R. Noori, A. R. Karbassi, A. Moghaddamnia, D. Han, M. H. Zokaei-Ashtiani, A. Farokhnia and M. G. Gousheh, "Assessment of input variables determination on the SVM model performance using PCA, Gamma test, and forward selection techniques for monthly stream flow prediction", Journal of Hydrology, vol. 401, no. 3-4, **(2011)**, pp. 177-189.

[14] J. Yang, L. Yuan, Y. He and L. Chen, "Timely traffic identification on P2P streaming media", The Journal of China Universities of Posts and Telecommunications, vol. 19, no. 2, **(2012)**, pp. 67-73.

[15] R. Yuan, Z. Li, X. Guan and L. Xu, "An SVM-based machine learning method for accurate internet traffic classification", Information Systems Frontiers, vol. 12, **(2010)**, pp. 149-156.

[16] A. Este, F. Gringoli and L. Salgarelli, "Support vector machines for TCP traffic classification", Computer Networks, vol. 53, no. 14, **(2009)**, pp. 2476-2490.

[17] V. N. Vapnik, "The Nature of Statistical Learning Theory", Springer-Verlag New York, Inc, USA, **(1995)**.

[18] S. Arivazhagan and L. Ganesan, "Texture classification using wavelet transform", Pattern Recognition Letters, vol. 24, no, 9-10, **(2003)**, pp. 1513-1521.

[19] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equation", Philosophical Transactions of the Royal Society of London, vol. A-209, **(1909)**, pp. 415-446.

[20] A. J. Smola, B. Schölkopf and K. R. Müller, "The connection between regularization operators and support vector kernels", Neural Network, vol. 11, **(1998)**, pp. 637-649.

## Authors

**Xinlu Zong.** She received her M.Sc. in Computer Application Technology (2007) and PhD in Computer Application Technology (2011) from Wuhan University of Technology, China. Now she is a lecturer at School of Computer Science and Technology, Hubei University of Technology, China. Her current research interests include different aspects of Artificial Intelligence and Pattern Recognition.

**Chunzhi Wang.** She is a professor at the School of Computer Science in Hubei University of Technology, Wuhan, China. She is also the Dean of the School of Computer Science in Hubei University of Technology, Wuhan, China. Currently, her major field of study is cooperative management.

**Hui Xu.** She received her PhD in Computer Network (2010) from Central China Normal University, Wuhan, China. Now she is a lecturer at the School of Computer Science in Hubei University of Technology, Wuhan, China. Currently, her major field of study is network and service management.