Traffic Classification for the Dispatcher in a Server Farm based on SVM

V. Punitha
Department of Computer Science and Engineering
National Institute of Technology,
Tiruchirappalli, India
vpunitha21@gmail.com

C. Mala
Department of Computer Science and Engineering
National Institute of Technology,
Tiruchirappalli, India
mala@nitt.edu

ABSTRACT
The increased capacity of Internet connections leads to more and wider usage of Internet services. An effective task-server assignment policy is obligatory to improve the performance of the server farms. Scalable techniques that categories the traffic always assist to enhance the performance. Existing dispatcher of the server farm do not consider a timely classification approach for the improvement of QoS. So this paper proposes a classifier for the dispatcher which classifies the requests using dynamic properties of the server farm. The proposed meachine learning classification method based on SVM aims to minimize the makespan and to improve the execution fairness. The classification is performed and the performance in terms of precision and waiting time are measured and the simulated results have shown significant performance improvement.

CCS Concepts
• **Computing methodologies→Supervised learning by classification • Software and its engineering→Distributed systems organizing principles.**

Keywords
Machine learning; Support Vector Machine; Traffic Classification; Server assignment policy.

1. INTRODUCTION
Increasing demand of computational resources increases the use of open distributed systems. One of the open distributed systems is server farm or cluster server. The cluster sever has a set of servers organized together and offers mirroring service. This cluster architecture with a dispatcher has been widely used in various domains like commercial web farms, academic and scientific computing [3]. The performance of the server farm can always be improved by scalability i.e., increasing the number of servers in the server farm. The aim of scalability of parallel or distributed system is achieved by optimally balancing the load among the servers. Only an efficient server allocation policy can achieve the load balancing of this type of architecture. This paper focuses on balancing of load among servers while assigning a server for the incoming request. For perfect load balancing, the dispatcher should take real-time decision to identify the server by considering the current load of the server and request type.

Although diverse mechanisms exist for providing QoS in resource assignment, QoS is yet to be widely deployed [9]. This is a challenging task, and the classification of the requests helps in identifying the appropriate server. Perfect classification of traffic (requests) assists better exploitation of features of sever farm and this in turn improves QoS.

SVM (Support Vector Machine), a popularly used supervised machine learning method, is identified for classification due to its good theoretical justification. SVM finds a separating hyper plane that accurately distinguishes the samples into two classes. SVM classification works on higher dimensional space. So the features extracted from network traffic can be mapped easily into higher dimensional space. The authors of [4] also put forward that performance of classification using multi flow properties is superior than using single flow property as the input vector. Much research work has been done on SVM based traffic identification [5]. The authors of [4] proved that the SVM based method has higher accuracy than the centroid clustering method.

Existing classification which are modeled to identify the application are not efficient in server farm because they have not been devised to make use of dynamic state information of the server farm.

2. RELATED WORK
The aim of this section is to present a brief overview of some important existing works on traffic classification.

Algorithm proposed in [8] for Internet traffic identification is based on Genetic Algorithm(GA) and Particle Swarm Optimization (PSO-SVM). Intelligent computation method GA, is used to find the best feature subset and then the corresponding weights of each selected feature is calculated by PSO. This approach helps in pre-eminently classifying the network applications. The port and signature based techniques produce good outcomes for some applications, but they are not contemplated in this paper. Authors never mapped the classification for the improvement of QoS.

Various methods of identification of P2P traffic are studied in [2]. Three characteristics, flow duration, ratio of the IP address number and port number are chosen for identification. Further it is analyzed and concluded that identification method based on SVM effectively identifies P2P traffic. A machine learning method based on SVM is proposed in [10]. Traffic flow is constructed and represented in the form of regular expressions, called application signature. This signature is mapped with popular applications and this paper mainly focuses on identifying an optimized feature set using biased and unbiased training samples. Here QoS is not focused which is necessary for the present scenario.

DOI: http://dx.doi.org/10.1145/3059336.3059344

Authors of [7] proposed a Genetic Algorithm to determine the optimal parameters of SVM to improve the accuracy of the classification. Here, a subset of the feature set which produces highest classification accuracy is selected and optimal parameters combination is extracted using GA. But the classification is limited to segregate the traffic according to its application. In DAG-SVM, the classification error in a layer propagates up to the leaf layer. Improved DAG-SVM presented in [11] defines accurate hyper-plane based on decision function and distance. Here the classification error is corrected in the same sublayer. This methodology is also applied to categorize the traffic subject to its application. Dipti Tiwari proposed a methodology in [1], to improve feature discretization using flow statistical feature. The number of features in feature dataset is reduced using correlation based FS algorithm. The reduced set is applied to 5 different classifiers and analyzed. Bayes Net classifier performed better while incorporating correlated data. This method is effective only when training data includes large number of features.

The above survey emphasizes that precise assessment of request classification for the improvement of QoS is not investigated in server farms. So this paper proposes a classifier, SCEW(SVM Classifier based on Expected Waiting time) for dispatcher. SCEW classifies the requests by predicting the load and waiting time at different servers. Based on this categorization SCEW selects the appropriate server for each request.

The rest of the paper is organized as follows. Section 3 introduces multiclass SVM model and the proposed SVM based user request classification model. The simulated results are analyzed in Section 4 and finally Section 5 summarizes the work.

3. PROPOSED MODEL

The SVM model is applied to categorize the input sample into two classes. The proposed traffic classification requires more than two classes i.e., the requests that are to be executed in a server are grouped as a separate class. Since there are many servers in server farm, multi-class classification is used.

3.1 Multi-class Classification

There are various ways to decompose a multiclass problem into number of binary classifications. The most commonly used are one-against-all, one-against-one, Error Correcting Output Codes(ECOC), Single machine and Divide-and-Conquer [12]. One-against-all: A SVM is built for every class. Each SVM is trained to identify samples of this class from the remaining classes. One-against-one: A SVM is built for each pair of classes. For a problem with ‘n’ classes, n*(n-1)/2 SVMs are built and trained to distinguish samples of one class from other. ECOC is implemented with more complex decompositions and results of Single machine and Divide-and-Conquer are very similar. One-against-one and other ECOC are more accurate than One-against-all [12]. So in the proposed model, one-against-one multiclass classification is used. In one-against-one classification method, n*(n-1)/2 SVMs are used. Samples are tested with all SVMs and combined with Max-Win algorithm. Max-Win algorithm makes each SVM to cast one vote for its categorized class. Then it finds the class which has the maximum votes. For example if a problem has three classes then three SVMs are needed, say SVM12 which is designed to distinguish class1 and class2, SVM23 distinguishes class2 and class3; SVM13 does for class1 and class3. All the SVMs are trained with their respective classes. While testing, sample is fed to all SVMs. Each SVM produces positive or negative result and Max-Win algorithm makes the SVMs to vote to their respective classes. Finally it finds the class that has maximum votes. Now the elected class is the class of the testing sample.

3.2 Proposed SVM based User Request Classification

Since SVM superiority has been proved already on various applications, it can also be applied to network flow identification. The proposed model adopts multiple network flow properties for classification which need higher dimensional mapping, so SVM is highly preferred for this proposal.

The proposed SCEW categorizes requests automatically using the features that are derived from traffic. Then it forwards the requests of the same category i.e., same class, to the same server. Alternatively, the requests may be classified pertaining to server IP, but, if the chosen server receives bulk requests then the response time will be poor. So SCEW includes expected waiting time as one of the features in feature vector. The expected waiting time is calculated for each request with the network flow features of input traffic. The SVM model is trained to classify the request based on least expected waiting time at servers. So the request is always routed to the server that is lightly loaded. This in turn minimizes the response time.

The proposed SCEW classifier includes two phases, training and testing. In training phase the captured traffic is preprocessed and network flow properties are constructed. These properties are processed and used as training data to build the SVM model. The preprocessing includes three steps, traffic selection, flow generation and feature vector construction. While testing, the unknown flow properties provide test data and tested with the proposed SCEW model.

3.2.1. Traffic Selection

For simplification and ease in analysis, this paper selects only limited applications traffic. Wireshark, a packet sniffer captures the message being sent/received from/by dispatcher. While packet capturing, the sniffer receives every link-layer frame that is sent or received by higher layer protocol such as HTTP, FTP, TCP, UDP, DNS or IP. Filter in Wireshark is modeled to select the traffic intended to specific destination and applications. The modeled filter recognizes application error and unseen packets and removes them. In the received traffic, there are duplicate packets and out of order packets due to communication delay and other interferences. The SCEW identifies all such packets and eliminates them. Then the selected traffic is further preprocessed to get feature vector.

3.2.2. Flow Generation

The captured traffic has many unidirectional flow records. This unidirectional flow record describes the packet that is sent from user to server or server to user. All the unidirectional flow records pertaining to a request are collected in sequence. This sequence is called network flow or flow. The flow starts with arrival of request and ends with final acknowledgement. The flow also includes all the related communications between user and the server pertaining to that request. Similarly flow of each request is generated. Here some requests may not be answered due to unavailability of services or within the traffic capturing time the request may not be completed. These types of flows are identified and removed by SCEW. The remaining flows that are complete are processed to construct training data (feature vectors).

3.2.3. Construction of Feature Vector

Each flow includes parameters like source IP and port, destination IP and port, arrival time, frame length etc. These network parameters are designated as parameters of feature vector. Few parameters are directly available in the packet header. Other parameters can be computed with the information that is available in the packet headers of each flow, called as network flow features. Table1 includes all the features that are computed while preprocessing the packet header information. Since all features do not influence classification, all the 12 parameters need not be used as features and also inclusion of many features may increase the time of classification. So, SCEW selects the features by incremental method. Initially basic three features are added and classification is performed and the performance is measured with precision value. Then each feature is combined and the classification performance is analyzed. The feature set that has best performance is chosen as feature vector for further classification. [server IP, server port, inter-arrival time, expected waiting time]

Table 1. Network flow features

Flow id	Network flow features
1	Number of packets in the flow
2	Number of packets sent for the flow
3	Average packet size in the flow
4	Duration of the flow
5	Protocol
6	Source port of the flow
7	Destination port of the flow
8	Inter-arrival time
9	Destination IP
10	Source IP
11	Waiting time
12	Arrival time

Table 2. Traffic category and Flow

Traffic cate-gory	Appli-cation	No. of flows
Bulk	ftp	202
WWW	http, https	637
Multi-media	skype	121

3.2.4. SVM Training and Testing

With the constructed feature vectors, SVM model is built by SCEW. First step in training is to setup the SVM model with SVM parameters. The parameters are type, coefficient, degree, gamma, cache size and are explained as follows:

Type: there are 5 types of svm, i.e C-SVM, nu-SVM, one-class-SVM, epsilon-SVm, nu-SVM regression. Regularization is simple in C-SVM and it also minimizes classification error [5], so this paper prefers C-SVM. *Coefficient*, *degree* and *gamma* are the parameters of kernel functions. *nu* is used in nu-SVC, nu-SVR and one-class-SVM. *Cache size* represents kernel cache size. *eps* is epsilon and this describes the tolerance of terminating criterion. *probability* is probability estimate that obtains probability prediction of SVM classification. C controls the tradeoff between error on SVM and overfitting. The feature vectors are transformed into the form,

<label>:<index1>:<value1>;……<index n>: <value n>

label is the class label of SVM; 0.0 for class 1, 1.0 for class 2 etc

'n' is number of features in feature vector. index and value represent each feature in feature vector and its value. Next step is scaling. The values in training sample are exactly converted into the range [-1,1]. Only the data points which define hyper plane are support vectors, so the number of support vectors is less than number of training points.

Then the trained SVM model is tested using 'Predict' and Predict probability methods. These methods examines the probability of prediction on test data against each class. The class with highest probability is the class of the test data.

4. SIMULATION AND PERFORMANCE ANALYSIS

The network traffic for experimentation was captured by the packet sniffer tool, Wireshark for limited period. Proposed SCEW identifies out of order and error packets and eliminates them. SCEW also generates the network flow using Java platform. After processing these filtered flow, feature vector (training data) are constructed. With the training data, SCEW generates and trains the SVM model using LibSVM, Java based tool. Other classifications like random forest and deep learning is planned to be implemented in the future work.

Performance measures that evaluate the proposed system are precision, probability estimate and waiting time [6]. *Precision* for a class is the ratio of number of data correctly classified that belongs to this class to the total number data belongs to this class. *Probability estimate* is predicted probability that a testing data belongs to its class. *Waiting time* is the total time that the request has been waiting at server queue.

Assumption: server farm has 3 homogenous servers.

The performance of SVM model relies on the kernel functions [5]. Three commonly used kernel functions are taken for analysis i.e., Linear, Polynomial and RBF. These kernels create three different non-linear separation surfaces. Commonly used three different traffic categories are focused in this paper, WWW, Bulk and Multimedia categories. All the three categories are captured and filtered. Using Java program the flows are identified. Table 2 describes the number of flows computed on different categories. The requests received during capturing period on WWW category is more than other categories. So number of flows in this category is large. Though the capturing period is large, the number of requests received on multimedia traffic category is low. So the number of flows is also less. Bulk traffic (FTP) requests are more than multimedia, but capturing period is short.

4.1. Penalty Parameter C and gamma

For better performance and precision of SVM model, the penalty parameter C should be selected properly. This paper uses v-fold cross validation method to select C and RBF parameter 'gamma'. For this validation, 100 training samples are selected from each traffic class, with v=10 and v-fold cross validation method is applied. The C and 'gamma' values are noted for various classification precision. The SVM classification always aims for higher precision, so during cross validation process, values of C and 'gamma' are noted when the precision is 94% and 100%. Contour map is drawn and it is shown in Figure 1. From the figure, it is observed that classification performance is stable over a range of C and 'gamma' values. Any value in this range may be chosen. To avoid hard margin, the values of C and gamma are chosen as C=1024 and 'gamma'=0.00012. These values are used for further experiments.

4.2. Precision and Probability Prediction

The experimental results shown in Table 3 and 4 are obtained with an unbiased sample data i.e. the traffic is captured on random period of time by Wireshark. But this capturing is limited to specific applications and servers, assuming only 3 servers in server farm. For easy analysis, only one application traffic per category is taken. Filtered traffic from Wireshark is further processed using Java program and flows are generated. Network flow properties are computed from flows and transformed as training data. For each category of traffic separate SVM model is necessary, because number of flows, values of network flow properties and period of traffic capture are not uniform in all the three traffic categories. For three traffic categories listed in Table 2, separate SVM models are trained with the respective training data.

Purpose of this classification is to reduce the waiting time spent on queue before execution at server. So the SVM model is trained to minimize the waiting time. While testing, new arrival request is classified by this SVM model and according to the classification, it is routed to one of the server. Certainly the request will be routed to the server which is lightly loaded. Similarly all the test data are classified and routed to the proper server. Every time TP(True Positive) and FN(False Negative) are noted and precision is calculated. These are listed in the Table 3.

Table 3. Precision and penalty parameter

	HTTP traffic			Multimedia traffic		
Kernel Function	Precision (%)	C	Gamma	Precision (%)	C	Gamma
Linear	100	1	0	100	1	0
Poly	100	200	0.0001	90	512	0.0001
RBF	100	8192	0.00003	95.83	8192	0.00003
RBF	94	512	0.00024	92.5	1024	0.00012

It is inferred from Table 3 and 4 that precision of HTTP traffic is higher than FTP and Multimedia. At the first glance, this shows that when number of training samples is more, classification precision is also high. But network characteristics are different in different applications and also the precision is achieved by applying different C and 'gamma' values. For further analysis, probability estimates are measured for testing data. Table 5 shows the range of probability estimates on each model with various traffic categories. This shows how the model decides the class of a traffic sample on different kernel functions against three traffic categories. Though polynomial kernel function achieves 100% precision for HTTP traffic presented in Table 3, it classifies that data only on the probability of 0.54 which is presented in Table 5. Whereas for the same HTTP traffic, precision of RBF kernel is only 94% but it classifies with the probability of 0.94. So RBF kernel is virtuous

Table 4. Precision and penalty parameter

	FTP traffic		
Kernel Function	Precision (%)	C	Gamma
Linear	97.5	1	0
Poly	95.8	512	0.0001
RBF	98.83	8192	0.00003
RBF	94.16	1024	0.00012

than polynomial kernel. For multimedia traffic, polynomial kernel delivers 90% precision with the probability of 0.999. It shows that all TP are classified with maximum probability. For the same multimedia traffic, RBF kernel achieves 92.5% precision with 0.99654 probability, which is very close to the probability achieved with polynomial kernel. For FTP traffic also RBF kernel produces higher probability classification result than other two kernels. Considering all the three traffic categories, probability of prediction is better in RBF kernel and also precision is not less than 94%. So RBF kernel is better than other two. If C = 8192, then RBF kernel produces classification precision in HTTP = 100%, FTP =98.8%, Multimedia = 98.8%. But this large C makes the SVM model as hard margin model. This leads to less generalized model when applied to testing data. So C is fixed as 1024 which produces soft margin model.

4.3. Classification and Waiting Time

While analyzing the application traffic on each category, it is found that the frequency of receiving the requests on HTTP traffic is high and on multimedia traffic, it is low. So the capturing period of multimedia category is increased to get more requests. For the same reason the FTP traffic capturing period is also increased. The network flow parameters are computed with the recorded flows shown in Table 2. These parameters structure the feature vector that trains the SVM model. As discussed in the previous section, 4.2

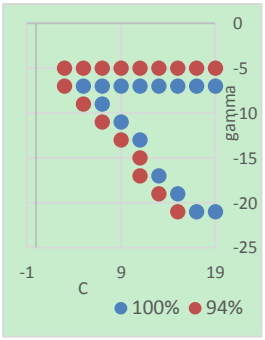

Table 5. Probability of Classification

Kernel function	HTTP traffic	Multimedia traffic	FTP traffic
Linear	0.9502	0.921	0.921
Poly	0.5429	0.9997	0.9896
RBF	0.9397	0.9965	0.9986

Figure 1. Cross validation

the model is trained with RBF kernel function and C=1024. For each test data (request), the model estimates the expected waiting time of each server. The request can be executed at any server in the server farm. In this situation, the concern of the classification is to reduce the waiting time. So the classification is based on waiting time, i.e., the request is always forwarded to the server that has least waiting time. The waiting time of each request is noted and plotted in Figure 2 for WWW traffic category. Similarly waiting time for each request in other two categories, Bulk, and Multimedia are also noted and plotted in Figure 3 and Figure 4 respectively. It is inferred from these figures that the waiting time of the requests that are classified using SVM is linearly increasing. Waiting time of the requests without any classification is not linear. To evaluate the advantage of SVM classification, average waiting time for different arrival rate is computed and plotted in Figure 5. It is inferred from this figure that regardless of traffic category average waiting time of classified requests are always less than non classified requests. Service time of Multimedia traffic is high, so the waiting time is also higher than other two categories. But it is lesser than the

waiting time of non classified request. Similarly service time of FTP traffic category is higher than HTTP traffic. Again the waiting times of non classified request on FTP and HTTP categories are higher than SVM classified requests. Minimizing the waiting time by SVM classification affords good outcome in total response time.

5. CONCLUSION

Classification of the requests facilitates to achieve perfect load balancing in server farm. SVM based traffic classification is proposed to identify a suitable server for every request. The proposed SVM Classifier based on Expected Waiting time, categorizes the requests using the request type, current load of the server, and expected waiting time at different servers. This classifier proficiently forwards the requests to the least loaded server, this in turn minimizes makespan. Traffics are captured using Wireshark and simulations are carried out using LibSVM. It is inferred from the experimental results that there is performance enhancement of the proposed classifier compared to traditional dispatcher without classification.

6. REFERENCES

[1] Dipti Tiwari and Bhawna Mallick. 2016. SVM and Naïve Bayes Network Traffic Classification using Correlation Information. *International Journal of Computer Applications*, 147, 3, (Aug. 2016) DOI: 10.5120/ijca2016911010

[2] Du Jiang and Long Tao. 2013. P2P traffic identification research based on the SVM. In *Proceedings of the Wireless and Optical Communication Conference (WOCC*, 2013), 683 – 686.

[3] Ehsan Saboori, Shahriar Mohammadi and Shafigh Parsazad. 2010. A new scheduling algorithm for server farms load balancing. In *Proceedings of the 2nd Int. conference on Industrial and Information Systems (IIS*, 2010)

[4] Gabriel Gómez Sena and Pablo Belzarena. 2009. Early traffic classification using Support Vector Machines. In *Proceedings of the 5th Int. Latin American Networking Conference (LANC '09)*.

[5] Hyeran Byun and Seong-Whan Lee. 2002. Applications of Support Vector Machines for Pattern Recognition: A Survey. In *Proceedings of the 1st Int. Workshop on SVM Pattern Recognition with Support Vector Machine.* (2002), 213-236.

[6] Jeffrey Erman, Anirban Mahanti, and Martin Arlitt. 2006. Internet Traffic Identification using Machine Learning. In *Proceedings of the 49th IEEE Global Telecommunications Conference* (Dec. 2006), GLOBECOM 2006.

[7] Jie Cao and Zhiyi Fang. 2016. Network Traffic Classification using Genetic Algorithms based on Support Vector Machine. *International Journal of Security and Its Applications*. 10, 2, (2016), 237-246.

[8] Jun Tan, Xingshu Chen and Min Du. 2012. An Internet Traffic Identification Approach Based on GA and PSO-SVM. Journal of Computers. 7, 1,(Jan 2012), 19-29.

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

[10] Ruixi Yuan, Zhu Li, Xiaohong Guan and Li Xu. 2010. An SVM-based machine learning method for accurate internet traffic classification. *Information System Frontiers.* 12, 2, (Apr.2010), 149-156.

[11] Shengnan Hao, Jing Hu, Songyin Liu, Tiecheng Song, Jinghong Guo and Shidong Liu. 2015. Network Traffic Classification Based on Improved DAG-SVM. In *Proceedings of the Int. Conference on Communications, Management and Telecommunications* (ComManTel 2015).

[12] Tzu-Kuo Huang, Ruby C. Weng and Chih-Jen Lin. 2006. Generalized Bradley-Terry Models and Multi-class Probability Estimates. *Journal of Machine Learning Research*. 7 (2006), 85–115.