# A Distributed Fuzzy Support Vector Machines Model for Real Network Traffic

JIANG Jie<sup>1+</sup>, QU Hua<sup>1</sup>, ZHAO Jihong<sup>1,2</sup> and ZHANG Yanpeng<sup>1</sup>

<sup>1</sup>Xi'an Jiaotong University, Shaanxi Xi'an, China. <sup>2</sup>Xi'an University of Posts & Telecommunications, Shaanxi Xi'an, China.

**Abstract.** Traffic classification has been widely applied for networking. Previous works paid little attention to robustness and massive data of practice network traffic. In this paper, we propose a new strategy for the Random Fuzzy Support Vector Machines(RA-FSVM) based on fuzzy membership, as well as employ genetic algorithm to find optimal parameters. Moreover, we introduce the distributed idea to structure clusters, so as to speed up RA-FSVM model's computation. The results of simulation demonstrate the robustness and timeliness of RA-FSVM model.

Keywords: Network traffic; Fuzzy Support Vector Machines; Genetic algorithm; Distributed computation

# 1. Introduction

Network traffic [1] classification has drawn plenty of attention and plays an important role in network resource management, intrusion detection and other fields. Support Vector Machine(SVM) [2] developed by V.N. Vapnik, which aims at seeking for an optimal separating hyperplane to classify two classes. The SVM has shown a better performance than other traditional methods in many areas, such as optimal classification and image classification.

However, for a real network, network traffic usually contains mixed noises, outliers and samples with some eigenvalues missing, that may cause low precision or overfitting. Fuzzy Support Vector Machine (FSVM) [3], proposed by professor Chun-fu Lin, solves this problem by the fuzzy membership. The fuzzy membership is associated with each training sample which makes noises, outliers and deficiency training samples with less contribution for classification. Hence deviant training samples make less influences in iteration. In FSVM, decision function also bases on maximization the margin between two class, but employs small weight to filter abnormal data. As a result, training samples with fuzzy membership can provide higher robustness and performance in practice network. The fuzzy membership function [4] is a key problem in FSVM, so as to determines the performance of the algorithm. In the previous works, Lin and Wang [3] proposed two strategies to define fuzzy membership: kernel-target alignment and using K-NN. Wang et al. [4] presented bilateral-weighted FSVM, which bases on the vague sets. However, above fuzzy membership algorithms can't distinct the normal samples with abnormal samples. For another, Inoue [5] applied FSVM on multiclass problem. But the above research paid little attention to self-adaption of parameters and timeliness in massive data.

In this paper, a new algorithm, called RA-FSVM, is proposed to improve traditional FSVM and make it suitable for the era of massive data. Firstly, we propose a new fuzzy membership function based on an improved K-means clustering algorithm and apply the distance between training samples and the nearest cluster's center to get the fuzzy membership. Further, we apply random forest method to getting a globally optimal solution. Secondly, we employ genetic algorithm to get the optimal Support Vector(SV). The

<sup>&</sup>lt;sup>+</sup> Corresponding author. Tel.: +8618375331885; *E-mail address*: jkikyou@stu.xjtu.edu.cn

advantages of genetic algorithm are that we can simply "negate" some individuals with awful performance instead of trying to find the optimal solution initiatively, as well as we can get adaptive parameters. Lastly, we achieve the distribution by getting each support vector in the cluster. An important property of SVM [6] is that the final model only relates to SV [5], so as to we can clearly find that distributed computing is very suitable for processing large amounts of data based on this method. Hence, we add the distributed idea to speed up the algorithm calculation speed in massive data, as well as decouple training process and real-time classification process, so as to realize the classification of real network traffic with robust.

• The main contribution of this work is threefold: we propose a novel strategy for setting the fuzzy membership in RA-FSVM.

• we introduce the distributed idea to build clusters, so as to speed up RA-FSVM model computation.

• we employ genetic algorithm to adaptively find optimal parameters.

The rest of this paper is organized as follows. In section 2, we briefly review FSVM and genetic algorithm. In section 3, we propose a novel model, called by RA-FSVM, for practice network traffic. In section 4, we detail our experimental results. We conclude the paper in Section 5.

# 2. FSVM and genetic algorithm

### 2.1. SVM and FSVM

In this section, we introduce SVM and FSV [7], as well as show the difference between SVM and FSVM.

Given the training samples  $D = \{(x_1, y_1, s_1), ..., (x_n, y_n, s_n)\}$ , where  $x_i \in \Box^d$ ,  $y_i \in \{-1, 1\}$ ,  $\varepsilon \le s_i \le 1$ ,  $s_i$  is the fuzzy membership for output  $y_i$  and the  $\varepsilon$  is the threshold value. The optimal SVM's hyperplane problem is:

$$\min_{w} \frac{1}{2} w^{T} w + C \sum_{i}^{n} \xi_{i}$$
Subject to  $y_{i}(w^{*} x_{i}) \ge 1 - \xi_{i}, \xi_{i} \ge 0, i = 1, ..., n$ , (1)

where *C* is a constant, regarding as a regularization parameter which can make balance between classification error and margin maximization [8]. The optimal SVM's hyperplane is a quadratic programming problem, so we introduce the Lagrange factor  $a = (a_1, ..., a_n)$  and kernel function  $k(x_i, x_j)$  to get the dual problems:

$$\max_{a} \sum_{i}^{n} a_{i} - \frac{1}{2} \sum_{i}^{n} \sum_{j}^{n} a_{i} a_{j} y_{i} y_{j} k(x_{i}, x_{j})$$
  
Subject to  $\sum_{i}^{n} a_{i} y_{i} = 0, 0 \le a_{i} \le C, i = 1, ..., n$ . (2)

As to FSVM, the optimal hyperplane formula(1) can rephrase as follow:

$$\max_{a} \sum_{i}^{n} a_{i} - \frac{1}{2} \sum_{i}^{n} \sum_{j}^{n} a_{i} a_{j} y_{i} y_{j} k(x_{i}, x_{j}) ,$$
  
Subject to  $\sum_{i}^{n} a_{i} y_{i} = 0, 0 \le a_{i} \le s_{i} C, i = 1, ..., n$ . (3)

We set lager *C* to get less misclassification and narrower margin, so that we can control the importance of training samples by  $s_i$ . Namely, lager  $s_i$  makes related training sample  $x_i$  get more effects for process of classification. The  $x_i$  names as support vector(SV), when condition is met  $0 < a_i \le s_i C$  and names as the misclassification sample when condition is met  $a_i > s_i C$ .

#### 2.2. Genetic algorithm

Genetic algorithm [9] is a method to search for the optimal solution by simulating the biological evolution process of natural selection and genetic mechanism. According to the principle of the fittest survival and the evolution, we set potential solutions as the original population, then we employ fitness function and genetic factors, which combined with the crossover and mutation of gene, to produce more the approximate optimal solution. The iteration until the approximate optimal solution meets the requirements,

so as to get the optimal parameters of the adaptive selection. The Architecture of Genetic algorithm show as Fig. 1.



Fig. 1: Architecture of genetic algorithm

## 3. RA-SVM model for network traffic

In this section, we describe RA-SVM model for network traffic classification. Firstly, we introduce the architecture of distributed RA-SVM model, then we introduce an improved membership function based on K-means to get fuzzy membership function, after that we focus on the genetic algorithm to get self-adaption of parameters, finally we design a distributed scheme to classification to speed up the algorithm calculation in massive data.

### 3.1. Architecture of RA-FSVM model

Fig. 2 shows that the recognition flow chart of RA-FSVM. The RA-FSVM model consists of two parts: the optimal SV acquisition and the classification part. In optimal SV acquisition, we store data pre-processing<sup>[10]</sup> network traffic data in Hadoop<sup>[11]</sup> Distributed File System (HDFS), then execute an improved K-means algorithm and Genetic Algorithm in distributed cluster to find the optimal SV. In classification part, we employ the total SV, receiving from the training model, to classify the network traffic.



Fig. 2: Architecture of RA-FSVM model

#### **3.2.** A new membership function of FSVM

Commonly, we adopt training samples associated with the fuzzy membership to training SVM model. The selection of fuzzy membership function is a key problem in FSVM, so as to determines the performance of the algorithm. Therefore, we propose a new fuzzy membership function to fit practice network traffic better. Namely, we adopt to an improved K-means<sup>[12]</sup> to find center of each cluster, and the membership is the distance between training samples and its nearest class center in feature space. The idea of random forest

is introduced to get globally optimal solutions. Heuristically, we suggest that the k of K-means<sup>[13]</sup> clustering algorithm is match to the number of unique training labels, and initialized centers are certainly training samples for each class, so as to achieve the optimal clustering effect. The average of  $x_i$  defined as following:

$$\mathbf{x}_{l} = \frac{1}{z} \left( \mathbf{x}_{l1} + \dots + \mathbf{x}_{lz} \right), \tag{4}$$

where z is the number of cross experiments.

The class centers  $\mathbf{x}_l$ , l = 1,...,m, where *m* is the total number of cluster label. The maximum radius of each cluster is  $r = \max \|\mathbf{x}_i - \mathbf{x}\|$ , and the measure of membership is

$$\delta = \left(1 - \frac{x_i - x_l}{r}\right)^2, i = 1, ..., n, l = 1, ..., m \quad .$$
(5)

The sample closer to the center, the higher value of  $\delta$ . Namely, that the confidence of  $x_i$  is higher. Thus, we define the fuzzy membership function as follows:

$$s_{i} = \begin{cases} \varepsilon_{1}, \delta \leq \varepsilon_{1} \\ \varepsilon_{1} + (1 - \varepsilon_{1}) \left( 1 - \frac{x_{i} - x_{l}}{r} \right)^{2}, \varepsilon_{1} \leq \delta \leq \varepsilon_{2}, \\ 1, \delta \geq \varepsilon_{2} \end{cases}$$
(6)

where i = 1, ..., n, l = 1, ..., m,  $\varepsilon_1, \varepsilon$  are constant and  $\varepsilon_1$  is a small constant to ensure  $s_i > 0$ .

### **3.3.** Kernel function of RA-FSVM

Aiming at non-linear training samples, we choose Gaussian kernel<sup>[14]</sup> function to fit RA-FSVM. The kernel function shows as following:

$$K(x_i, x_j) = \exp\left(-\frac{\|s_i x_i - s_j x_j\|}{2\sigma}\right),\tag{7}$$

where  $\sigma > 0$ , and  $\sigma$  is the bandwidth of the Gaussian kernel. Hence, the optimal hyperplane formula(3) can rewritten as follow:

$$\max_{a} \sum_{i}^{n} a_{i} - \frac{1}{2} \sum_{i}^{n} \sum_{j}^{n} a_{i} a_{j} y_{i} y_{j} k(s_{i} x_{i}, s_{j} x_{j}) ,$$
  
Subject to  $\sum_{i}^{n} a_{i} y_{i} = 0, 0 \le a_{i} \le s_{i} C, i = 1, ..., n ,$  (8)

### 3.4. Distributed deployment of RA-FSVM

Nowadays, clustering plays an outstanding role in many aspects such as scientific analysis, machine learning, natural language processing and others. HDFS and Spark are the most popular open source platforms for big data era. Iteration of HDFS generates reads and writes repeatedly, so as to waste lots of resources. However, the Spark based on memory, so as to more suitable for iterative operations. Moreover, the final model is only related to SV in SVM. Therefore, we design distributed deployment based on Spark <sup>[15]</sup>.

This cluster consists of five stand-alone computer: one [M1] for resource allocation and task management, one [M2] for practice classification, three [M3,M4,M5] for seeking optimal SV. In the training stage, M1 evokes M3,M4,M5. Then each computer, as a resilient distributed dataset(RDD), executes an improved K-means with 1/3 training samples. After this operation, we get cluster centers  $x_i$ , as well as the fuzzy membership  $s_i$ . Then training an improved FSVM based on  $s_i$  to get initial classification. Furthermore, we carry out the GA with initial classification to get optimal parameters { C,  $a = (a_1,...,a_n)$ ,  $\varepsilon_1$ ,  $\varepsilon_2$ ,  $\sigma$  } and the SV. Finally, returns the union of three SVs. In the classification, M2 accepts total SVs to classify the network traffic.

# 4. Experiments

### 4.1. Experiments environment and data set

This paper uses PyCharm as an experiment platform and Python to carry out. We call SVM, decomposition, Feature\_selection, K-means under the Sklearn package and GAFT for training FSVM. Hardware configuration: Dell laptop, 8G memory, core i5 processor.

Moore data set [16] was used in the experiment, which is composed of 10 data subsets, namely, Data\_set={entroy01...,entroy10}. The Data\_set has 377526 network traffics, and each sample data is composed of 248 dimensional characteristics and 1 label. For the convenience of calculation, the labels in the samples are mapped to numbers and we define 8 classes to test the accuracy. The detailed classification of Moore set is shown in Table 1.

Table 1: The detail classification of Moore Set					
Classification	Application Type	Number of Samples			
WWW	http, https	328,091			
MAIL	imap, pop, smtp	28,567			
BULK	ftp-Control, ftp-Pasv, ftp-Data	11,539			
SERVICES	X11, dns, ident, ldap, ntp	2099			
DATABASE	postgres, sqlnet oracle, ingres	2648			
MUITIMEDIA	Windows Media Player, Real	1152			
P2P	KaZaA, BitTorrent, GnuTella	2094			
ATTACK	worm, virus	1793			

### 4.2. Specific experiment

#### 1) Experiment to balance the Moore Set

Table 1 reports the detail classification of Moore Set, we notice the count of WWW class is much larger than others. Thus, we apply undersampling<sup>[17]</sup> to further reduce WWW class samples, as well as to execute SMOTE [18] algorithm to oversampling the SERVICES, DATABASE, MUITIMEDIA, P2P and ATTACK class. The result of undersampling is yield 32800 WWW samples, 396 SERVICES samples, 10592 DATABASE samples, 4068 MUITIMEDIA samples, 8376 P2P samples and 7172 ATTACK samples as processed Moore Set. Fig. 3 shows the class distribution of processed Moore Set. In addition, we performed experiment on dataset of the raw Moore Set and the processed Moore Set. Fig. 4 shows the results of our simulation, which further confirms the advantages of the processed Moore Set.



Fig. 3: The class distribution



2) Experiment to find optimal features based on PCA In this experiment, we apply PCA<sup>[19][21]</sup>to remove irrelevant and redundant features from the processed Moore Set. The number of features changes from 10 to 50 and changes from 13 to 18. Fig. 5 shows the results of our test. We notice the accuracy of RA-FSVM is better than others when the number of features equals 16. Consequently, we set the number of features as 16 in next experiments.



Fig. 5: The accuracy of RA\_FSVM with different features.(a) from 10 to 50.(b) from 13 to 18.

## 3)Experiment to confirm the robustness of the RA-FSVM

We conduct simulations of SVM<sup>[20]</sup>, FSVM<sup>[21]</sup> and RA-FSVM using the same sample sets. For each sample set, we make sure of the independent between training set and test set. The miss rate  $\tau$ , which is simulates the missing eigenvalues to verify the robustness of the algorithm, is chosen such that 0%, 1%, 3%, 5%, 7%, 10%, and 15% of test set. Namely, we randomly change the character value of test to test the robustness of the RA-FSVM. Table 2 shows the results of our simulations. The accuracy of SVM and FSVM dropped rapidly with the decline of  $\tau$ . By checking the accuracy of the algorithms, we find that RA-FSVM performs better than SVM and FSVM in the test set when the  $\tau$  is more than 3%. More intuitively, Fig. 6 shows the trend of decreasing accuracy of each algorithm.

τ	Algorithm	$\mathcal{E}_1$	ε2	С	σ	Accuracy	Ranks
0%	SVM	-	-	7	0.125	96.5%	1
	FSVM	-	-	122	0.011	95.2%	2
	RA-SVM	0.12	0.97	56	0.032	94.7%	3
	SVM	-	-	7	0.125	94.2%	3
1%	FSVM	-	-	122	0.011	95.7%	1
	RA-SVM	0.12	0.97	56	0.032	94.3%	2
	SVM	-	-	7	0.125	87.5%	3
3%	FSVM	-	-	122	0.011	90.2%	2
	RA-SVM	0.12	0.97	56	0.032	91.3%	1
	SVM	-	-	7	0.125	82.3%	3
5%	FSVM	-	-	122	0.011	87.4%	2
	RA-SVM	0.12	0.97	56	0.032	89.6%	1
	SVM	-	-	7	0.125	76.7%	3
7%	FSVM	-	-	122	0.011	83.5%	2
	RA-SVM	0.12	0.97	56	0.032	85.2%	1
	SVM	-	-	7	0.125	74.6%	3
10%	FSVM	-	-	122	0.011	82.1%	2
	RA-SVM	0.12	0.97	56	0.032	83.9%	1
	SVM	-	-	7	0.125	72.9%	3
15%	FSVM	-	-	122	0.011	77.6%	2
	RA-SVM	0.12	0.97	56	0.032	80.3%	1

 Table 2: The parameters used in SVM, FSVM and RA-FSVM, and the training data accounted for 90%



Fig. 6: The accuracy of each algorithm declines

4) Experiment to confirm the timeliness of the RA-FSVM

We carry out experiment on the accuracy and convergence time of each algorithm. In this simulation, the size of samples is chosen such that 5000, 10000, 20000, 30000 and 50000. Then, we execute SVM,FSVM, and RA-FSVM model and employ test sets with 5%  $\tau$  to get the accuracy. Further, each

experiment was performed three times, so as to get the average of the cost time and accuracy. Table 3 shows the results of our simulations. By checking the accuracy, we find the accuracy of RA-SVM is higher than SVM's and FSVM's in 5% missing eigenvalues of test sets. As well as checking the cost time, we find RA-FSVM based on distributed performs well, and the rate of convergence is nearly twice as fast as the other two algorithms. Further, according to the distributed computing characteristics, it is reasonable to believe that the convergence time of RA-FSVM model will be further reduced as the distributed cluster expands. More intuitively, Fig. 7 and Fig. 8 show the trend of the cost time and accuracy of each algorithm.

Index	Train	Test	SVM		FSVM		RA-FSVM	
	Sets	Sets	Cost_Time(s)	Accuracy	Cost_Time(s)	Accuracy	Cost_Time(s)	Accuracy
1	4500	500	172	79.2%	206	85.2%	95	85.6%
2	9000	1000	301	81.7%	327	85.7%	171	87.2%
3	18000	2000	536	82.2%	561	86.3%	269	86.5%
4	27000	3000	795	81.6%	823	86.1%	392	87.3%
5	45000	5000	1297	80.9%	1406	85.6%	612	87.9%

Table 3: The train sets accounted for 90% of the Data\_set and the test sets, which contains 5% missing eigenvalues, accounted for 10%.



Fig. 7: The cost time of each algorithm



Fig. 8: The accuracy of each algorithm with miss

These results show that our RA-FSVM model performs more robust when the dataset contains missing eigenvalues, as well as RA-FSVM based on distributed converges faster.

# 5. Conclusions

In this paper, we adopt a novel robust RA-FSVM model to classify practice network traffic, and propose a new strategy for setting the fuzzy membership in RA-FSVM, as well as employ genetic algorithm to find optimal SV. Moreover, we also introduce the distributed idea to build clusters, so as to speed up RA-FSVM model computation. The experiment results demonstrate the robustness and timeliness of RA-FSVM model. But distributed computing may damage the integrity of the data set and decrease the accuracy of RA-FSVM. Therefore, how to ensure the lossless decomposition of data set is the key for our next research.

## 6. Acknowledgements

This work was supported in part by the National Major Project under Grant No.2018ZX03001016, the National Natural Science Foundation of China under Grant No.61531013.

### 7. References

- Ji-hye Kim, Sung-Ho Yoon and Myung-Sup Kim, "Study on traffic classification taxonomy for multilateral and hierarchical traffic classification," 2012 14th Asia-Pacific Network Operations and Management Symposium (APNOMS), Seoul, 2012, pp. 1-4.
- [2]. C. Cortes, V. Vapnik, "Support Vector Networks", Machirie Learning, vol. 20, pp. 273-297, 1995.
- [3]. Chun-fu Lin and Sheng-de Wang, "Training algorithms for fuzzy support vector machines with noisy data," 2003 IEEE XIII Workshop on Neural Networks for Signal Processing (IEEE Cat. No.03TH8718), Toulouse, France, 2003, pp. 517-526.
- [4]. Y. Yang, J. Wang and Y. Yang, "Exploiting rotation invariance with SVM classifier for microcalcification detection," 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI), Barcelona, 2012, pp. 590-593.
- [5]. Z. Hao, L. Shaohong and S. Jinping, "Unit Model of Binary SVM with DS Output and its Application in Multi-class SVM," 2011 Fourth International Symposium on Computational Intelligence and Design, Hangzhou, 2011, pp. 101-104.
- [6]. Y. Yang, J. Wang and Y. Yang, "Improving SVM classifier with prior knowledge in microcalcification detection1," 2012 19th IEEE International Conference on Image Processing, Orlando, FL, 2012, pp. 2837-2840.
- [7]. C. J. C. Burges, "A tutorial on support vector machines for pattern recognition", Data Mining and Knowledge Discovery, vol. 2, no. 2, pp. 121-167, 1998.
- [8]. J. Ni, Q. Zhang, L. Su, J. Liang and W. Huo, "L1/2Regularization Sar Imaging Via Complex Image Data: Regularization Parameter Selection for Target Detection Task," IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, 2018, pp. 2298-2301.
- [9]. "IEE Colloquium on 'Genetic Algorithms for Control Systems Engineering' (Digest No. 1993/130)," IEE Colloquium on Genetic Algorithms for Control Systems Engineering, London, UK, 1993, pp. 0\_1-.
- [10]. X. Wang and J. Su, "Research of Distributed Data Store Based on HDFS," 2013 International Conference on Computational and Information Sciences, Shiyang, 2013, pp. 1457-1459.
- [11]. Liu Jiang, Bing Li and Meina Song, "THE optimization of HDFS based on small files," 2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT), Beijing, 2010, pp. 912-915.
- [12]. R. M. Esteves, T. Hacker and C. Rong, "Competitive K-Means, a New Accurate and Distributed K-Means Algorithm for Large Datasets," 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, Bristol, 2013, pp. 17-24.
- [13]. S. Banerjee, A. Choudhary and S. Pal, "Empirical evaluation of K-Means, Bisecting K-Means, Fuzzy C-Means and Genetic K-Means clustering algorithms," 2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), Dhaka, 2015, pp. 168-172.
- [14]. S. Banerjee, A. Choudhary and S. Pal, "Empirical evaluation of K-Means, Bisecting K-Means, Fuzzy C-Means and Genetic K-Means clustering algorithms," 2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), Dhaka, 2015, pp. 168-172.

- [15]. N. Chimpai and J. Parnklang, "The electrical effects of iridium and standard spark plug for simple spark ignition engine," 2018 3rd International Conference on Control and Robotics Engineering (ICCRE), Nagoya, 2018, pp. 70-73.
- [16]. T. Auld, A. W. Moore and S. F. Gull, "Bayesian Neural Networks for Internet Traffic Classification," in IEEE Transactions on Neural Networks, vol. 18, no. 1, pp. 223-239, Jan. 2007.
- [17]. S. Sabeti and J. B. Harley, "Guided wave retrieval from temporally undersampled data," 2017 IEEE International Ultrasonics Symposium (IUS), Washington, DC, 2017, pp. 1-4.
- [18]. F. Koto, "SMOTE-Out, SMOTE-Cosine, and Selected-SMOTE: An enhancement strategy to handle imbalance in data level," 2014 International Conference on Advanced Computer Science and Information System, Jakarta, 2014, pp. 280-284.
- [19]. W. Puyati and A. Walairacht, "Efficiency Improvement for Unconstrained Face Recognition by Weightening Probability Values of Modular PCA and Wavelet PCA," 2008 10th International Conference on Advanced Communication Technology, Gangwon-Do, 2008, pp. 1449-1453.
- [20]. S. Alam, Moonsoo Kang, Jae-Young Pyun and G. Kwon, "Performance of classification based on PCA, linear SVM, and Multi-kernel SVM," 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, 2016, pp. 987-989.
- [21]. C.-F. Lin, S.-D. Wang, "Fuzzy support vector machines", IEEE Transactions on Neural Networks, vol. 13, no. 2, pp. 464-471, 2002.