Genetic Algorithm based Rule Extraction from Pruned Modified Fuzzy Hyperline Segment Neural Network for Pattern Classification

S. B. Bagal
Associate Professor,
Dept. of Electronics & Telecomm. Engineering
KCT's LGNSCOE, Nashik, (M. S.), India

U. V. Kulkarni
Professor and Head,
Dept. of Computer Science and Engineering
SGGSIOET, Nanded, (M. S.), India.

ABSTRACT
The Pruned modified fuzzy hyperline segment neural network (PMFHLSNN) is pruned extension of Fuzzy hyperline segment neural network (FHLSNN) with modification in the testing phase. In this paper, a genetic algorithm based rule extractor (GA-PMFHLSNN) is proposed to extract a small set of compact and comprehensible fuzzy if-then rules with high classification accuracy from the PMFHLSNN. After pruning, open hyperline segments are generated from the remaining hyperline segments and a "don't care" approach is adopted by GA rule extractor to minimize the number of features in the extracted rules with higher classification accuracy. The performance of FHLSNN, PMFHLSNN and GA-PMFHLSNN are evaluated using tenfold cross-validation for five benchmark problems and handwritten character database. All the results show that the proposed approach can extract a set of compact and comprehensible rules with high classification accuracy for all the selected datasets.

General Terms
Pruned modified fuzzy hyperline segment neural network, Rule extraction, Pattern Classification.

Keywords
PMFHLSNN, genetic algorithm, confidence factor, fuzzy if-then rules extraction, don't care antecedent, pruning, tenfold cross validation, pattern classification.

1. INTRODUCTION
The research over the past few decades in the field of pattern classification and recognition has shown that, of all the available pattern classification technologies have its own strengths or weakness. Therefore, many times it is beneficial to use number of pattern classification techniques collectively rather than exclusively. This results in the construction of complementary hybrid intelligent systems. Such classification systems are expected to "possess humanlike expertise within a specific domain, adaptions and learn to do better in changing environments, and explain how they make decisions or take actions". Thus, presently hybrid systems which combine the strengths of three components of computational intelligence, that include artificial neural network (ANN), fuzzy logic (FL) and/or genetic algorithm (GA), are popular for pattern recognition and classification [1].

Many hybrid Fuzzy neural network (FNN) systems which combine the strength of ANN and FL are reported for the applications of pattern recognition and classification. Simpsonproposed supervised fuzzy min-max neural network (FMN) [2] and unsupervised fuzzy min-max clustering neural network [3]. Kwan and Caihave proposed "unsupervised four layer feedforward fuzzy neural network" for character recognition [4]. Gabrys and Bargielahave proposed "general fuzzy min-max neural network" (GFMM), which is a generalization and extension of the fuzzy min-max clustering and classification [5]. In the same sequel, Kulkarni U. V. et al. have proposed "fuzzy hyperline segment neural network" (FHLSNN) for recognition of rotation invariant handwritten character [6] and unsupervised "fuzzy hyperline segment clustering neural network" [7]. Patilet al. have proposed "general fuzzy hyperline segment neural network" (GFHLSNN), which is a generalization and extension of fuzzy hyperline segment classification and clustering [8].

In the later phase, the hybrid approach of combining neural networks and FNN with other soft computing tools has been spotlighted as a new path in the design and development of better performance classification systems. Wang and Fan have mathematically and concisely model pattern recognition as the problem of optimizing a composite function. They surveyed the status of applying GA on pattern recognition with advantages and disadvantages of different types of approaches [9]. Chopresents a hybrid approach in which neural network and GA are combined for recognition problem of totally unconstrained handwritten numerals [10]. Oh et al. have proposed an advanced architecture of "genetically optimized hybrid fuzzy neural network" (gHFNN) to model a three-input nonlinear function, time series of gas furnace and NOx emission process [11].

Although the synergic use of ANN, FL, and/or GA can improve the performance of systems, it is often said that neural networks are practically 'black boxes' due to their complexity. Therefore, the problem of determining the proper size of neural network for solving a particular task is crucial and fundamental issue in the neural network applications [12]. Basically, there are two methods to find suitable required size of a neural network. In the first method, designer has to begin with small network and slowly go on adding the connections to it, until required stopping condition is achieved [13]. In the second method, designer has to begin with a network structure that is knowingly too large for the specific problem and then trim it down to the suitable size. This is known as "neural network pruning" [14].

In the several applications of FNN, it is expected to extract information or knowledge from learned neural networks for the users to gain a better understanding of how the networks solve the assigned tasks. These requirements are especially very useful in decision support applications such as airlines, power stations and medical applications. The "black box" nature of ANN can be converted into a "white box" by

translating the internal contents (information/knowledge) of it into a set of "comprehensible and meaningful rules" [15-16].

Now days various rule extraction methods are available. Andrews et al.overview various rule extraction techniques and focuses on "mechanism, procedures and algorithms designed to insert knowledge into ANN, extract rules from trained ANN and utilize ANN to refine existing rule bases" [17].Anas et al. proposed a "two stage pattern classification and rule extraction system for modified fuzzy min-max neural network"[18]. Yap et al. proposed an "improved generalized adaptive resonance theory" (IGART) model to extract the if-then rules with improved classification performance for the power systems [19]. Chen et al. proposed a classification system with the objective to select useful features along with rule extraction using tenfold cross-validation [20]. Yang et al. presented a novel GA based method especially useful in the mining multilevel association rules in big data related applications [21].

Bagal and Kulkarni proposed the modifications in the testing phase of FHLSNN to improve its classification performance [22].Theyalso proposed the confidence factor (CF) based pruning of this modified FHLSNN. The resulting classifier is Pruned modified fuzzy hyperline segment neural network (PMFHLSNN) [23].This paper describes a GA based rule extractor to extract a small set of compact and comprehensible fuzzy if-then rules from the PMFHLSNN. After pruning, open hyperline segments (HLS) are generated from the remaining HLS and a "don't care" approach is used by GA rule extractor to minimize the number of features in the extracted rules with higher classification accuracy. The performance of FHLSNN, PMFHLSNN and GA-PMFHLSNN is evaluated using tenfold cross-validation for five selected benchmark problems and handwritten character database. The results are evaluated, discussed and compared with the FHLSNN and other state of art classifiers.

Rest of the paper is organized as follows. Section 2 explains the FHLSNN and PMFHLSNN in brief. Section 3 describes GA-based rule extractor. The experimental procedure, simulation result, description of data sets and discussions on the results are explained in Section 4. Finally, conclusions are presented in Section 5.

2. PRUNED MODIFIED FHLSNN

2.1 FHLSNN

The Figure 1 shows the four layer architecture of FHLSNN. As shown in the figure, first, second, third and fourth layer of the architecture are represented as F_R, F_E, F_D,and F_C respectively. The input pattern is applied to the F_R layer and it consists of n processing elements, one for each dimension of the pattern. The F_E layer consists of m processing nodes that are constructed during training. There are two connections from each F_R to each F_E node. Each connection represents an end point for that particular HLS. These end points are stored in the two matrices V and W. The third F_D layer gives soft decision and last F_C node delivers nonfuzzy output. Each F_E node represents HLS fuzzy set and is characterized by the membership function. The membership function of the k^{th} F_E node is given by

$$e_k(R_h, V_k, W_k) = 1 - f(x, \gamma, l) \quad (1)$$

where, $R_h = (r_{h1}, r_{h2,} \dots\dots\dots\dots, r_{hn})$ is represents the h^{th}input pattern, $V_k = (v_{k1}, v_{k2}, \dots\dots\dots v_{kn)}$ is the one end point of HLS e_k, $W_k = (w_{k1}, w_{k2,} \dots\dots\dots\dots, w_{kn})$ is the other end point of HLS e_k, and

$$x = l_1 + l_2 \quad (2)$$

where, l_1 is the distance of R_h from end point W_k, l_2 is the distance of distances R_h from end point V_k and l is the length of the HLS.

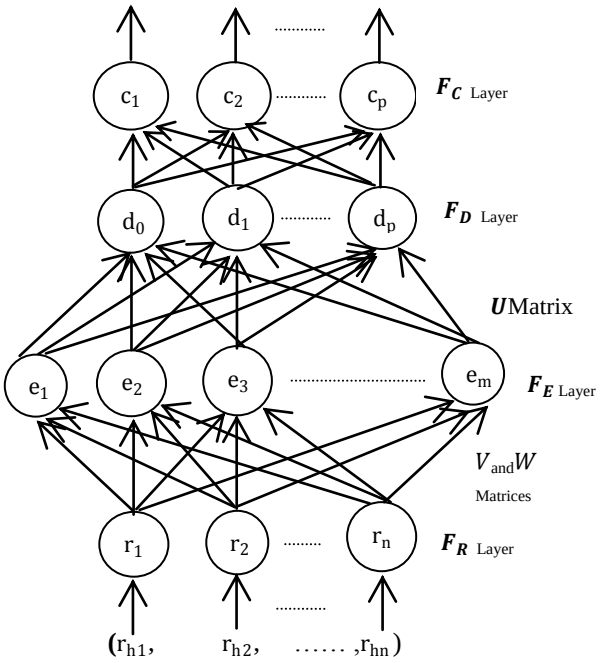

Fig1: Fuzzy Hyperline Segment Neural Network

2.2 PMFHLSNN

After learning of FHLSNN, CF based pruning procedure is incorporated to reduce the number of HLS and the resulting classifier with modification in the testing phase is referred as PMFHLSNN [23].

When, the patterns are applied to the FHLSNN for testing, it calculates the membership value for all the HLSs created during the learning phase. In FHLSNN the classification of pattern is based on the membership values. The applied pattern is classified to the class associated with the HLS that gives the highest membership value for this pattern. In PMFHLSNN, Euclidean distance is computed between the applied input pattern and centroid of the patterns falling on the HLS, to decide the class of patterns. Finally, the HLS with the smallest Euclidean distance is selected as winner and the class of that HLS is assigned to the applied input pattern

The value of CF for each HLS created during learning is depends on its usage frequency, predictive accuracy and length factor on the prediction data set and given by

$$CF_k = [(1-\alpha) - \gamma']U_k + \gamma' A_k + \alpha L_k \quad (3)$$

where, $\gamma' = \gamma(1-\alpha)$ and U_k is the usage of k^{th} HLS. $A_k$is the accuracy of k^{th} HLS, L_k is the length factor of k^{th} HLS, and $\alpha, \gamma \in [0, 1]$ are the balancing factors.

Thus, after learning of FHLSNN, the CF for each HLS created during learning is calculated by using (3) and the HLSs having CF lower than user defined threshold are removed. After pruning, remained HLSs are used for the performance evaluation of PMFHLSNN and for the rule extraction in the GA-PMFHLSNN.

3. GA BASED RULE EXTRACTOR

The Figure 2 shows the block diagram of the proposed GA-PMFHLSNN system. As shown in the block diagram, the remaining HLSs (after pruning) are used to create "open" HLSs. All the open and closed HLSs are fed to the GA for evolution. Finally, the HLSs whose dimensions are minimized by GA are used for rule extraction. The GA and GA-based rule extraction processes are explained as follows.

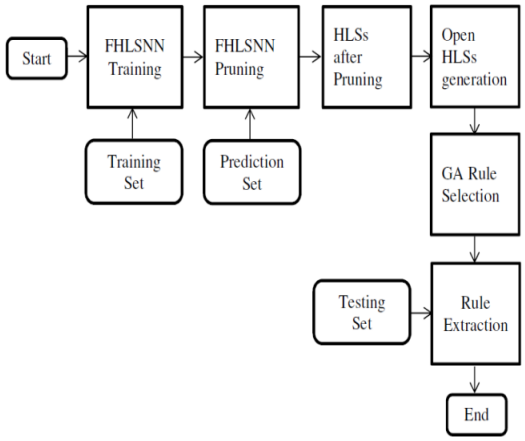

Fig 2:Block Diagram of proposed GA-PMFHLSNN System

3.1 Genetic Algorithm

1) Generating open hyperline segments: The dimensions of HLSs created during the learning of FHLSNN (as well as by PMFHLSNN) are same as of the input features. A HLS whose both (minimum and maximum) end points are defined is called as closed HLS. If a HLS has dimensions that are not defined by any one end point, then that HLS is known as "open" HLS, and that non-declared dimension is denoted as the "don't care" dimension. The "don't care" dimension is said to fully cover that particular "don't care" feature of the input space.

After pruning, all the possible combinations of open HLSs are generated from remaining HLSs. The number of possible open HLSs (except whose all dimensions are denoted as "don't care") is$(2^n - 2)$, where n is the dimension of the input space. For example, in 2-D input space, the associated open HLS are 2 and so on. The HLSs whose all dimensions are denoted as "don't care" are meaningless to consider because such HLS does not exist.

2) Evolution of hyperline segments using GA: A GA is used to evolve and select a set of HLSs that gives higher test accuracy with a less number of features. The GA proposed by Ishibuchi et al. [24] with modifications in the crossover step is used in this GA-PMFHLSNN system. Let m be the number of HLSs remained after pruning and n is the number of dimensions of each HLS. The chromosome of the population initialized by GA is a binary string that represents a solution containing all the possible open HLSs as below.

$$S = \{D_1^1, D_2^1, \ldots . D_n^1, D_1^2, D_2^2, \ldots . D_n^2, \ldots . D_1^m, D_2^m, \ldots . D_n^m\} \quad (4)$$

A value of one for the allele in S indicates that the dimension is a filled one, and its minimum and maximum points are used in the calculation of the membership function. On the other hand, a value of zero for the allele in Sindicates that the dimension is a "don't care" dimension. The length of Sis $m \times n$ and the GA search space is $m(2^n - 2)$.

In GA-PMFHLSNN, the length of Sand the GA search space are depends on the number of HLSs remained after pruning. Thus, the "curse-of-dimensionality" problem in GA-PMFHLSNN can be alleviated by pruning. This is particularly very useful when the classifier is applied to the high dimensional data sets such as Sonar data set.

To achieve the goal of higher recognition rate with minimum feature count, the GA fitness function used is given by

$$\text{maximise } f(S) = W_{NCP} \cdot NCP(S) - W_S \cdot |S| \quad (5)$$

where, $NCP(S)$is the correctly classified patterns count by the selected HLS set, $|S|$ is feature count, W_{NCP} and W_S are positive weights, and $0 < W_S \ll W_{NCP}$, $0 < W_S \leq 1$ and $0 < W_{NCP} \leq 1$.

The GA operates through a cycle of various operations as given below.

Step 1) Initialization: In all the practical application of GA, a pool of potential solutions known as population is initialized randomly. The population size varies from one application to the other.

The population strings (N_{pop}) are initialized in each generation. The strings are created by allocating 0 for "don't care" features and 1 for other features. The preferred size of N_{pop} is 100. Once population string is generated, it is examined for consecutive all ones and zeros over the span of length n. If continuous all one or zero are found over the span of length n in population string, then that string is discarded and new random population string is generated. Also each string that passes the test of all consecutive ones and zeros is compared with the previously generated stings. If newly generated string is found similar to the previously generated strings, then this newly generated string is also discarded. In this way N_{pop} numbers of population strings are generated. These population strings specify the solution which has set of HLSs having open dimensions as well as closed dimensions.

Step 2) Selection of string: Selection is the process of choosing parent strings from the current population to apply the crossover genetic operator. The objective of selection process is to find the right parent strings from current population with expectation that in the next generation their child strings will be produced with higher fitness value. In this GA, based on the selection probability, $N_{pop}/4$ pairs of parent strings are selected from the current population.

The selection probability $P(S)$of a string S in a population Ψ is given by

$$P(S) = \frac{\{f(S) - f_{min}(\Psi)\}}{\sum_{S \in \Psi}\{f(S) - f_{min}(\Psi)\}} \quad (6)$$

where,

$$f_{min}(\Psi) = \min\{f(S) | S \in \Psi\} \quad (7)$$

All population strings are arranged in descending order based on the value of probability of selection. Thus, depending upon the value of probability of selection only 50 strings are selected and allowed to participate in the process of giving birth to the new population strings by the process of crossover and mutation. The remaining 50 strings are discarded.

Step 3) Giving birth to new individual strings by crossover: The $N_{pop}/4$ pairs are the parents to give the birth to $N_{pop} \times 2$ children. In this step, four types of crossover operations are used to generate $N_{pop} \times 2$ strings. First $N_{pop}/2$ children are generated by crossover of 1st string with second and so on

with the probability of crossover is equal to 0.5. Other $N_{pop}/2$ children are generated by swapping the pair of string exactly at the center. Other $N_{pop}/2$ children are produced by crossover of 1st string with second and so on with probability of crossover equal to 0.7. The remaining $N_{pop}/2$ children are produced by mitosis of individual parent string means 1 parent can produce 1 child. The mitosis is implemented by center swapping process over a parent string.

Step 4) Mutation: In the process of crossover $N_{pop} \times 2$ children strings are generated. All the children string should be unique; hence over identical children strings one bit mutation operation is applied. Even by the process of mutation if it is not possible to avoid identical children string then they are discarded and removed from population.

As per the theory of GA, children are even better than their parents. Hence, with the expectation of higher value of fitness function from the children strings, all the children strings are evaluated for the fitness function again.

Step 5) Elitist strategy: In next iteration of GA, the string having highest fitness value is added by randomly removing any one string from the population N_{pop} .

Step 6) End test: When predefined end condition is achieved, GA stops. Otherwise, GA will go back to the initial step. In this GA, the predefined conditions are the number of iterations and recognition rate of 100 %. When, out of these two conditions if any one condition is achieved, GA will be stop.

The HLS set evolved and selected by the GA is used for further fuzzy if-then rule extraction.

3.2 Fuzzy if-then Rule Extraction

Extraction of rules is done by converting each evolved HLS into one fuzzy rule. The CF is tagged to each fuzzy if-then rule that is extracted from the corresponding HLS. Thus, each HLS is converted into one fuzzy rule and each fuzzy rule is attached with CF to show its certainty level to the domain users.

In the beginning of rule extraction, the min values and max values of each input feature are quantized by using round-off method. For this all features are considered one by one. Each input feature is converted in the range of [0 to 1] so that equal fuzzy partitions can be made. The selection of quantization level (Q) depends on the required fuzzy partitions in the quantized rules. In this work, as the Q = 5, input feature A_q is quantized to five levels as "very low, low, medium, high and very high" in a fuzzy rule. Here, interval [0, 1] is divided into Q intervals, and the input feature is assigned to the quantization points evenly with one at each of the end points using

$$A_q = (q-1)/(Q-1) \quad (8)$$

where, q = 1, . . . , Q.

The format of the fuzzy if–then rules extracted is as follows.

Rule R_k : If, r_{h1} is A_q and … h_{hn} is A_q,THEN, R_h is of class C_k, with $CF = CF_k$, for k=1, 2, …, N.

where, N is the number of HLSs, $R_h = (r_{h1}, r_{h2}, \ldots\ldots\ldots, r_{hn})$ is h^{th} input pattern having n dimension, A_q the antecedent feature value and CF_k is CF of the corresponding k^{th} HLS.

4. EXPERIMENTS AND RESULTS

These proposed approaches are implemented using MATLAB R2013a and ran on Intel core i3 2328M, 2.2GHz PC. To explore the different capabilities of a proposed pattern classification system, five benchmark datasets are selected from the "UCI machine learning repository" [25] and handwritten character database.

4.1 Benchmark Problems

The five selected benchmark datasets are Wine dataset, Iris dataset, PID dataset, Glass dataset, and Sonar dataset. The Table 1 shows the brief information of selected benchmark datasets and handwritten database.

Table 1. Information of selected datasets

Sr. No.	Datasets	Patterns	Classes	Features
1.	Wine dataset	178	03	13
2.	Iris dataset	150	03	04
3.	PID dataset	768	02	08
4.	Glass dataset	214	06	09
5.	Sonar dataset	208	02	60
6.	Handwritten dataset	1000	10	16

The features of Wine dataset are "alcohol, malic, ash, alkalinity, magnesium, phenols, flavonoids, nonflavonoid, proanthocyanins, color, hue, 1OD280/OD315 of diluted wines, and proline". The features of Iris dataset are "sepal length, sepal width, petal length, and petal width" of three plants. In this dataset "one class is linearly separable from the other two classes, but the other two classes are not linearly separable from each other". The patterns of PID dataset are overlapping patterns. Therefore classification of patterns of this dataset is a challenging task for classifiers. The features of this dataset are "number times pregnant, plasma glucose concentration, diastolic blood pressure (mm Hg), triceps skin fold thickness (mm), 2-hour serum insulin (mu U/ml), body mass index, diabetes pedigree function, and age" from two category of healthy and diabetic people. The size of Glass dataset is small but the numbers of continuous features are more. The features of this dataset are "refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron" for six types of Glass. The Sonar dataset is a high-dimensional dataset with 60 input features (s1, s2, . . .s60) form two classes of sonar signals from mine and rocks. This high dimensional dataset is helpful to test the "scalability capability" of classifier.

4.2 Handwritten Character Data Set

This database consists of 1000 handwritten Devanagari numeral character. One to ten Devanagari numerals written by 100 persons are scanned and stored in BMP format. After moment normalization [26], the rotation invariant ring-data features defined by Ueda and Nakamura [27] and extended by Chiu and Tseng [28], are extracted from the character. Ring width was set to two during the feature extraction. The extracted ring-data feature vector is a 16-dimensional.

4.3 Experimental procedure and Results

The FHLSNN, PMFHLSNN, and GA-PMFHLSNN were implemented and evaluated using tenfold cross-validation for five selected benchmark classification datasets and handwritten character database. In tenfold cross-validation, all the available patterns of dataset were divided into ten subsets. Out of these ten subsets, nine subsets were used for learning and the remaining one subset was used for performance

evaluation in terms of recognition rate. The PMFHLSNN and GA-PMFHLSNN required a prediction data set for pruning. Therefore, prediction dataset was formed by selecting any one subset from training dataset. The results were averaged by repeating this experimentation procedure ten times so that each of the ten subsets was evaluated.

FHLSNN classifier is trained using training dataset and its performance is evaluated using testing dataset in terms of recognition rate (%) and number of HLS created during the training for the different values of theta. In PMFHLSNN, CF based pruning of HLSs is used along with modification in its testing phase. Hence, the CF is calculated for all the HLS created during the training using (3). The HLSs having confidence factor lower than a user defined threshold are removed and remaining selected HLSs are used for the performance evaluation of the PMFHLSNN. Modification is applied in testing phaseto improve its recognition rate. The remained selected HLSs after pruning are also given to GA-PMFHLSNN to create "open" HLSs. All the created closed as well as open HLSs are given to the GA for evolution. The HLSs showing good classification accuracy with a less number of features are evolved and selected by GA. A set of HLSs evolved and selected by the GA is used for rule extraction. Each evolved and selected HLS is converted into one fuzzy rule.

The tuning parameter theta is varied from small to large values and the average results of percent recognition rate with number of HLSs from ten-fold cross validation are recorded for FHLSNN, PMFHLSNN, and GA-PMFHLSNN. The weighing parameters α and γ of CF are selected and adjusted in the range of 0 to 1 to get higher recognition rate. The threshold to prune the HLSs of less CF is adjusted by the user in such way that, after pruning the remaining HLSs should be of all classes for the selected dataset.

Table 2.Result of FHLSNN, PMFHLSNN and GA-PMFHLSNN for selected datasets

Datasets	Theta θ	FHLSNN		PMFHLSNN		GA-PMFHLSNN	
		RecognitionRate (%)	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS
Wine Dataset	0.03	62.22	83	67.22	8	83.33	8
IrisDataset	0.085	92.66	70	82.66	5	97.33	5
PIDdataset	0.08	49.09	352	60	7	78.31	7
Glass Dataset	0.03	57.72	102	49.09	9	65.45	9
Sonar Dataset	0.45	44.28	162	57.14	8	92.38	8
Handwritten Dataset	0.5	46.8	452	24.3	22	34.9	22

Table3.Result of FHLSNN, PMFHLSNN and GA-PMFHLSNN for Wine dataset

Theta θ	FHLSNN		PMFHLSNN		GA-PMFHLSNN	
	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS
0.03	62.22	83	67.22	8	83.33	8
0.04	61.66	78	64.44	7	79.44	7
0.1	61.66	64	67.22	6	79.44	6
0.15	61.66	58	67.77	5	78.33	5
0.3	57.22	55	64.44	5	75	5

Table4.Result of FHLSNN, PMFHLSNN and GA-PMFHLSNN for Iris dataset

Theta θ	FHLSNN		PMFHLSNN		GA-PMFHLSNN	
	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS
0.04	92.66	92	87.33	7	98.66	7
0.06	92.66	78	88.66	6	98	6
0.085	92.66	70	82.66	5	98	5
0.095	92.66	69	80.66	5	98	5
0.3	92.66	67	90	5	97.33	5

Table5.Result of FHLSNN, PMFHLSNN and GA-PMFHLSNN for PID dataset

Theta θ	FHLSNN		PMFHLSNN		GA-PMFHLSNN	
	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS
0.02	49.09	499	53.76	10	76.10	10
0.04	49.09	382	57.14	8	80.51	8
0.08	49.09	352	60	7	80.38	7
0.1	49.09	349	60.12	7	79.74	7
0.3	49.09	347	52.33	7	79.22	7

Table6.Result of FHLSNN, PMFHLSNN and GA-PMFHLSNN for Glass dataset

Theta θ	FHLSNN		PMFHLSNN		GA-PMFHLSNN	
	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS
0.02	57.72	107	49.09	9	70	9
0.05	57.72	99	47.72	9	67.72	9
0.07	57.72	97	45.45	8	66.36	8
0.08	57.72	96	45.09	8	66.36	8
0.2	57.72	96	43.63	8	66.36	8

Table 7.Result of FHLSNN, PMFHLSNN and GA-PMFHLSNN for Sonar dataset

Theta θ	FHLSNN		PMFHLSNN		GA-PMFHLSNN	
	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS
0.1	44.28	187	59.52	9	91.90	9
0.55	44.28	155	57.61	8	89.52	8
0.8	44.28	134	54.76	7	94.76	7
0.85	44.28	128	49.52	6	92.38	6
0.9	44.28	124	49.52	6	92.38	6

The Table 2 shows the classification accuracy in terms (%) recognition rate and number of HLSs for tuned value of theta using five benchmark problems and handwritten recognition database for FHLSNN, PMFHLSNN, and GA-PMFHLSNN classifier. The summarized experimental results show that the proposed GA-PMFHLSNN system gives a very good classification performance with small number of HLSs. This means that many of HLSs in FHLSNN could be removed without adversely affecting its classification accuracy. Thus, this proposed approach improves the classification performance with reduced number of HLSs and hence the network complexity.

The Table 3 to Table 8 shows the classification accuracy in terms of (%) recognition rate and number of HLSs for the various values of theta for Wine data set, Iris dataset, PID dataset, Glass dataset, Sonar dataset, andhandwritten character database respectively using FHLSNN, PMFHLSNN, and GA-PMFHLSNN classifier. As shown in the Table 3 for the Wine dataset, Table 5 for PID dataset and Table 7 for Sonar data the PMFHLSNN give higher recognition rate with very less HLSs. The GA-PMFHLSNN improves this recognition rate to highest value. Thus, the pruned network can give more recognition rate than the original FHLSNN, because the proposed pruning approach prunes the HLSs which were responsible for misclassification.

As shown in the Table 4 for Iris dataset, Table 6 for Glass dataset and Table 8 for handwritten character dataset, the PMFHLSNN gives slightly less recognition rate as compare to FHLSNN. The GA-PMFHLSNN system improves this to higher value. In PMFHLSNN and GA-PMFHLSNN, HLSs used are very less.Thus, the GA-PMFHLSNN shows better classification performance than FHLSNN and PMFHLSNN.and a very comparative classification performance as compare to the reportedstate of art classifiers [29] shown in Table 9. The number of HLSs used by PMFHLSNN and GA-PMFHLSNN are same. Thus, the classification accuracy of GA-PMFHLSNN can be increased by rearranging the information assimilated in the pruned network.

Table 8.Result of FHLSNN, PMFHLSNN and GA-PMFHLSNN for Handwritten dataset

Theta θ	FHLSNN		PMFHLSNN		GA-PMFHLSNN	
	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS	Recognition Rate (%)	Number of HLS
0.4	46.8	455	22	22	33.5	22
0.5	46.8	452	24.3	22	34.9	22
0.55	46.8	451	23.3	22	34.1	22
0.6	46.8	451	23.6	22	34.9	22
0.65	46.8	451	23.1	22	34.3	22

Table9.Recognition rate (%) for different Classification systems using different dataset

Methods	Recognition Rate (%) for the various Dataset				
	Glass Dataset	Wine Dataset	Iris Dataset	PID Dataset	Sonar Dataset
C4.5	70.23	91.09	91.60	71.02	-
C4.5 Rules	67.96	91.90	91.58	71.55	-
ITI	67.49	91.09	91.25	73.16	-
LMDT	60.59	95.40	95.45	73.51	-
CN2	70.23	91.09	91.92	72.19	-
LVQ	60.69	68.90	92.55	71.28	-
OCI	57.72	87.31	93.89	50.00	-
Nevprop	44.08	95.41	90.34	68.52	-
FMN	69.07	96.85	95.60	68.42	81.20
FHLSNN	57.72	62.22	92.66	49.09	44.28
PMFHLSNN	49.09	67.22	82.66	60.00	57.14
GA-PMFHLSNN	65.45	83.33	97.33	78.31	92.38

Lastly, these evolved and selected HLSs were used to extract fuzzy if-then rules by the process of quantization. The “don’t care” antecedents are used to achieve compact and comprehensible rules with tagged CF.

Table 10.Rule extracted for Wine Dataset with don’t care antecedent

IF	Alcohol is “high to very high”
	Malic is “don’t care”
	Ash is “medium”
	Alkalinity is “low”
	Magnesium is “don’t care”
	Phenols is “don’t care”
	Flavonoids is “don’t care”
	Nonflavonoid is “don’t care”
	Proanthocyanins is “don’t care”
	Color is “medium to high”
	Hue is “low to high”
	1OD280/OD315 of diluted wines is “medium to high”
	Proline is low to “very high”.
THEN	Output is class 1 with CF of 0.8699.

The Table 10 shows a sample rule extracted for the Wine dataset with “don’t care” antecedents. This rule for Wine dataset can be simplified to get compact and comprehensible rule by neglecting “don’t care” antecedents as presented in the Table 11. Similarly, the Table 12 present the rule extracted for the handwritten character dataset in compact form. Thus, the GA-PMFHLSNN system can extract a set of “compact and comprehensible” rule with high classification accuracy using pruning approach and “don’t care” antecedent.

Table 11.Compact Rule extracted for Wine Dataset without don’t care antecedent

IF	Alcohol is “high to very high”
	Ash is “medium”
	Alkalinity is “low”
	Color is “medium to high”
	Hue is “low to high”
	1OD280/OD315 of diluted wines is “medium to high”
	Proline is “low to very high”.
THEN	Output is class 1 with CF 0.8699.

Table 12.Compact Rule extracted for Handwritten Dataset without don't care antecedent

IF	feature 2 is "medium high"
	feature 7 is "medium"
	feature 10 is "low"
	feature 11 is "very low"
	feature 13 is "very low"
	feature 16 is "very low"
THEN	Output is of class 2 with CF of 0.8352.

5. CONCLUSIONS

A geneticalgorithm based rule extractor is proposed to extract a small set of compact and comprehensible fuzzy if-then rules with higher classification accuracy from PMFHLSNN. The performance of FHLSNN, PMFHLSNN and GA-PMFHLSNN are evaluated using tenfold cross validation for five benchmark datasets and handwritten character recognition database. The experimental outcome shows the advantages and the important properties of the proposed approaches as follows.

1. The pruning based on CF with the effect of length factor gives comparable performance for all the selected databases with a very less number of HLSs.
2. The proposed modification in the testing phase of FHLSNN shows the remarkable improvement in the classification accuracy.
3. The proposed GA rule extractor improves the classification accuracy of GA-PMFHLSNN.
4. This proposed rule extractor minimizes the features count of the extracted rules with higher recognition rate by using the "don't care" approach.
5. The GA-PMFHLSNN system alleviates the 'curse of dimensionality' problem because of pruning.

The performance of these approaches can be improved by using the GA for optical selection of threshold value and optimal number of hyperline segments in the pruning process. In GA-PMFHLSNN, use of don't care antecedents and GA affect the incremental learning capability of FHLSNN. Therefore, the approach can be revised to design GA-PMFHLSNN with incremental learning.

6. REFERENCES

[1] Jang, J., Sun, C. and Mizutani, E. 2008Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Pearson Prentice Hall, South Asia, New Delhi, India.

[2] Simpson, P. "Fuzzy min–max neural networks—Part 1: Classification," IEEE Trans. on Neural Networks, Vol. 3, No. 5, pp. 776–786, Sep. 1992.

[3] Simpson, P. "Fuzzy min–max neural networks—Part 2: Clustering," IEEE Trans. on Fuzzy systems, Vol. 1, No. 1, pp. 32–45, Feb. 1993.

[4] Kwan, H.and Yaling, C. "A fuzzy neural network and its applications to pattern recognition," IEEE Trans. on Fuzzy Systems,Vol. 2, No. 3, pp. 185-192, Aug. 1994.

[5] Gabrys, B. and Bargiela, A. "General Fuzzy Min-Max Neural Network for Clustering and Classification," IEEE Trans. on Neural Networks, Vol.11, pp. 769-783, May 2000.

[6] Kulkarni, U., Sontakke, T. and Randale, G. 2001 Fuzzy hyperline segment neural for rotation invariant handwritten character recognition. In Proceeding of the Int. joint conf. on neural networks: IJCCNN'01 held in Washington DC, USA, pp. 2918–2923.

[7] Kulkarni, U., Sontakke, T. and Kulkarni, A. "Fuzzy Hyperline Segment Clustering Neural Network," in IEE Electronics Letters Vol. 37, No. 05, pp. 301-303, Mar. 2001.

[8] Patil, P. and Sontakke, T. 2006"Rotation, scale and translation invariant handwritten Devanagari numeral character recognition using general fuzzy Neural Network",Journal of Pattern Recognition, Vol. 40, pp. 2110–2117, 2007.

[9] Wang, Y. and Fan, K. 1996Applying genetic algorithms on pattern recognition: an analysis and survey. In International Conference on Pattern Recognition, Vienna, Vol. 2, pp.740-744.

[10] Cho, S. 1999Pattern recognition with neural networks combined by genetic algorithm, Fuzzy Sets and Systems, Vol.103, No. 2, pp.339-347, May 1998.

[11] Oh, S., Pedrycz, W. and Park, B. "Multilayer hybrid fuzzy neural networks: synthesis via technologies of advanced computational intelligence", IEEE Trans. on Circuits and Systems-I, Vol.53, No.3, pp. 687-703, Mar. 2006.

[12] Sabo, D. and Yu, X. 2008A New Pruning Algorithm for Neural Network Dimension Analysis in IEEE International Joint Conference on Neural Networks, Hongkong, China, pp. 3313-3318.

[13] Marsland, S., Shapiro, J. and Nehmzow, U. "A self-organizing network that grows when required", Neural Networks, Vol. 15, pp.1041- 1058, Special issue 2002.

[14] Reed, R. "Pruning algorithms-A survey," IEEE Trans. on Neural Networks, Vol. 4, pp. 740-747, 1993.

[15] Benitez, J., Castro, J. and Requena, I. "Are artificial neural networks black boxes?", IEEE Trans. on Neural Networks, Vol. 8, No. 5, pp.1156–1164, Sep. 1997.

[16] Kolman, E. and Margaliot, M. "Are artificial neural networks white boxes?", IEEE Trans. on Neural Networks, Vol. 16, No. 4, pp.844–852, Jul. 2005.

[17] Andrews, R., Diederich, J. and Tickle, A. B. "Survey and critique of techniques for extracting rules from trained artificial neural networks", Knowledge Based Systems, Vol. 8, No. 6, pp.373-389, Dec. 1995.

[18] Anas, Q., Lim, C.P. and Tan, K.S."A modified fuzzy min–max neural network with a genetic-algorithm-based rule extractor for pattern classification", IEEE Trans. on Systems, Man, and Cybernetics—Part A: Systems and Humans, Vol. 40, No. 3, pp.641-650, May2010.

[19] Yap, K., Lim, C. and Au, M."Improved GART neural network model for pattern classification and rule extraction with application to power systems", IEEE Trans. on Neural Networks, Vol. 22, No. 12, pp.2310-2323, Dec. 2011.

[20] Chen, Y., Pal, N. and Chung, I."An integrated mechanism for feature selection and fuzzy rule extraction for classification", IEEE Trans. on Fuzzy Systems, Vol. 20, No. 4, pp. 683-698, Aug. 2012.

[21] Yang, X., Mingming, Z., Quanhui, L. and Xiaofeng, W."A genetic algorithm based multilevel association rule mining for big datasets", Mathematical Problems in Engineering, Vol.2014, pp.1-9, Aug. 2014.

[22] Bagal, S.and Kulkarni, U.2014. Modified Fuzzy hyperline segment neural network for pattern classification and recognition. In Proceeding of the International Conference of Computational Intelligence and Intelligent Systems (ICCIIS'14), London, UK, Vol.1.

[23] Bagal, S.and Kulkarni, U."Pruned modified fuzzy hyperline segment neural network and its application to pattern classification", International Journal of Computer Application, Vol. 93, No. 12, pp.43–50, May 2014.

[24] Ishibuchi, H., Murata, T. and Turksen, I. "Single-objective and two- objective genetic algorithms for selecting linguistic rules for pattern classification problems", Fuzzy Sets and Systems, Vol. 89, No. 2, pp.135–150, Mar. 1996.

[25] Murphy, P. and Aha, D. 1995UCI Repository of Machine Learning Databases, (Machine-Readable Data Repository). Irvine, CA: Dept. Inf. Computer Sci., Univ. California.

[26] Perantonis, S.and Lisboa, P. "Translation, rotation and scale invariant pattern recognition by high-order neural networks and moment classifiers," IEEE Trans. on Neural networks, Vol. 3, No. 2, pp. 241-251, 1992.

[27] Udea, K. and Nakamura, Y. 1984Automatic verification of sealimpression pattern. In Proceeding of 9th Int. Conf. on pattern recognition.Vo1. 2, pp. 1019-1021.

[28] Chiu, H. and Tseng, D. "Invariant handwritten Chinese character recognition using fuzzy min-max neural networks," Pattern Recognition Letters, Vol. 18, pp. 481-491, 1997.

[29] Hoang, A.1997 Supervised classifier performance on the UCI data set, in Department of Computer Science, M. Sc. Thesis, University of Adelaide, Australia.