International Journal of Innovative Computing, Information and Control
Volume **x**, Number **0x**, x **2007**

ICIC International ©2005 ISSN 1349-4198
pp. **0–0**

IMPROVING NEURAL NETWORK CLASSIFICATION USING FURTHER DIVISION OF RECOGNITION SPACE

Zhenxiang Chen and Haiyang Wang

School of Computer Science and Technology, Shandong University
73 Jingshi road, Jinan, Shandong, 250061, China
czx@sdu.edu.cn; why@sdu.edu.cn

Ajith Abraham

Centre for Quantifiable Quality of Service in Communication Systems
Norwegian University of Science and Technology, Trondheim, Norway
ajith.abraham@ieee.org

Crina Grosan

Department of Computer Science, Babes-Bolyai University
Cluj-Napoca 400084, Romania
cgrosan@cs.ubbcluj.ro

Bo Yang*,Yuehui Chen and Lin Wang

School of Information Science and Technology, University of Jinan
106 Jiwei Road,Jinan, Shandong, 250022, China
yangbo@ujn.edu.cn; wanglin_1983ujn.edu.cn; yhchen@ujn.edu.cn
* Corresponding author

Abstract. *Further Division of Recognition Space (FDRS) is a novel technique used for neural network classification. Recognition space is a space that is defined to categorize data sample after sample, which is mapped by neural network learning. It is divided manually into few parts to categorize samples, which can be considered as a line segment in the traditional neural network classification. In addition, the data recognition space is divided into many partitions, which will attach to different classes automatically. Experiments results using network traffic and intrusion detection data illustrate that the method has favorable performance especially with respect to the optimization speed and classification accuracy.*
Keywords: Classification, Neural network, Recognition space, Further division

1. **Introduction.** Classification is an important research area in data mining. Many classification techniques including decision tree [1, 2], neural network (NN) [3], support vector machine (SVM) [4, 5] and other rule based classification systems have been proposed. Neural network classification, which is supervised, has been proved to be a practical approach with lots of success stories in several classification tasks. However, its training efficiency is usually a problem, which is the current focus in this paper.

Many attempts have been made to speed up the convergence and improve the accuracy of neural network classification. Commonly known heuristic approaches such as momentum [6], variable learning rate [7] lead only to a slight improvement. Better results have been obtained with the artificial enlarging of errors for neurons operating in the saturation region [8]. A significant improvement on realization performance can be observed by using various second order approaches namely Newton's method, conjugate gradient's, or the Levenberg-Marquardt optimization technique [9], which is widely accepted as the most efficient one in the sense of realization accuracy. Generalization ability has been improved by using universal learning networks[10]. Fuzzy logic is integrated into the neural network learning algorithms to speed up the learning process[11]. Recently, several related methods in such topics were proposed to improve the neural network performance and then applied in some real applications [12][13][14].

In the conventional neural network classification, a sample from the original data set is mapped to a space between 0 and 1 (which is defined as recognition space) by the neural network. The sample is deemed as class 0 if it is more close to 0 than to 1. Otherwise, the sample belongs to class 1. If this space can be divided into partitions, then the mapped sample can get close to these partitions freely, and the mapping relationship (using a neural network) could be formed easily. Based on this idea, the performance of neural network classification including training speed and accuracy could be improved and this is the basic inspiration of our research.

This paper proposes a novel improvement of Neural Network classification, called Further Division of Recognition Space(FDRS). At first, the recognition space is defined. Then some rules are made to divide it into partitions, and the divided partitions are colored by using the proposed method. With the support of Particle Swarm Optimization (PSO) algorithm [15], a optimized neural network and colored recognition space is achieved. Then new samples could be classified based on the colored recognition space. Experiment results in several applications illustrate that the proposed method has favorable performance especially with respect to the optimization speed and the classification accuracy.

The remainder of this paper is organized as follows. The detailed theoretical model is described in Section 2. Section 3 illustrates the data classification method and application in intrusion detection and network traffic classification. Conclusion and future works are presented in Section 4.

2. Further Division of Recognition Space (FDRS).

First, the space which can be used to partition categories is defined as recognition space. Then, rules are defined to divide it into partitions. Further, the training samples are mapped to the recognition space by neural networks, and the recognition space is signed by the distribution of mapped samples. This is the defined by a coloring procedure. Finally, once the neural network has been optimized and its corresponding signed recognition space is formed, it can be applied for data classification.

2.1. Defining and Dividing the recognition space.

The recognition space is defined as a space that can be used to recognize samples as some class after what has been mapped or transmitted by conventional neural network. We expand the recognition space from traditional one dimensional to two dimensional space. In the proposed method, the recognition space is divided into many small partitions. If m equals 1, the divided partitions become small line segments. If m equals 2, the recognition space is divided into rectangle partitions. Analogously, cube partitions are achieved from the recognition space when m equals 3, and so on and the total partition number is obtained as follows:

$$TotalPartitionNumber = \prod_{d=1}^{m} partitionsnumber_d \tag{1}$$

Where $partitionsnumber_d$ is the number of partitions in d axis, $TotalPartitionNumber$ is the number of partitions in the whole recognition space.

2.2. **Mapping Training Samples to Recognition Space.** As traditional neural network classification, input samples can be mapped to output with some values in the recognition space and training data samples should be mapped to the expanded and divided recognition space. The forward propagation learning method for artificial neural network is selected as the mapping tool. The dimension of input data set vector is defined as n and dimension of recognition vector is defined as m. Every element of recognition vector is mapped by an isolated neural network from the same input vector $\vec{I}$. Target recognition space vector is $\vec{P}$. $\vec{P_i}$ is the i th element of $\vec{P}$ and $\vec{I}$ is an input data set vector. The mapping formula is given by (2).

$$P_i = NN_i\left(\vec{I}\right), i = 0...m-1 \tag{2}$$

When the recognition space is defined as a two dimensional space, two neural networks are used to decide the map position in the recognition space.

2.3. **Color Partitions to Sign Recognition Space.** For classification, different classes can be differentiated by different colors in the recognition space. So, the divided partitions are colored by different color to distinguish the classes. After partitions are colored, neural network and its corresponding colored recognition space are used for the evaluation of new samples training and classification.

2.3.1. *Select Color Points.* When coloring, the samples in the data set are mapped to the recognition space by Neural Networks. Therefore the whole mapped data samples can be selected to color the recognition space. Considering the generalization ability, only parts of them, which taken out by rate $DP(0<DP<1)$ are used to color the partitions. These selected and mapped samples are called color points. After DP was decided, there are two sequences to be used to generate color points in one class: order and random.

2.3.2. *Coloring Rules.* Each recognition space is blank after it is divided into partitions. When coloring, the following rules should be obeyed.
R1: Different classes must be colored with different colors.
R2: One or more partitions could be colored by one class.
R3: If color points of one class are majority(weighted) in one partition, this class will control the partition, and the partition should be colored by the color of related class.
R4: After the recognition space has been finish coloring, the partitions which still blank are called uncolored partitions. Normally training samples does not drop into these partitions and if they do, the partition is colored. So, the test points that fall into these partitions are called unclassified points. The corresponding sample can be considered as unclassified.

2.3.3. *Weights for Different Classes.* However, the data set is often uneven, so the weight of color points is an important factor for calculating the number of color points for each class. It is found that the category of majority in one partition should control the partition that it belongs to. Process that assigns partitions by category of majority is called partition coloring. But, it is unfair for every class data to color the recognition space with same weight, because the number of each class data in the data set is truly different. A class will control most area in the recognition space, if its number is much bigger than any other in the same data set. In order to solve this problem, a weight is defined. The weight of the first class should be smaller than second one, if the number of the first one is bigger than the second. The proportion of class c in the whole data set is defined as R_c as follows:

$$R_c = \frac{Num_c}{Num_{total}} \tag{3}$$

Where number of samples in class c is Num_c, and Num_{total} is the total number of samples in the whole data set. The weight of class c is calculated by:

$$W_c = \frac{1}{R_c} \tag{4}$$

2.3.4. *Coloring.* The coloring procedure should obey the defined rules and the coloring algorithm as depicted below:

```
Step 1:  Majority:=0
Step 2:  For C=0 to Number of Classes-1
         Begin
Step 3:  If (Color points' number of class C in this partition) *
(Weight of class C) > (Color points' number of class majority in
this partition)
* (Weight of class majority)
         Then majority = C
         End.
Step 4:  Partition Color: =Color(Majority)
```

2.4. **Training the FDRS.** Particle Swarm Optimization(PSO) algorithm [15] is employed to optimize neural network for its predominant features. FDRS is supervised, and so the evaluation of neural network is calculated at each generation. The correctness, which used as the fitness function of PSO, is defined as (5) and (6), where L is the total number of samples of the training data set:

$$Right = \begin{cases} 1, & Color(point) = Color(partition) \\ 0, & Else. \end{cases} \tag{5}$$

$$Correctness = \frac{\sum_{i=0}^{L-1} Right_i}{L} \tag{6}$$

3. **Experiment Results and Analysis.** After training an optimized neural network and its corresponding colored results are obtained, then the trained FDRS is applied to classify the samples. New sample should be mapped by this neural network, and then its category is judged according to the color of the partition. If the recognition color is not blank, this sample is categorized by its corresponding class of the color. Otherwise, the sample is seen as unclassified by FDRS.

All experiments were performed using 2 parallel processors with 2.4 GHz and 512MB of RAM. In order to evaluate the experiments, 4 criterions of Training Accuracy (TA), Generalization Accuracy (GA), Accuracy of Classified Samples (ACS) and Proportion of Unclassified Samples (PUS) are defined as follows:

$$TA = \frac{Number^{right}_{training}}{Number^{total}_{training}} \tag{7}$$

$$GA = \frac{Number^{right}_{testing}}{Number^{total}_{testing}} \tag{8}$$

$$ACS = \frac{Number^{right}_{testing}}{Number^{classified}_{testing}} \tag{9}$$

$$PUS = \frac{Number^{unclassified}_{testing}}{Number^{total}_{testing}} \tag{10}$$

$Number^{total}_{training}$ is defined as the number of total samples in training data, $Number^{right}_{training}$ is the number of right classified samples in the training data, $Number^{total}_{testing}$ is the number of total samples in testing data, $Number^{right}_{testing}$ is the number of right classified samples in the test data, $Number^{classified}_{testing}$ is the number of classified samples in the test data.

Where $Number^{unclassified}_{testing}$ is defined as the number of unclassified samples in the test data set.

3.1. **Intrusion Detection System (IDS).** Misuse intrusion detection uses well-defined patterns of the attack that exploit weaknesses in system and application software to identify the intrusions. The data for our experiments contains randomly generated 11982 records having 41 features [16,17,18]. This data set has five different classes: Normal, Probe, DOS, U2R and R2L. The training and test comprises of 5092 and 6890 records respectively. All the IDS models were trained and tested with the same set of data. As the data set has five different classes we performed a 5-class binary classification. The normal data belongs to class 1, Probe belongs to class 2, DOS belongs to class 3, U2R belongs to class 4 and R2L belongs to class 5.

Three layered forward propagation neural network with 20 hidden nodes was used. The selected parameters for this data set are: DP =0.5, m=2, $PartitionsNumber_0$ =10, $PartitionsNumber_1$=10, Max Generations = 1000, Population size =50, ϕ_1=0.02, ϕ_2=0.02 and VMAX =1. Test and training speed accuracies are used for evaluating the proposed method. For the IDS data set, 5 trails were made for each class to evaluate the mean (M) and Standard Deviation (SD). Experimental results of five different trials are illustrated in Tables 1-5. As an obvious advantage, the average performance of TA, GA

Table 1. Accuracy of Class 1 (Normal) in intrusion detection data

	Traditional Method			FDRS			
	TA	GA	ACS	TA	GA	ACS	PUS
1	97.52%	81.15%	81.15%	97.72%	81.97%	83.69%	2.06%
2	97.84%	77.07%	77.07%	98.32%	76.83%	76.83%	0.00%
3	96.18%	82.07%	82.07%	97.80%	99.65%	99.82%	0.17%
4	97.14%	85.7%	85.7%	98.44%	90.08%	90.08%	0.00%
5	97.88%	80.20%	80.20%	98.52%	38.05%	84.81%	55.13%
M	97.31%	81.24%	81.24%	**98.16%**	**83.32%**	**87.05%**	11.47%
SD	0.7%	3.12%	3.12%	0.39%	23.59%	8.56%	24.42%

Table 2. Accuracy of Class 2(Probe) in intrusion detection data

	Traditional Method			FDRS			
	TA	GA	ACS	TA	GA	ACS	PUS
1	99.34%	97.00%	97.00%	99.12%	97.96%	98.38%	0.43%
2	98.78%	94.80%	94.80%	99.14%	96.02%	96.29%	0.28%
3	98.92%	94.80%	94.80%	99.20%	87.57%	93.01%	5.85%
4	99.22%	92.40%	92.40%	98.76%	93.15%	93.15%	0.00%
5	98.76%	71.30%	71.30%	99.12%	90.86%	98.04%	7.32%
M	99.00%	90.10%	90.10%	**99.07%**	**93.11%**	**95.77%**	2.78%
SD	0.26%	10.60%	10.60%	0.18%	4.12%	2.58%	3.52%

Table 3. Accuracy of Class 3(DOS) in intrusion detection data

	Traditional Method			FDRS			
	TA	GA	ACS	TA	GA	ACS	PUS
1	89.92%	75.50%	75.50%	97.28%	75.00%	75.00%	0%
2	89.94%	68.60%	68.60%	94.34%	78.47%	79.09%	0.78%
3	93.36%	68.90%	68.90%	96.42%	88.55%	91.54%	3.27%
4	92.04%	81.00%	81.00%	97.62%	76.95%	76.95%	0.00%
5	95.84%	86.50%	86.50%	92.7%	58.15%	58.15%	0%
M	92.22%	76.10%	76.10%	**95.67%**	**79.42%**	**76.15%**	0.81%
SD	2.5%	7.75%	7.75%	2.09%	10.98%	11.95%	1.42%

and ACS of FDRS is much higher than the direct conventional method, and the SD of GA is also higher.

3.2. **Network Traffic Classification.** For further evaluation, the network traffic classification was selected. While this study focused solely on the TCP-based applications, only TCP flows were identified in the evaluation. We identified the start of a connection using TCP's 3-way handshake and terminated a connection when FIN/RST packets were received. In addition, we assumed that a flow is terminated if the connection was idle for over 5 seconds. The statistical flow characteristics considered include: number of packets in each direction, mean packets length in each direction, variance of packets length in each direction and duration [19]. The source of data is a public available packet trace

TABLE 4. Accuracy of Class 4(U2R) in intrusion detection data

	Traditional Method			FDRS			
	TA	GA	ACS	TA	GA	ACS	PUS
1	99.84%	99.30%	99.30%	99.84%	100.0%	100%	0%
2	99.84%	99.90%	99.90%	99.84%	99.20%	99.20%	0%
3	99.78%	100%	100%	99.78%	99.18%	100%	0.82%
4	99.82%	99.90%	99.90%	99.78%	99.92%	100%	0.08%
5	99.78%	100%	100%	99.86%	99.97%	100%	0.03%
M	99.81%	99.80%	99.80%	**99.82%**	**99.91%**	**99.84%**	0.19%
SD	0.03%	0.29%	0.29%	0.04%	0.42%	0.36%	0.36%

TABLE 5. Accuracy of Class 5(R2L) in intrusion detection data

	Traditional Method			FDRS			
	TA	GA	ACS	TA	GA	ACS	PUS
1	98.70%	99.10%	99.10%	99.29%	98.04%	99.05%	1.01%
2	96.01%	96.70%	96.70%	98.88%	98.48%	98.58%	0.10%
3	98.05%	98.90%	98.90%	98.80%	96.23%	97.62%	1.39%
4	98.40%	98.50%	98.50%	98.97%	97.97%	99.13%	1.16%
5	98.56%	98.40%	98.40%	99.27%	94.86%	97.92%	3.06%
M	97.94%	98.30%	98.30%	**99.04%**	**98.92%**	**98.46%**	1.34%
SD	1.11%	0.97%	0.97%	0.23%	1.53%	0.67%	1.08%

called Auckland IV. We used a subset of the Auckland IV trace from February 20, 2001 at 21:01:22 to February 21, 2001 at 02:00:00. There are 9575122 packets in this trace. Auckland IV traces include no payload information. Thus, to determine the connections "true" classifications port numbers are used. For this trace, we believe that a port-based classification will be largely accurate, as this archived trace predates the widespread use of dynamic port numbers. The classes considered for the Auckland IV data sets are HTTP, HTTPS, FTP(control), SMTP, POP3 and THE-OTHERS. We extracted samples randomly from data sets and formed two new data sets: one for training and another for testing. There are 2268 samples in training data set and 2268 samples in test data set.

We used a neural network with 30 hidden nodes. The best group of parameters used are as follows: DP=0.5, m=2, $PartitionsNumber_0$=10, $PartitionsNumber_1$=10, Max Generation=1000, Population Size=40,ϕ_1= 0.1,ϕ_2= 0.1 and VMAX=2. Tables 6-11 illustrate the results of accuracy test. The average performance of TA, GA and ACS on most classifications are higher by using FDRS. FDRS has favorable performance for traffic classification and at the same time, training speed is increased obviously by using FDRS.

4. **Conclusions.** In this paper, we presented a novel technique of Further Division Recognition Space(FDRS) model, which can improve the classification performance and accuracy by expanding and further dividing the recognition space. The traditional neural network classification could be regarded as a specific example of FDRS. Network traffic and intrusion classification problems were used to test the proposed method. Experimental results on the several data sets proved that the FDRS model is very effective. As

Table 6. Accuracy information of HTTP

	Traditional Method			FDRS			
	TA	GA	ACS	TA	GA	ACS	PUS
1	77.89%	77.31%	77.31%	80.00%	79.12%	79.33%	0.26%
2	78.68%	77.71%	77.71%	80.22%	77.76%	77.96%	0.26%
3	78.33%	78.33%	78.33%	79.91%	79.12%	79.33%	0.26%
4	77.84%	77.09%	77.09%	79.81%	78.39%	79.13%	0.93%
5	78.50%	77.53%	77.53%	79.98%	80.29%	81.33%	1.28%
6	78.77%	78.06%	78.06%	80.04%	78.02%	78.82%	1.01%
7	78.63%	77.62%	77.62%	79.91%	79.87%	80.01%	0.18%
8	79.03%	79.52%	79.52%	80.18%	78.90%	79.39%	0.62%
M	78.46%	77.90%	77.90%	**80.01%**	**78.93%**	**79.41%**	0.60%
SD	0.42%	0.76%	0.76%	0.14%	0.87%	0.97%	0.42%

Table 7. Accuracy information of HTTPS

	Traditional Method			FDRS			
	TA	GA	ACS	TA	GA	ACS	PUS
1	82.11%	81.45%	81.45%	81.61%	80.82%	80.89%	0.09%
2	82.07%	81.06%	81.06%	81.61%	80.69%	80.72%	0.04%
3	82.54%	78.22%	78.22%	82.72%	81.13%	81.13%	0.00%
4	81.31%	83.11%	83.11%	82.45%	79.19%	79.29%	0.13%
5	82.19%	83.32%	82.32%	81.83%	82.63%	82.81%	0.22%
6	82.10%	79.14%	79.14%	82.72%	82.41%	82.74%	0.40%
7	82.98%	82.72%	82.72%	83.69%	82.28%	82.57%	0.35%
8	81.48%	82.05%	82.05%	81.83%	82.01%	82.01%	0.00%
M	82.10%	81.26%	81.26%	**82.31%**	**81.39%**	**81.52%**	0.15%
SD	0.53%	1.74%	1.74%	0.73%	1.16%	1.23%	0.16%

Table 8. Accuracy information of FTP

	Traditional Method			FDRS			
	TA	GA	ACS	TA	GA	ACS	PUS
1	93.17%	94.13%	94.13%	97.22%	95.58%	97.00%	1.46%
2	93.65%	92.72%	92.72%	97.40%	96.43%	96.60%	0.18%
3	92.50%	93.61%	93.61%	95.94%	95.72%	95.72%	0.00%
4	92.33%	91.40%	91.40%	97.44%	95.98%	97.10%	1.15%
5	92.42%	93.17%	93.17%	96.56%	94.89%	94.93%	0.04%
6	96.96%	96.47%	96.47%	96.30%	95.72%	96.57%	0.88%
7	93.47%	94.09%	94.09%	97.18%	95.86%	95.86%	0.00%
8	93.12%	92.50%	92.50%	96.60%	93.51%	93.89%	0.40%
M	93.45%	93.51%	93.51%	**96.83%**	**95.46%**	**95.96%**	0.51%
SD	1.50%	1.50%	1.50%	0.56%	0.90%	1.11%	0.57%

Table 9. Accuracy information of POP3

	Traditional Method			FDRS			
	TA	GA	ACS	TA	GA	ACS	PUS
1	95.15%	96.83%	96.83%	96.38%	97.23%	97.70%	0.49%
2	95.02%	96.30%	96.30%	95.94%	96.91%	96.91%	0.00%
3	95.19%	94.00%	94.00%	96.52%	94.40%	94.40%	0.00%
4	94.58%	96.25%	96.25%	95.86%	96.43%	96.56%	0.13%
5	94.18%	95.81%	95.81%	95.63%	96.43%	96.47%	0.04%
6	94.49%	95.94%	95.94%	96.30%	97.05%	97.69%	0.66%
7	94.93%	96.47%	96.47%	95.99%	96.38%	96.38%	0.00%
8	96.16%	96.43%	96.43%	95.94%	96.74%	97.64%	0.93%
M	94.96%	96.00%	96.00%	**96.07%**	**96.45%**	**96.72%**	0.28%
SD	0.60%	0.87%	0.87%	0.30%	0.88%	1.09%	0.36%

Table 10. Accuracy information of SMTP

	Traditional Method			FDRS			
	TA	GA	ACS	TA	GA	ACS	PUS
1	96.38%	96.30%	96.30%	97.31%	96.83%	97.08%	0.26%
2	96.95%	96.78%	96.78%	97.13%	97.13%	97.13%	0.00%
3	96.60%	97.31%	97.31%	97.00%	96.29%	97.02%	0.75%
4	96.69%	96.08%	96.08%	97.05%	97.27%	97.40%	0.13%
5	96.69%	96.65%	96.65%	97.44%	97.13%	97.31%	0.18%
6	96.74%	96.91%	96.91%	96.96%	97.18%	97.18%	0.00%
7	96.83%	97.00%	97.00%	97.18%	96.78%	97.69%	0.93%
8	96.83%	96.96%	96.96%	97.05%	96.96%	97.00%	0.04%
M	96.71%	96.75%	96.75%	**97.14%**	**96.95%**	**97.23%**	0.29%
SD	0.17%	0.40%	0.40%	0.16%	0.32%	0.23%	0.36%

Table 11. Accuracy information of THE-OTHERS

	Traditional Method			FDRS			
	TA	GA	ACS	TA	GA	ACS	PUS
1	84.80%	83.88%	83.88%	84.76%	84.41%	84.52%	0.13%
2	84.71%	84.19%	84.19%	84.67%	84.10%	84.17%	0.09%
3	84.71%	84.14%	84.14%	84.76%	84.23%	84.23%	0.00%
4	84.67%	84.05%	84.05%	84.76%	83.97%	84.00%	0.04%
5	84.89%	84.67%	84.67%	84.67%	84.14%	84.14%	0.00%
6	84.67%	84.23%	84.23%	84.67%	83.75%	84.04%	0.35%
7	85.20%	84.85%	84.85%	84.71%	83.92%	83.92%	0.00%
8	85.20%	85.24%	85.24%	84.76%	84.27%	84.27%	0.00%
M	**84.86%**	**84.40%**	**84.40%**	84.72%	84.10%	84.16%	0.08%
SD	0.22%	0.47%	0.47%	0.04%	0.21%	0.19%	0.12%

evident from experiment results, FDRS-based classification method can optimize neural network easily.

REFERENCES

[1] Qinlan, J. R., Introduction of decision trees. *Machine Learning*, vol.1, pp.86-106. 1986.

[2] Freund,Y., Boosting a weak learning algorithm by majority, *Information Computation*, vol.121, no.2, pp.256-285, 1995.

[3] Hongjun,C., R.Setiono and L.Huan, Effect data mining using neural networks. *IEEE Transaction on knowledge and data engineering*, vol.8, no.6, pp.957-961, 1996.

[4] Boser,B. E., I. M. Guyon and V. N. Vapnik, A training algorithm for optimal margin classifiers, *Proc. 5th Annual ACM Workshop on Computational Learning Theory*, New York, NY, pp.144-152, 1992.

[5] Vapnik,V. N., *The Nature of Statistical Learning Theory*, Springer Verlag, 1995.

[6] Miniani, A. A. and R. D. Williams, Acceleration of back-propagation through learning rate and momentum adaptation, *Proc. International joint conference on neural networks* , San Diego, CA, pp.676-679, 1990.

[7] Jacobs, R. A., Increased rates of convergence through learning rate adaptation, *Neural Networks*, vol.1, no.4, pp. 295-308, 1988.

[8] Balakrishnan, K. and V. Honavar, Improving convergence of back propagation by handling flat-spots in the output layer, *Proc. Second International Conference on Artificial Neural Networks*, Brighton, U.K., pp.139-144, 1992.

[9] Hagan, M. T. and M. Menhaj, Training feedforward networks with the Marquardt algorithm, *IEEE transactions on Neural Networks*, vol.5, no.6, pp.989-993, 1994.

[10] Kim, S.-H., K. Hirasawa, J. Hu, Improvement of generalization ability for identifying dynamicsystems by using universal learning networks, *Proceedings. IJCNN '01. International Joint Conference on neural networks*, Washington, DC, pp.1203-1208, 2001.

[11] Yan,W., S. Hongbao, Improvement of neural network learning algorithm and itsapplication in control, *Proc. the 3rd World Congress on neural networks*, Hefei, Anhui, pp.971-975, 2000.

[12] Makoto, M., K. Seiichi and H. Hironori, A pulsed neural network incorporating short term synaptic plasticity for engineering applications,*international journal of innovative Computing, Information and Control*, vol.1, no.3, pp.417-428, 2005.

[13] Tadashi, K. and U. Junji, Revised gmdh-type neural network algorithm with a feedback loop identifying sigmoid function neural network , *International Journal of Innovative Computing, Information and Control*, vol.2, no.5, pp.985-996, 2006.

[14] Lina, M. and T. Fumiaki, analysis on the robustness of the pressure-based individual identification system based on neural networks, *international journal of innovative Computing, Information and Control*, vol.3, no.1, pp.97-110, 2007.

[15] Kennedy,J. and R. C. Eberhart, A new optimizer using particle swarm theory, *Proc. Sixth Int. Symposium on Micromachine and Human Science*, Nagoya, Japan, pp.39-43, 1995.

[16] KDD cup 99 Intrusion Detection Data Set, http://kdd.ics.uci.edu/database/kddcup99/kddcup.data 10 percent.gz.

[17] Yuehui, C., A. Abraham and Y. Ju, Feature selection and intrusion detection using hybrid flexible neural tree. *Proc.ISNN'2005*, Heidelberg, Berlin, pp.439-444, 2005.

[18] Lizhi, P., C. Yuehui, Y. Bo, C. Zhenxiang, A novel classification method based on data gravitation, *Proc. International Conference on Neural Networks and Brain (ICNN&B)*, Beijing, China, pp.667-672, 2005.

[19] Zander,S., T. Nguyen and G. Armitage, Automated traffic classification and application identification using machine learning. *Proc. LCN05*, Sydney, Australia, pp.15-17, 2005.