ISSN 2321-807X
Volume 12 Number10
Journal of Advances in Chemistry

AN EFFICIENT FUZZY NEURAL NETWORK TRAINING MODEL FOR SUPERVISED PATTERN CLASSIFICATION SYSTEM

Dr. R.Manjula Devi, Dr. S.J. Suji Prasad, Ms.C.Sagana
Assistant Professor(Senior Grade) / CSE, Kongu Engineering College, Perundurai
rmanjuladevi.gem@gmail.com
Assistant Professor(Senior Grade) / EIE, Kongu Engineering College, Perundurai
sjsujiprasad@gmail.com
Assistant Professor / CSE, Kongu Engineering College, Perundurai
sagana.c@gmail.com

Abstract

Among the existing NN architectures, Multilayer Feedforward Neural Network (MFNN) with single hidden layer architecture has been scrutinized thoroughly as best for solving nonlinear classification problem. The training time is consumed more for very huge training datasets in the MFNN training phase. In order to reduce the training time, a simple and fast training algorithm called Exponential Adaptive Skipping Training (EAST) Algorithm was presented that improves the training speed by significantly reducing the total number of training input samples consumed by MFNN for training at every single epoch. Although the training performance of EAST achieves faster, it still lacks in the accuracy rate due to high skipping factor. In order to improve the accuracy rate of the training algorithm, Hybrid system has been suggested in which the neural network is trained with the fuzzified data. In this paper, a z-Score Fuzzy Exponential Adaptive Skipping Training (z-FEAST) algorithm is proposed which is based on the fuzzification of EAST. The evaluation of the proposed z-FEAST algorithm is demonstrated effectively using the benchmark datasets - Iris, Waveform, Heart Disease and Breast Cancer for different learning rate. Simulation study proved that z-FEAST training algorithm improves the accuracy rate.

Keywords: *Adaptive Skipping, Neural Network, Training Algorithm, Training Speed, MFNN,Fuzzification.*

1. INTRODUCTION

Due to the implicit characteristics of approximating any nonlinear classification problem, Multilayer Feedforward Neural Network (MFNN) with a single hidden layer architecture has been scrutinized thoroughly as best for solving this problem (Mehra and Wah 1992; Hornik et al 1989). For training the above network, the Back Propagation learning algorithm has been practiced (Rumelhart and McClelland 1986; Saman and Bryan 2011). In order to enhance the training performance, the training speed is the factor that is considered to be very important. The training speed is highly depends on the dimensionality of training dataset. In general, training MFNN with a larger training datasets will generalize the network well. But, lengthy training time is needed for larger training dataset [3] which influence the training speed. In order to improve the training speed, EAST algorithm was exercised [6]. It exhibits the training input samples randomly for training which diminishes the total training input samples exponentially which in reduce the overall total training time, thereby speeding up the training process. But the accuracy rate is greatly affected. Since the Fuzzy Logic (FL) enhances the NN generalization capability and also Neuro fuzzy hybrid system are universal approximators (Kosko 1994) , a new Neuro fuzzy hybrid system with z-Score function has been put forward for improving the accuracy rate of EAST.

2. RELATED WORKS

Typically, by incorporating the advantage of both neural network and the fuzzy system, Neuro_fuzzy hybrid system is more impressive than either the neural network or the fuzzy system. Anandakumar et al [1] proposed a classification model using Modified Levenberg-Marquardt learning algorithm that improves the accuracy and also consumes less time for convergence. Initially, the statistical ANOVA ranking technique is applied to find the higher ranked dataset. In order to analyze the public transportation system service quality, an ANN model is adapted [3]. Kulkarni and Shinde [5] proposed Neuro-fuzzy classification model for supervised data classification. By using Fuzzification method, membership value is calculated for each attribute values of the given class in the membership matrix. For ANN training, this matrix is fed as an input to the model and obtains the corresponding membership value for each pattern to the target classes. At the end of each iteration, the target class for each pattern is predicted using Defuzzification method.

MFNN has been trained by Levenberg-Marquardt (LM) algorithm [9], CAST [8], EAST [6] and LAST [7] to develop a fast ANN model for nonlinear pattern classification. Patricia Melin et al [10] applied competitive neural network trained with learning vector quantization algorithm for electrocardiogram signals classification. Taskin kavzoglu et al. [13] described the way of representing the training datasets for improving the performance of classification methods. The data representation relates the training dataset size and quality. In order to identify the outlier in the training dataset, the quality analysis is used. After some refinements, representation data is formed by conducting the training data selection which is an iterative process. Quang Hung do et al [14] implemented Neuro-Fuzzy approach for solving multiclass classification problem to predict the students' academic performance.

ISSN 2321-807X
Volume 12 Number10
Journal of Advances in Chemistry

3. PROPOSED Z-FEAST METHODOLOGY

3.1 Overview of z-FEAST Architecture

The overall architecture of z-Feast classifier is represented diagrammatically in Figure 1. This Neuro-Fuzzy classifier for pattern classification consists of various blocks as specified in the figure. It consists of three steps: Fuzzification, ANN training with the backpropagation algorithm and Defuzzification[5].

Assume that the network contains *n* input nodes in the input layer, *p* hidden nodes in the hidden layer and *m* output nodes in the output layer. Since the above network is highly interconnected, the nodes in each layer are connected with all the nodes in the next layer. Let P represent the number of input patterns in the training dataset. The input matrix, X, of size p × n is presented to the network. The number of nodes in the input layer is equivalent to the number of columns in the input matrix, X. Each row in X is considered to be a real-valued vector $x_i \in \Re^{n+1}$ where $1 \le i \le n$.

In the Fuzzification process, the given training dataset is fed as input and the Z-score function is used as membership function. The membership matrix is obtained as output of this fuzzification process. The size of the matrix is S×D×C, where S is the number of input samples in the training dataset, D is the number of features / attributes and C is the number of target class. Then, this membership matrix, that is fuzzified data, is fed as input to the MFNN. The summed real-valued vector generated from the hidden layer is represented $z_i \in \Re^{p+1}$ where $1 \le i \le p$. The estimated output real-valued vector generated from the network is denoted as $y_i \in \Re^m$ where $1 \le i \le m$ and the corresponding target vector is represented as $t_i \in \Re^m$ where $1 \le i \le m$. Let *it* signifies the it^{th} iteration number.

Then, the network generated output, y_i, is given as input to the defuzzification process. The MAX defuzzification method is applied for this process by assigning the pattern to the highest membership class. The defuzzified vector is compared with the target vector for calculating the error rate.

Let $f_N(x)$ be the activation function used in the hidden layer and $f_L(x)$ be the activation function used in the output layer. Let v_{ij} be the $n \times p$ weight matrix contains input-to-hidden weight coefficient for the link from the input node *i* to the hidden node *j* and v_{oj} be the bias weight to the hidden node *j*. Let w_{jk} be the $p \times m$ weight matrix contains hidden-to-output weight coefficient for the link from the hidden node *j* to the output node *k* and w_{ok} be the bias weight to the output node *k*.

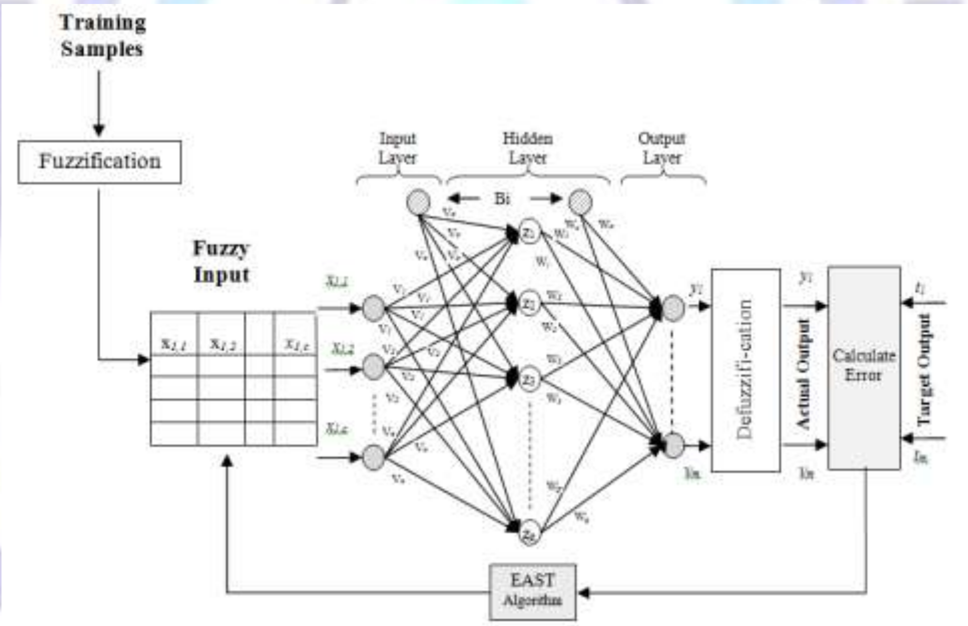

Figure 1. Architecture of z-FEAST classifier

3.2 Proposed z-FEAST Algorithm

The working principle of the z-FEAST algorithm that is incorporated in the BPN algorithm is summarized below:

Step 1. Weight Initialization: Initialize weights to small random values;

Step 2. Furnish the input sample: Disseminate to the input layer an input sample vector $\mathbf{x}_k$ having desired output vector $\mathbf{y}_k$;

Step 3. Fuzzification Process:

Convert Crisp to fuzzy value for the input vector $\mathbf{x}_k$. Input vectors are fuzzified using Z-score method. Z-score is modeled mathematically as

$$Z(x;c;\sigma) = \begin{Bmatrix} e^{\left(\frac{-(x-c)}{\sigma}\right)}, \frac{x-c}{\sigma} > 0 \\ e^{\left(\frac{(x-c)}{\sigma}\right)}, \frac{x-c}{\sigma} < 0 \end{Bmatrix} \quad (1)$$

Where

- x – Feature value
- C – MF centre
- σ is the MFs width

The Membership Function's centre, *C,* is given by,

$$C_d = \frac{1}{S}\sum_{j=1}^{S} x_{jd} \tag{2}$$

Where

- d=1, 2… D,
- j=1, 2… S and
- x_{jd} is the d[th] feature of sample j.

The Membership Function's width, σ is given by,

$$\sigma_d = \frac{1}{S-1}\left(\left|x_{1d} - c_d\right| + \left|x_{2d} - c_d\right| \ldots\ldots \left|x_{sd} - c_d\right|\right) \tag{3}$$

Where

- $x_{1d}, x_{2d}, \ldots\ldots, x_{sd}$ are the d[th] feature of the s[th] pattern and
- C_d denote the mean value of d[th] feature given in Equation (2)

The membership matrix, *fx,* that is generated using the equation (1). In this matrix, $g_{s,c}(d)$ represent the membership value of d[th] feature of s[th] pattern to the c[th] class. *fx* is given as

$$fx = \begin{pmatrix} z_{1,1}(x_1) & z_{1,2}(x_1) & \ldots & z_{1,C}(x_1)\ldots & z_{1,1}(x_D) & z_{1,2}(x_D) & \ldots z_{1,C}(x_D) \\ z_{2,1}(x_1) & z_{2,2}(x_1) & \ldots & z_{2,C}(x_1)\ldots & z_{2,1}(x_D) & z_{2,2}(x_D) & \ldots z_{2,C}(x_D) \\ . & . & . & . \quad . & . & . & . \quad . \\ . & . & . & . \quad . & . & . & . \quad . \\ z_{S,1}(x_1) & z_{S,2}(x_1) & \ldots & z_{S,C}(x_1)\ldots & z_{S,1}(x_D) & z_{S,2}(x_D) & \ldots z_{2,C}(x_D) \end{pmatrix}$$

Step 4. Forward Phase: Starting from the first hidden layer and propagating towards the output layer:

a. **Calculate the activation values** for the Hidden layer as:

i. Estimate the net output value

$$z_{inj}(it) = v_{oj}(it) + \sum_{i=1}^{n} \mathrm{fx_i(it).\,v_{ij}(it)} \tag{3}$$

ii. Estimate the actual output

$$z_j(it) = \frac{1}{1 + e^{-z_{inj}}} \tag{4}$$

b. **Calculate the activation values** for the Output layer as:

i. Estimate the net output value

$$y_{ink}(it) = w_{ok}(it) + \sum_{j=1}^{p} z_j(it).\,w_{jk}(it) \tag{5}$$

ii. Estimate the actual output

$$y_k(it) = \frac{1}{1 + e^{-y_{ink}}} \tag{6}$$

Step 5. Output errors: Calculate the error terms at the output layer as:

$$\delta_k(it) = [t_k - y_k(it)].\,f'(y_k(it)) \tag{7}$$

Differentiate the activation function in Equation 6,

$$f'(y_k(it)) = \frac{\partial(y_k(it))}{\partial x} = y_k(it) \times (1 - y_k(it)) \tag{8}$$

Substitute the resultant value of Equation (8) in (7)

$$\delta_k(it) = y_k(it).\,[1 - y_k(it)].\,[t_k - y_k(it)] \tag{9}$$

Step 6. Backward Phase: Propagate error backward to the input layer through the hidden layer using the error term

$$\delta_j(it) = \left[\sum_{k=1}^{m} \delta_j(it).w_{jk}(it)\right].f'\left(z_j(it)\right) \qquad (10)$$

Differentiate the activation function in Equation 4,

$$f'\left(z_j(it)\right) = \frac{\partial\left(z_j(it)\right)}{\partial x} = z_j(it) \times \left(1 - z_j(it)\right) \qquad (11)$$

Substitute the resultant value of Equation (11) in (10)

$$\delta_j(it) = \left[\sum_{k=1}^{m} \delta_j(it).w_{jk}(it)\right] z_j(it).\left[1 - z_j(it)\right] \qquad (12)$$

Step 7. Weight Amendment: Update weights using the Delta-Learning Rule

a. **Weight amendment** for Output Unit

$$W_{jk}(it+1) = W_{jk}(it) + \alpha(it).\delta_k(it).z_j(it) \quad (13)$$

b. **Weight amendment** for Hidden Unit

$$V_{ij}(it+1) = V_{ij}(it) + \alpha(it)\,\delta_j(it)\,x_i(it) \qquad (14)$$

Step 8. EAST Algorithm: Incorporating the EAST algorithm

a. **Compare** the error value, $|t_k - y_k|$ with threshold value, d_{max}

$$|t_k - y_k(it)| < d_{max} \qquad (15)$$

If equation 15 generates 0, then the x_i is correct

b. **Compute** the probability value for all input samples

$$prob(x^i) = \begin{cases} 0, \text{if } x_i \text{ is correct and epoch number} < n \\ 1, otherwise \end{cases} \quad (16)$$

c. **Calculate** the skipping factor, sf_i, for all input samples

i. Initialize the value of sf_i to zero (for first epoch)

ii. Increment the value of sf_i exponentially for correctly classified samples alone.

d. **Skip** the training samples with prob (=0) for the next sf_i epoch

Step 9. Defuzzification Process:

The Fuzzy to Crisp conversion for the output variable is done using centroid method

$$x^* = \frac{\int \mu_{\tilde{C}}(x).x dx}{\int \mu_{\tilde{C}}(x) dx}$$

To assign the class label,

$$A = \begin{bmatrix} O_{1,1} & O_{1,2} & .... & O_{1,C} \\ O_{2,1} & O_{2,2} & .... & O_{2,C} \\ . & . & . & . \\ . & . & . & . \\ O_{S,1} & O_{S,2} & .... & O_{S,C} \end{bmatrix}$$

Where $O_{s,c}$ is the output of s^{th} pattern to the c^{th} class

Step 10. Repeat steps 1-7 until the halting criterion is satisfied, which may be chosen as the Root Mean Square Error (RMSE), elapsed epochs and desired accuracy

Working Flow of z-FEAST Algorithm

The block diagram of the proposed strategy is diagrammatically represented in the following figure

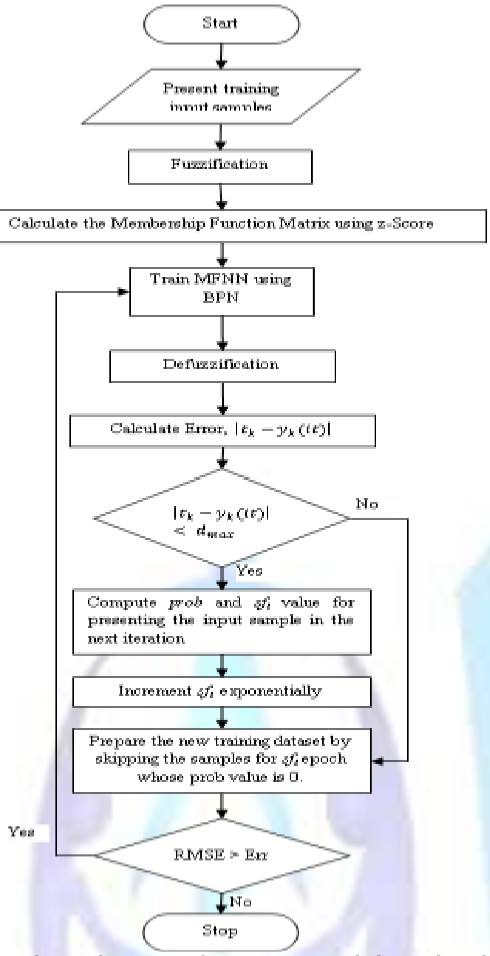

Figure 2: Flow Diagram of z-FEAST Training Algorithm

4. EXPERIMENTAL SETUP AND RESULT

4.1 Experimental Layout

A 3-layer feedforward neural network is adopted for the simulation of all the training algorithms with the selected training architecture and training parameters mentioned in the Table 1. The simulation of all the training algorithms is repeated for two different learning rates such as 1e-4 (0.0001) and 1e-3(0.001).

Table 1 Selected Training Architectures and Parameters

Datasets	Learning Rate	MFNN Architecure	Momentum
Heart	1e – 4	13×5×1	0.9
	1e – 3		
Breast Cancer	1e – 4	31×15×1	0.9
	1e – 3		
Iris	1e - 4	4 × 5 × 1	0.8
	1e – 3		
Waveform	1e – 4	21×10×1	0.7
	1e – 3		

For training Heart dataset, 13, 5 and 1 neurons in the input, hidden and output layers respectively is used. And, for training Breast Cancer dataset, the NN architecture with 31, 15 and 1 neurons in the input, hidden and output layers respectively, is used. The NN architecture with 4, 5 and 1 neurons in the input, hidden and output layers respectively, is used for training Iris database. For training waveform dataset, 21, 10 and 1 neurons in the input, hidden and output layers respectively, is used.

According to the idea of Nguyen-Widrow algorithm (Nguyen and Widrow 1990), the NN weight coefficients are initialized with the random values within the specified range -0.5 to +0.5 for faster learning.

4.2 Evaluation Method

The Fivefold cross validation method (Witten and Frank 2000) is performed on the above datasets to evaluate and compare the performance of the proposed training algorithms empirically on unseen data. The input sample in each dataset is randomly split into five equal sized disjoint folds. Among these five folds, a single fold is retained as the validation data for testing the network, and the remaining four folds are used as training data for training the network. The validation process is then repeated for five times, with each of the five folds used exactly once as the validation data. The results taken from the five training folds can then be averaged to produce a final result. The advantage of using this validation method is that all observations are used for both training and validation, and each observation is used for validation exactly once and also to avoid over-fitting (Peterson et al 1995).

The performance measures that are considered to evaluate the training algorithm are training time and classification accuracy. A good training algorithm will cut down the training time, while accomplishing better accuracy which is proved in our proposed work. The classification accuracy is calculated using the following formula

$$\text{Classification accuracy} = \frac{\text{Number of samples correctly classified}}{\text{Total number of samples}}$$

The simulations of all the proposed training algorithms are done using MATLAB R2010b on a machine with the configuration of Intel® Core I5-3210M processor, 4 GB of RAM and CPU speed of 2.50GHz.

4.3 Dataset Description

The performance of all proposed AST algorithm is assessed for the classification problem on the benchmark two-class classification and multi-class classification datasets. The real-world benchmark datasets utilized for two-class classification problem are Heart and Breast Cancer Dataset, and multiclass classification problem are Iris and Waveform Dataset. The fore-mentioned datasets were fetched from the UCIMLR (University of California at Irvine Machine Learning Repository) (Asuncion and Newman 2007).

The specification of the benchmark datasets utilized for training in the research is summarized in Table 2.

Table 2. Datasets Description used in the Research

Datasets	No. of Attributes	No. of Classes	No. of Instances
Iris	4	3	150
Waveform	21	3	5000
Heart	13	2	270
Breast Cancer	31	2	569

4.3.1 Multiclass Problems

Iris Dataset

In the IRIS dataset, the number of iris flower samples is 150 which is gathered from three different flower varieties equally. The varieties are listed as Iris Setosa, Iris Versicolour and Iris Virginica which is identified using width and length of Iris sepal, and width and length of Iris petal. Among these varieties, Iris Setosa is easier to be separated from the other two varieties, while the other two varieties, Iris Virgincia and Iris Versicolour, are partially obscured and harder to be distinguished.

Waveform Dataset

In the Waveform database generator dataset, the total number of wave’s samples is 5000 with 21 attributes which are equally divided into three wave classes (Asuncion and Newman 2007). These samples are collected from the generation of 2 of 3 “base” waves.

4.3.2 Two-Class Problems

Heart Dataset

In the Statlog Heart disease database, the samples with 13 attributes are collected from 270 patients. Among these samples, the number of samples with heart disease ‘absent’ is 150 and with heart disease ‘present’ is 120.

Breast Cancer Dataset

In the Wisconsin Breast Cancer Diagnosis Dataset, the samples are collected from the patient’s characteristics of 569 with 32 features. Among these samples, 357 samples are diagnosed as benign and 212 samples are diagnosed as malignant class.

4.4 EXPERIMENTAL RESULT

Table 3 to 10 shows the experimental results of EAST, FEAST and z-FEAST algorithms observed at each step across five repeats of fivefold cross validation using two different learning rates such as 1e-4 and 1e-3. From these table 3 to 10, the EAST algorithm yields improved computational training speed in terms of the total number of trained input samples as well as total training time over FEAST and z-FEAST. But, when the skipping factor goes higher, the accuracy of EAST system is affected highly. But, z-FEAST improves the accuracy rate of the system.

Table 3. Comparison Results Trained by the Iris Dataset with 1e-4 Learning Rate

Testing Fold	Number of Epochs	EAST			FEAST			z -FEAST		
		Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)
1	5442	208755	8.2995	73.33	227585	11.345	88.8	217585	9.4567	90.4
2	5902	240293	8.5218	76.67	270392	12.0234	87.03	253009	9.1345	88.8
3	5332	206029	8.296	80	256029	11.9865	92.5	225920	9.4769	94
4	5439	223245	8.2565	80	267254	11.2345	92.5	232453	9.3426	94
5	5161	203116	7.8261	76.67	241613	10.1237	90.4	211361	8.4567	92.5
AVG	**5455**	**216288**	**8.24**	**77.33**	**252575**	**11.34**	**90.25**	**228066**	**9.17**	**91.94**

Table 4. Comparison Results Trained by the IRIS Dataset with 1e-3 Learning Rate

Testing Fold	Number of Epochs	EAST			FEAST			z -FEAST		
		Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)
1	547	22339	0.7867	76.67	25339	0.998	88.8	23239	0.8476	90.4
2	526	21369	0.7537	80	24369	0.9876	87.03	22096	0.8123	88.8
3	535	21735	0.7667	76.67	24735	0.9964	92.5	22145	0.8123	94
4	545	22120	0.7756	80	26120	1.0276	92.5	23102	0.8597	94
5	510	20735	0.7306	76.67	23735	0.9306	90.4	21342	0.8176	92.5
AVG	**533**	**21660**	**0.76**	**78**	**24860**	**0.99**	**90.25**	**22385**	**0.83**	**91.94**

Table 5. Comparison Results Trained by the Waveform Dataset with 1e-4 Learning Rate

Testing Fold	Number of Epochs	EAST			FEAST			z -FEAST		
		Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)
1	8187	16974989	17.28264	79.8	28191603	28.8946	92.50	18048595	20.9867	94.4
2	8973	17897431	30.3537	80.2	30488836	39.9352	94.50	19081160	33.2345	96.29
3	8929	17812293	30.22541	81.1	30142334	38.8432	94.50	19008000	32.7654	96.29
4	8903	17806977	29.0942	80.9	30095407	36.2356	92.50	19059754	31.4357	94.4
5	8887	17144339	28.6921	79.9	30024085	39.0178	92.50	19557704	30.3569	94.4
AVG	**8776**	**17527206**	**27.13**	**80.38**	**29788453**	**36.59**	**93.30**	**18951043**	**29.76**	**95.16**

Table 6. Comparison Results Trained by the Waveform Dataset with 1e-3 Learning Rate

Testing Fold	Number of Epochs	EAST			FEAST			z -FEAST		
		Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)
1	823	1611594	2.6747	81.1	2795278	5.2589	92.50	1873866	3.5973	94.4
2	894	1785336	2.9381	80.6	3164808	6.1254	94.50	1924189	3.234	96.29
3	891	1761213	2.8975	79.9	3159738	6.0124	94.50	1903561	3.6974	96.29
4	890	1784880	2.8904	80.5	3155227	6.1123	92.50	1931837	3.1520	94.4
5	890	1659327	2.8696	80.1	3151527	6.4532	92.50	1921729	3.5678	94.4
AVG	**878**	**1720470**	**2.85**	**80.44**	**3085316**	**5.99**	**93.30**	**1911036**	**3.45**	**95.16**

Table 7. Comparison Results Trained by the Heart Dataset with 1e-4 Learning Rate

Testing Fold	Number of Epochs	EAST			FEAST			z -FEAST		
		Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)
1	7485	713559	23.2651	75.93	1374624	36.2	92.50	902743	24.56	94.40
2	7529	809372	25.3458	74.07	1412889	36.44	92.50	939929	27.81	94.40
3	7569	820114	27.84309	75.93	1427326	34.6	88.80	966050	29.23	90.70
4	7567	813699	26.6308	79.63	1426992	38.7	96.29	965139	29.33	98.10
5	7567	811180	25.9578	77.78	1425689	35.49	92.59	964781	28.01	94.40
AVG	**7543**	**793585**	**25.81**	**76.67**	**1413504**	**36.29**	**92.54**	**947729**	**27.79**	**94.40**

Table 8. Comparison Results Trained by the Heart Dataset with 1e-3 Learning Rate

Testing Fold	Number of Epochs	EAST			FEAST			z -FEAST		
		Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)
1	830	95137	3.3133	74.07	116885	5.14	92.50	99031	3.29	94.40
2	828	98116	3.382314	75.93	116828	4.1	92.50	103076	2.86	94.40
3	829	90205	3.533761	75.93	116809	4.8	88.80	95020	3.12	90.70
4	829	93136	3.554815	74.07	116808	4.19	96.29	97929	3.13	98.10

5	829	99092	3.993784	77.78	116799	4.08	92.59	102895	3.49	94.40
AVG	**829**	**95137**	**3.56**	**75.56**	**116826**	**4.46**	**92.54**	**99590**	**3.18**	**94.40**

Table 9. Comparison Results Trained by the Breast Cancer Dataset with 1e-4 Learning Rate

Testing Fold	Number of Epochs	EAST			FEAST			z -FEAST		
		Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accurac y (%)	Total Number of Input Samples	Trainin g Time (in Sec)	Accuracy (%)
1	6279	1055844	34.08077	83.33	1840880	40.9876	94.40	1230721	36.6754	96.9
2	6460	966328	30.7942	79.82	1932488	36.9845	92.50	1008750	32.156	94.4
3	7976	1286262	46.8745	84.21	2283816	52.2546	94.40	1301532	48.3458	96.9
4	7691	1138979	43.9744	80.07	2195447	50.0123	88.80	1260383	46.4567 6	90.25
5	7439	1097278	31.3622	84.07	2108865	37.1249	90.40	1214473	33.5678	92.5
AVG	**7169**	**1108938**	**37.42**	**82.3**	**2072299**	**43.47**	**92.10**	**1203172**	**39.44**	**94.19**

Table 10. Comparison Results Trained by the Breast Cancer Dataset with 1e-3 Learning Rate

Testing Fold	Number of Epochs	EAST			FEAST			z -FEAST		
		Total Number of Input Samples	Training Time (in Sec)	Accuracy (%)	Total Number of Input Samples	Training Time (in Sec)	Accurac y (%)	Total Number of Input Samples	Trainin g Time (in Sec)	Accuracy (%)
1	609	101916	5.4285	83.33	179370	8.243	94.40	117756	6.2147	96.9
2	647	107089	5.895	84.21	186480	8.9649	92.50	113460	6.6893	94.4
3	785	132372	6.4982	84.21	226882	11.0626	94.40	143034	7.2648	96.9
4	750	128676	5.895	83.33	217547	9.0528	88.80	131007	6.3914	90.25
5	743	120608	5.7421	84.07	213966	11.9146	90.40	138690	6.7473	92.5
AVG	**707**	**118132**	**5.89**	**83.83**	**204849**	**9.85**	**92.10**	**128790**	**6.6615**	**94.19**

4.5 RESULT ANALYSIS

4.5.1 Training Samples Comparison

The comparison results of the total number of input samples consumed for training by EAST, FEAST and z-FEAST with the learning rate of 1e-4 and 1e-3 are shown in Fig.3-6.

Herewith, it is assured from the Figure 3 that the total number of training samples consumed by EAST algorithm for training under the learning rate of 1e-4 is reduced by an average of nearly 17% and 6% of FEAST and z-FEAST algorithm for Iris Dataset, 69% and 8% for Waveform Dataset, 78% and 9% for Heart Dataset and 87% and 8% for Breast Cancer Dataset respectively.

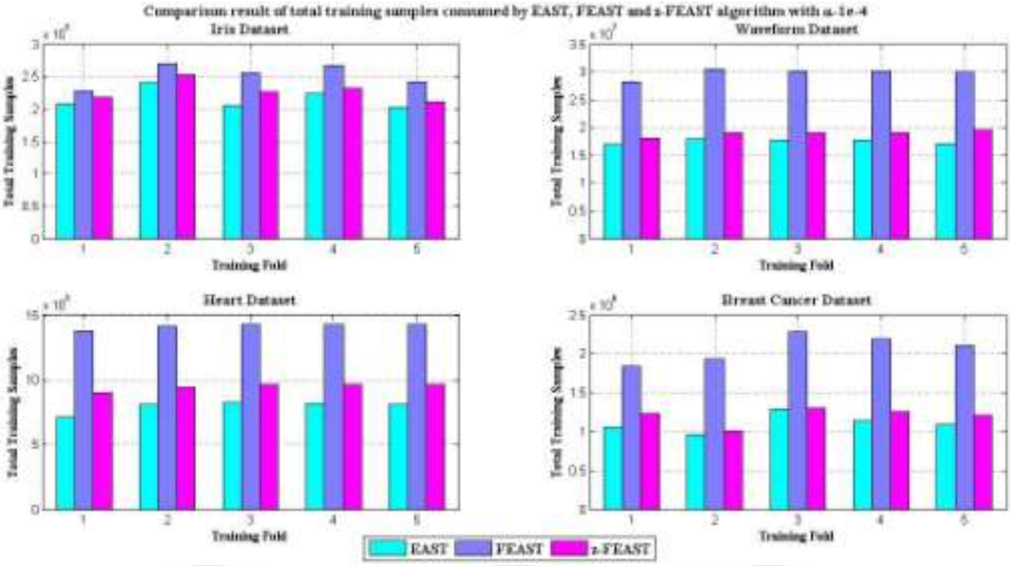

Figure 3: Comparison Result of total training samples consumed with 1e-4 learning rate

Herewith, it is assured from the Figure 4 that the total number of training samples consumed by EAST algorithm for training under the learning rate of 1e-3 is reduced by an average of nearly 15% and 3% of FEAST and z-FEAST algorithm for Iris Dataset, 79% and 11% for Waveform Dataset, 23% and 5% for Heart Dataset and 73% and 9% for Breast Cancer Dataset respectively.

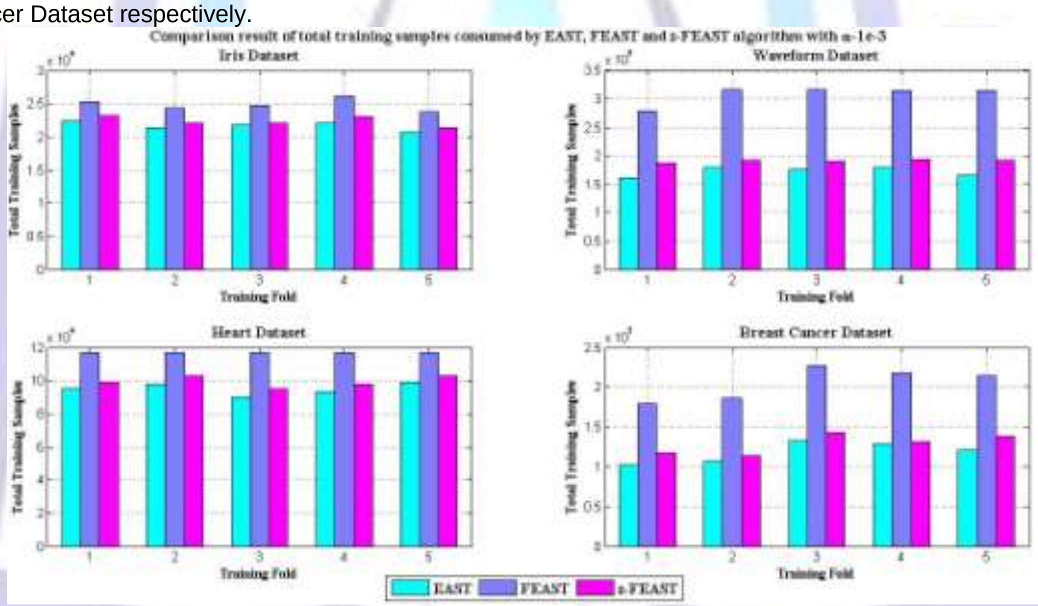

Figure 4: Comparison Result of total training samples consumed with 1e-3 learning rate

4.5.2 Training Time Comparison

Herewith, it is concluded from the Figure 5, for training IRIS dataset, the total training time consumed by EAST algorithm with the learning rate of 1e-4 is reduced to an average of 37% of FEAST algorithm and 11% of z-FEAST algorithm, for Waveform Dataset by 35% of FEAST algorithm and 10% of z-FEAST algorithm, for Heart Dataset by 41% of FEAST algorithm and 9% of z-FEAST algorithm and for Breast Cancer Dataset by 16% of FEAST algorithm and 6% of z-FEAST algorithm respectively.

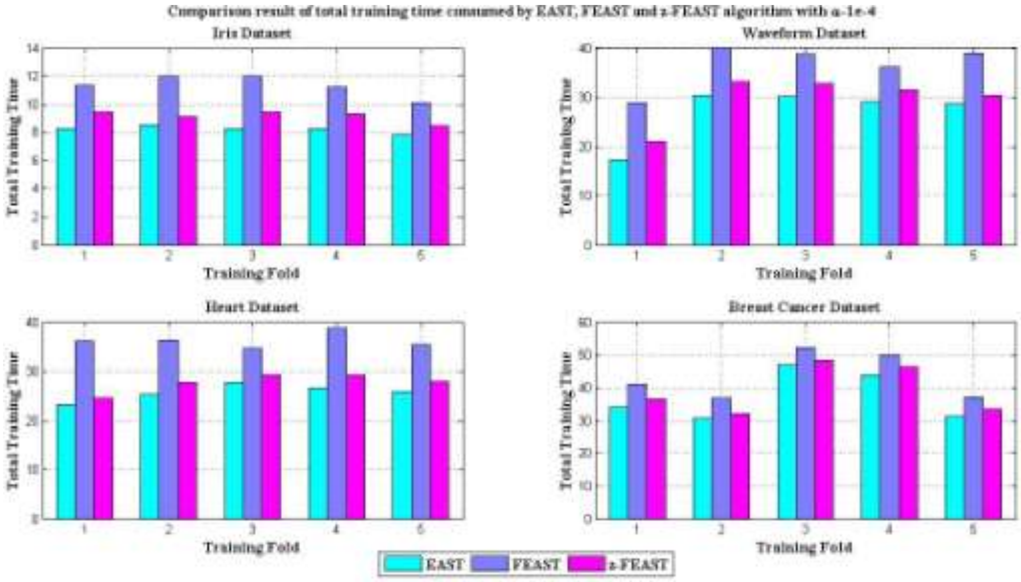

Figure 5: Comparison Result of total training time consumed with 1e-4 learning rate

Herewith, it is concluded from the Figure 6, for training IRIS dataset, the total training time consumed by EAST algorithm with the learning rate of 1e-3 is reduced to an average of 29% of FEAST algorithm and 8% of z-FEAST algorithm, for Waveform Dataset by 45% of FEAST algorithm and 10% of z-FEAST algorithm, for Heart Dataset by 25% of FEAST algorithm and 10% of z-FEAST algorithm and for Breast Cancer Dataset by 67% of FEAST algorithm and 13% of z-FEAST algorithm respectively.

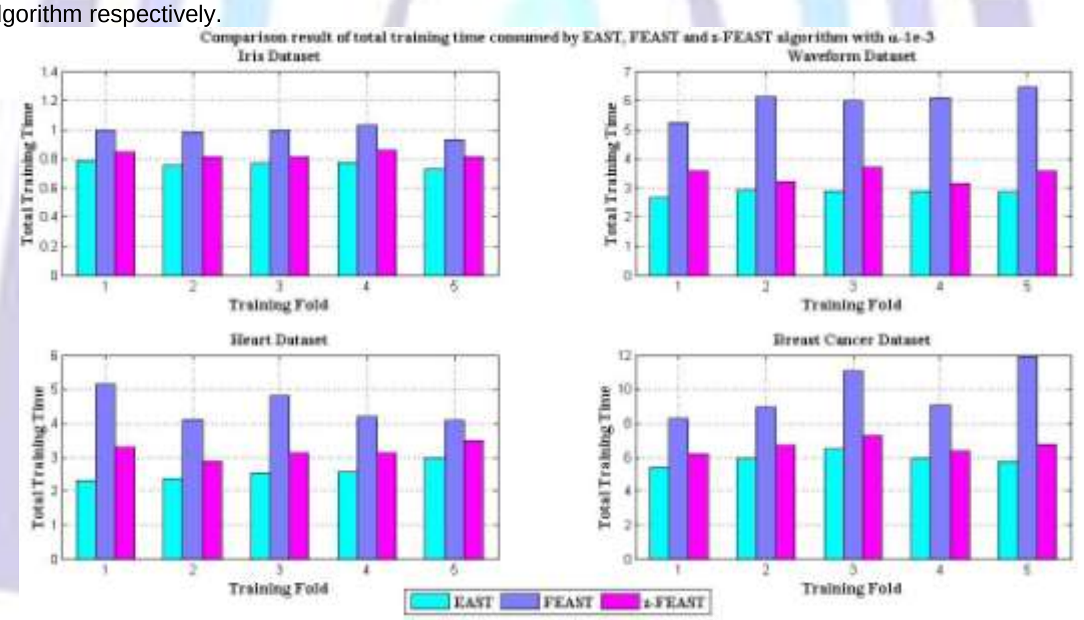

Figure 6: Comparison Result of total training time consumed with 1e-3 learning rate

5. CONCLUSION

Thus, the z-Score Fuzzy Exponential Adaptive Skipping Training (Z-Feast) Algorithm is systematically investigated in order to improve the accuracy rate of EAST algorithm. And also, It is further concluded that the proposed z-FEAST algorithm is much faster than the standard BPN, LAST, CAST, HOT and EAST algorithm and also the accuracy rate is highly improved compared to EAST algorithm. The proposed z-FEAST Algorithm can be incorporated in any supervised algorithm used for training real-world supervised pattern classification.

REFERENCES

[1] Anandakumar.K., and Punithavalli.M. (2012), "Efficient cancer classification using Fast Adaptive Neuro-Fuzzy inference System based on statistical technique," *Advanced computer science and application,* Vol.13, No.3 pp. 132–137.

[2] Ashish Gosh.,Uma Shankar.B.,. and Saroj K.Meher. (2009), "*Neural network*", Vol. 22, No.3, pp.100-109.

[3] Concepcion Garrido, Rocio de ona., and Juan de ona. (2014), "Neural networks for analyzing service quality in public transportation," *Expert System with Applications*, vol.41, no.17, pp.6830-6838.

[4] Hamed Malek, Mohammed Mehdi Ebadzadeh and Mohammed Rahmati. (2011), "Three new fuzzy neural networks learning algorithm based on clustering,training error and genetic algorithm,"*Springer*,vol.23,no.3,pp.534-542

[5] Kulkarni U.V. and Shinde .S.V. (2013), "Neuro fuzzy classifier based on the Gaussian membership function," *ICCCNT*, July4-6.

[6] Manjula Devi, R., Kuppuswami S., and Suganthe R.C. (2014), "EAST: An Exponential Adaptive Skipping Training Algorithm for Multilayer feedforward neural network," *WSEAS Transaction on computers*, vol. 13, E-ISSN: 2224-2872.

[7] Manjula Devi, R. Kuppuswami, S. and Suganthe, R. C. (2013), "Fast Linear Adaptive Skipping Training Algorithm for Training Artificial Neural Network," Mathematical Problems in Engineering, vol. 2013, Article ID 346949, 9 pages, 2013. doi:10.1155/2013/346949

[8] Manjula Devi R. and Kuppuswami S. (2016), "CAST: A constant Adaptive Skipping Training Algorithm for Improving the Learning Rate of Multilayer Feedforward Neural Networks," Research Journal of Applied Sciences, Engineering and Technology, vol.12, no.(8), pp:790-812.

[9] Paras kumar., Nigam S.P. and Narotam kumar (2014), "Vehicular traffic noise modelling using artificial neural network," *Transport research*, vol.40, no.5, pp.191-203.

[10] Patricia Melin, Jonathan Amezcua, Fevrier valdez and Oscar castillo (2014), "A new neural network model based on the LVQ algorithm for multi-class classification of arrhythmias," *Information Sciences*, vol.279, pp.483-497.

[11] Ramsingh.R. (2014), "Analysis of line to ground fault in transformer by Elman's Network using gradient descent Backpropagation Algorithm," *Journal of Information Science, Engineering and Technology*, Vol.14, No.2, pp.691-700.

[12] Ruperz M.J, Martin Guerrero, Monserrat and Alcaniz .M. (2012), "Artificial Neural Network for predicting dorsal pressure on the foot surface while walking," *Expert system with applications*, vol.39, no.2, pp.5349-5357.

[13] Taskin Kavzoglu (2009), "Increasing the accuracy of neural network classification using refined data," *Environmental modelling and softwares*, vol.24, pp.850-858.

[14] Qang Hung do, and Jen Fung-chen. (2013), "Neuro Fuzzy approach in the classification of student's Academic performance," *Computational Intelligence and Neurosciences*, Vol. 31, pp.120-129.

[15] Kosko,B.(1994), "Fuzzy Systems as universal approximators," IEEE Transactions on Computer, vol.43, no.11,pp.1329-1333.