Iranian Journal of Fuzzy Systems

Volume 20, Number 6, (2023), pp. 155-169

Original paper

University of Sistan and Baluchestan

Intelligent data classification using optimized fuzzy neural network and improved cuckoo search optimization

P. Patro[1], K. Kumar[2], G. S. Kumar[3] and A. K. Sahu[4]

[1] *Department of Engineering Mathematics, Koneru Lakshmaiah Education Foundation, Hyderabad, Telangana, India, 500075*
[2] *Department of Applied Science and Humanities, MIT School of Engineering, MIT Art Design and Technology University, Loni Kalbhor, Pune, India*
[3] *Department of Engineering Mathematics, Koneru Lakshmaiah Education Foundation Vaddeswaram, Guntur, Andhra Pradesh, India*
[4] *Department of Computer Science and Engineering, Amrita School of Computing, Amaravati Campus, Amrita Vishwa Vidyapeetham, Amaravati, Andhra Pradesh, 522503, India*

pramoda.mtech09@gmail.com, krishna.icoer@gmail.com, drgsk006@kluniversity.in, adityasahu.cse@gmail.com

Abstract

In data mining, classification is one of the most critical steps in predicting the target class. It is performed by an improved model in existing work in which feature selection is performed based on the bat optimization method to increase the classification accuracy. This study uses an enhanced neural network for classification, including intuitive, interpretable correlated-contours fuzzy rules. Further, a practical model is created based on the extraction of fuzzy rules, where data partitioning is performed via a similarity-based directional component. However, the dataset used for experimentation is noisy and incomplete data values. Due to incompleteness, knowledge discovery is obstructed, and the classification results are affected. Here bat provides very slow convergence and easily falls into local optima. To solve this issue, an improved framework is introduced in which missing value imputation is performed by using k means clustering, and then for feature selection, an improved cuckoo search optimization is used. An enhanced classifier based on fuzzy logic and alex net neural network structure (F-ANNS) is used for classification, and hybrid ant colony particle swarm optimization (HASO) is used for optimizing parameters of the alex net neural network classifier. The results show that the proposed work is more effective in precision, recall, accuracy, and f-measure as shown by experimental results.

Keywords: Hybrid ant colony particle swarm optimization, AlexNet neural network, cuckoo search, missing data Imputation, artificial neural network.

1 Introduction

Classification is among the most commonly observed human activity tasks for decision-making. A classification issue arises when assigning an attribute to a predefined category or class is appropriate. Classification is used in various fields like pattern recognition, disease prediction, face recognition, etc. Recently it has been found that many machine learning (ML) algorithms like support vector machines (SVM), clustering, and various neural networks are proposed for classification however, there is a need to improve the accuracy of prediction results [17]. To overcome these issues, classification problems are solved using fuzzy neural networks and previously have been used to solve several problems, including pattern classification, time series forecasting, and dynamic system modelings. neural networks and fuzzy logic are complements of each other [5, 33]. Consequently, they can be merged into a promising system to benefit from fuzzy and neural networks. Previously some learning algorithms are introduced in fuzzy neural network [1, 3]. Usually, learning is accomplished in two steps. The network topology is defined first. This process includes identifying fuzzy

Corresponding Author: P. Patro
Received: February 2023; Revised: July 2023; Accepted: October 2023.
https://doi.org/10.22111/IJFS.2023.44767.7887

sets for each input variable, deciding the required number of neurons, and creating network connections. Clustering and evolutionary optimization are the most widely used approaches for structure specifications. Free parameters are calculated after the network structure is specified. This step has already used several approaches, such as reinforcement learning, gradient-based methods, genetic algorithms, and least squares [1, 3, 17]. Fuzzy inference systems (FIS) have three basic components: a rule base containing a collection of fuzzy rules, a database describing the fuzzy rules with membership functions, and a reasoning method that provides rules to perform an inference procedure and specified data to produce a logical production or conclusion. An innovative learning technique for fuzzy logic neuron-based networks that may produce accurate and transparent models is presented in existing work. The learning method is based on regularisation theory and extreme learning machine concepts to achieve a low time complexity and produce sparse and accurate models. The resulting network topology can be used to obtain a condensed collection of imprecise fuzzy rules. However, it does not suitable for multiclass classification. To overcome these issues in existing work approximation of function was performed, which is named as enhanced neural network including intuitive, interpretable correlated-contours fuzzy rules (EC-FR) [11, 12], in which an appropriate framework was built based on fuzzy rules extraction where data partitioning is performed via a similarity-based directional component. However, the dataset used for this work's experimentation contains noisy and incomplete data values. Due to this incompleteness, knowledge discovery will be hindered, and classification result is also affected. Poor classification performance is exhibited by a simple neural network architecture in fuzzy neural networks [7, 18]. To overcome these issues for classification, an improved framework with four stages is introduced in this work for avoiding these problems. Missing data imputation-based clustering is the first preprocessing step. In the second step, to select features, improved cuckoo search optimization is used [4, 32] and in the third stage, for classification, an improved classifier based on fuzzy logic and AlexNet neural network structure(F-ANNS) is used. Hybrid ant colony particle swarm optimization (HASO) is used for optimizing parameters in the training phase in the last stage [5, 13, 19, 26]. The GWO algorithm models the leadership structure and hunting strategy of grey wolves in nature. Four different varieties of grey wolves, including alpha, beta, delta, and omega, are used to mimic the leadership hierarchy. The three fundamental components of hunting–looking for prey, surrounding prey, and attacking prey–are also used. The technique is then tested against 29 well-known benchmark functions, and the findings are confirmed by a comparison with Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Differential Evolution (DE), Evolutionary Programming (EP), and Evolution Strategy (ES) [19]. The Whale Optimization Algorithm (WOA) imitates the humpback whales' social behaviour. The bubble-net searching tactic served as the basis for the algorithm. Bubble-net feeding is used to describe the humpback whales' foraging activity. Hunting krill or small fish in schools near the surface is preferred by humpback whales. This foraging behaviour has been seen to involve the formation of unique bubbles along a circular or "9"-shaped path [19]. The overall organization of the paper is as follows. After the introduction, we present the review of different methods for classification in Section II. Section III methodology of the proposed model is discussed; Section IV discusses the simulation results. The conclusion and work intended for the future are discussed in Section V.

2 Literature survey

Various fuzzy neural network systems are reviewed in this section for function approximation. Alturki (2018) used Fuzzy Neural Network (FNN) to present new traffic sign detection and recognition technique with three stages. Images are segmented in the first stage for extracting ROIs. In most cases, adaptive thresholding is used to perform segmentation to overcome color segmentation problems. Traffic shapes are detected in the second stage. Octagonal, rectangular, circular, and triangular are the limited geometric forms of traffic signs. From ROIs computed in the first stage, traffic shapes are identified using this geometric information. The information included in pictograms is used for recognizing traffic signs in the third stage. However, this work extracts six types of features. For performing recognition, an FNN classifier is given with these features. Support Vector Machine (SVM), Artificial Neural Network (ANN), and FNN classifiers are tested with a new descriptor. German traffic sign detection and recognition benchmark dataset is used for experimentation and to test proposed methods. Better results are obtained when compared with state-of-the-art techniques. Kermany et al. [16] introduced a new dual-insulating detection method with various smart grid (SG) link points for microgrids (MGs). This technique is based on islanding probability (PoI) measured at the SG side and then sent to microgrid central control (CCMG). At SGs sides, active, passive, and utility signals are measured using communication islanding detection techniques for computing PoI values. An artificial neural network (ANN) with discrete wavelet transform is used for processing these utility signals. The high accurate fuzzy network-based technique is used for re-computing PoI and ANN if it is greater than the significance level suggesting a high insulation possibility. An adaptive neuro-fuzzy inference system is used for determining fuzzy parameters. If PoI is high, only active islanding is performed in the proposed technique and PoI fuzzy is proportional to disturb signal amplitudes.

Ranjan & Prasad [28] used context-semantic features for implementing the connectionist classification technique and for text classification, they proposed an LFNN-based incremental learning algorithm. The proposed technique considers the dynamic dataset for classification, which enables the classifier's dynamic learning models. Back Propagation Lion (BPLion) neural network is adopted in the incremental learning process and for feasible weight selection, lion algorithm (LA) and fuzzy bounding are included in it. Existing methods like I-BPLion, FI-BP, and I-BP are used for comparing the proposed methods effectiveness with respect to error and accuracy in comparative analysis. In LFNN, classification accuracies of 81.49 %, 83.41 %, 88.76 %, and 95 % are possible, with minimum error values of 8.11, 7.49, 3.02, and 4.92 for 20 newsgroup, reuter datasets, WebKB, and RCV1 as shown in comparative analysis. Deng et al. [6] rectified the fixed representation shortcomings by introducing fuzzy learning concepts into DL. The proposed fuzzy model is a deep neural hierarchical network that derives information from neural and fuzzy representation. Final data representation needs to be classified and formulated by combining knowledge developed from these two perspectives. Brain MRI segmentation, high-frequency financial data prediction, and image categorization tasks are used to verify the proposed models' effectiveness. These tasks have a higher degree of raw data ambiguity. Ding et al. [7] combined ensemble learning technology and fuzzy neural networks to propose new multiclass classification technology. Various fuzzy neural networks, including fuzzy wavelet neural networks and fuzzy TS neural network, are assumed as sub-classifiers. Fuzzy neural network parameters are trained using gradient descent and Restricted Boltzmann machines algorithms. Sub-classifier weights are computed using particle swarm optimization and hypothesis margin-based adaptive boosting (AdaBoost) algorithm. The validity of the proposed method is illustrated using numerical simulation. In the same year, Zobeidi et al. [20] adapted a novel neural system classification model to characterize the writings and compared it to certain comparative methods, as shown in Table 1. To this end, right off the bat, preprocessing is led on the words, which incorporates five stages. From that point onward, the element extraction stage is applied, comprising three steps, with the principal component analysis (PCA) technique utilized for dimensionality decrease. At last, a managed learning strategy, particularly the Multi-Level Fuzzy min-max neural system classifier (MLF), is used for content grouping. The reproduction aftereffects of the investigations, contrasted with three other guideline techniques, demonstrate that the MLF strategy has a high exactness of 94% for the Reuters-21578 dataset and 95% for the 20-newsgroup dataset. Likewise, the suggested method's execution time is not precisely another directed learning calculation. Mudiyanselage et al. [21] created another profound fluffy neural system to deal with vulnerability in quality information to produce helpful information for precise illness analysis. Another half calculation is intended to preprocess data and select instructive qualities for exact malignancy identification. Different tests utilizing six distinctive malignancy informational indexes show that the new strategy has preferred and progressively dependable execution over other customary order techniques with various quality choice strategies.

Table 1: Inferences are drawn from the existing research

Name of the Author	**Methodologies**	**Advantages**	**Disadvantages**
Kermany et al. (2016)	Artificial neural network (ANN) with discrete wavelet transforms	Provides better performance	Time-consuming nature
Deng et al. (2016)	Deep neural hierarchical network	More Flexible	Cost expensive
Ding et al. (2017)	Ensemble learning technology	Provide better performance	Increases time consumption.
Zobeidi et al. (2017)	Multi-Level Fuzzy min-max neural network classifier (MLF)	Reduces the time consumption	Higher error rate
Alturki (2018)	Fuzzy Neural Network	Provides better performance than other methods	Time consuming nature
Ranjan and Prasad (2018)	connectionist classification approach	Provides a minimum error rate	Does not test for real-time data
Mudiyanselage et al. (2019)	Deep fuzzy neural network	Reliable performance	Need to use another classifier

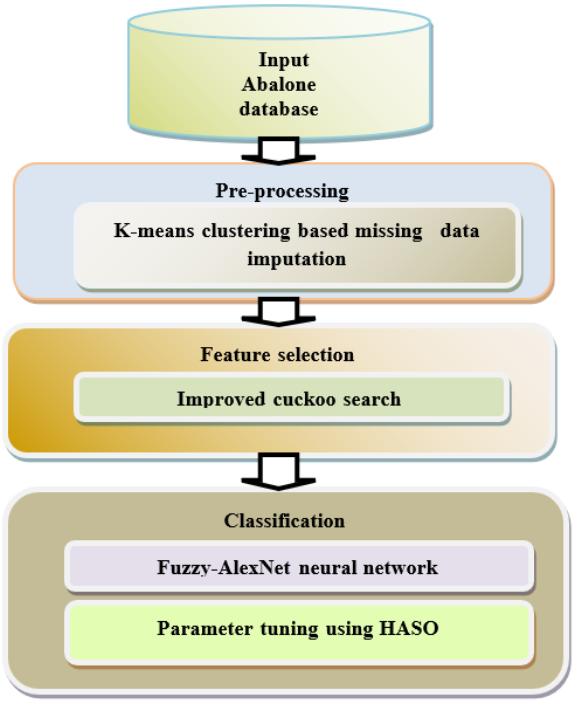


Figure 1: Architecture of the proposed work

Material and methods

3 Proposed methodology

A fuzzy neural network hybrid system based on the proposed model is discussed in this section which is having four modules as shown in Figure 1. Missing data imputation using K-means clustering is the first preprocessing step in which some values could be missed in the input databases which will affect the classification result, so here imputing those all-missing values of samples. Input samples can have many features if we take all for the computation, it will take more time to produce the result, so to reduce the computation time, in the second step, a significant feature of the input samples will be selected using an improved cuckoo search algorithm. And in the third stage, for classification, an improved classifier based on fuzzy logic and AlexNet neural network structure(F-ANNS) is used for those selected features will be the input. Hybrid Ant Colony Particle Swarm Optimization (HASO) is used to optimize F-ANNS parameters in the last training phase stage.

3.1 K-means clustering-based data imputation method

Clustering is a process of grouping an abstract or physical set of objects into similar objects classes. An instance of the entire dataset represented as S is grouped using the K-means clustering technique in this work. The whole database S is separated into clusters with similar instances. If there is one discrete attribute in S, similarities of these discrete attributes are computed using a simple matching technique and for continuous attributes, euclidean distance is used for processing. Distance between cluster center and instance are mixed one, which is a continuous and discrete attributes combination. This is because instances with incomplete data are more likely to have the same target value attribute as an example that is close to it according to the concept of the euclidean distance principle. So, the clustering technique is adopted to separate the entire dataset into clusters according to their distance difference. For every cluster, missing values are dealt with nonparametric techniques. Because of its effectiveness and simplicity, K-means as a clustering algorithm is adopted here [8].

3.1.1 Kernel function imputation strategy

Because of effective computation, stability, and robustness, missing values are dealt with effectively using a kernel function imputation technique. In the statistical area, kernel nonparametric regression imputation also corresponds to the kernel function. For example, missing values are imputed using the kernel method. To make an inference of datas distribution function (DF), variance, and mean, this work proposes a kernel function nonparametric random imputation technique [15, 30].

Assume nd dimensional vector as X and a variable influenced by it as Y. X is represented as a conditional attributes or factor attributes (FA) and Y as target attributes (TA). There would not be any missing values in X and there exists a missing value in Y. For simplification,$(X_{i1}, X_{i2}, .., X_{id}, Y_i, \delta_i), = 1, .., n$ represents the dataset, where, the indicator function is given by δ_i and it will be zero, if Y_i is not missing. In real-time, X and Y satisfy,

$$Y_i = m(X_{i1}, X_{i2}, .., X_{id}) + \epsilon_i, i = 1, ...n. \tag{1}$$

Where the unknown function is represented as $m(X_{i1}, X_{i2}, .., X_{id}, Y_i, \delta_i)$ and random error is represented as ϵ_i and it has variance σ^2 and 0 mean. It can also be stated as a relationship between Y and X and is unknown. In the case of linear unknown function m(.), the deterministic imputation method exhibits better performance in making Ys mean inference. It indicates that, in making an inference to Ys distribution function, the random imputation technique must be used if the unknown function m(.) is arbitrary. This is due to the fact that, in various complex practical conditions, the unknown function m(.) is not a linear function. For example, if Y_i is missing in expression (1), kernel method is utilized for computing $m(X_i) = m(X_{i1}, X_{i2}, , X_{id})$ values as,

$$\hat{m}(X_{i1}, X_{i2}, .., X_{id}) = \frac{\sum_{j=1}^{n} \delta_j Y_j \prod_{s=1}^{d} k\left(\frac{X_{is} - X_{js}}{h}\right)}{X}, i = 1, 2, ..., n. \tag{2}$$

$$\sum_{j=1}^{n} \delta_j \prod_{s=1}^{d} k\left(\frac{X_{is} - X_{js}}{h}\right) + n^{-2}. \tag{3}$$

In the above equation (2), $\hat{m}(X_{i1}, X_{i2}, .., X_{id})$ represents the estimated value for the missing attribute values of instance X_i. The summation and product operators are used to calculate the weighted average of the available attribute values (Y_j) based on their distances to the missing instance (X_i). The kernel function k is applied to normalize the distances between X_{is} and X_{js} using the bandwidth parameter h. The variable n represents the total number of instances in the dataset, and d represents the number of attributes. The term δ_j is an indicator variable that equals 1 if the j-th instance has a non-missing value for all attributes, and 0 otherwise. In the equation (3), the term j is an indicator variable that equals 1 if the j-th instance has a non-missing value for all attributes, and 0 otherwise. The product of the kernel function k applied to the normalized differences $(X_{is} - X_{js})/h$ for each attribute s = 1 to d. The variable n represents the total number of instances in the dataset, and d represents the number of attributes. Finally, the term n^{-2} represents the reciprocal of n squared. The kernel function is represented in expression (4) and may take various forms. The Gaussian kernel is the most commonly used one.

$$k(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right). \tag{4}$$

$$k(x) = \begin{cases} 1/2 & |X| \leq 1 \\ 0 & |X| > 1 \end{cases}. \tag{5}$$

In the learning process, if optimum bandwidth is received, there would not be any difference in kernel function type selection. This research uses a Gaussian kernel function as a common method for calculating the average of the neighbouring points.

3.1.2 Missing value imputation algorithm based on clustering

This sub-section describes the CMI (Complete the Missing Information) model for filling in missing data. On factor attributes, the clustering method divides the entire dataset into clusters. After clustering, missing-valued instances are filled in every cluster using the kernel method. The continuous condition of Y is dealt by using the kernel technique in this paper. For missing value completion in the discrete condition of Y, nearest neighbor method (NN) is used. According to points mentioned above, the following algorithm presents the CMI algorithm.

All missing values are imputed in the input database based on the procedure mentioned above. The feature selection step is given with this output database as input.

Algorithm 1: CMI algorithm

Data: Incomplete-estimated data set S, k;
Result: Complete S Dataset;
Randomly divide the dataset into different groups of clusters (C1, C2, ..., Ck).
for *each cluster Ci* **do**
  for *each missing-valued instance -Ik in cluster Ci* **do**
    Compute $\hat{m}(X)$, I, and R using Eq. (2).
  for *for each missing-valued instance Ik in cluster Ci* **do**
    Fill the missing value in Ik using m-hat(X)I.
  S' ∪(i=1 to k) C(i)U(i)

3.2 Feature selection using improved cuckoo search optimization

For step size α and probability factor p_a, modification of the standard cuckoo search algorithm is done to enhance system stability and it is termed as an improved cuckoo search algorithm (ICS). Step size α and probability factor p_a values are fixed in traditional CS algorithms in which the following two conclusions or problems are found.

- Algorithm performance is reduced with the huge value of α and a small value of p_a and obtaining optimum solutions, it requires more iterations.
- Algorithm performance is enhanced with a small value of α and a huge value of p_a, increasing convergence speed. But it will not find the optimum solution.

To rectify the above mentioned limitations, the improved cuckoo search algorithm is initialized with large values of p_a and α. But, below mentioned expressions are used for updating their values gradually, which tends to fine-tune the solution vectors. Model is stabilized using this way [4, 14, 15, 31, 32]. The following expressions are used for modifying the p_a and α values.

$$P_a(it) = P_{amax} - \frac{it}{N_i}(P_{amax} - P_{amin}), \tag{6}$$

$$\alpha(it) = \alpha_{max} \exp(K * it), \tag{7}$$

$$K = \frac{1}{N_i} Ln\left(\frac{\alpha_{max}}{\alpha_{min}}\right). \tag{8}$$

Here the equation (6), calculates the value of p_a at iteration it based on the maximum p_{amax} and minimum values of and the current iteration number (it). It uses a linear decreasing function to decrease p_a over the iterations. N_i represents the total number of iterations. And equation (7),calculates the value of alpha at iteration it based on the maximum α_{max} and minimum α_{min} values of alpha, and the parameter K. It uses an exponential increasing function to increase alpha over the iterations. The value alpha max represents the maximum value that alpha can take. And equation (8), calculates the value of K based on the total number of iterations (N_i), the maximum α_{max} and minimum α_{min} values used in equation (7). It uses the natural logarithm function (L_n) to calculate the logarithm of the ratio $\frac{\alpha_{max}}{\alpha_{min}}$.K is used as a scaling factor in the calculation of alpha in equation (7).

Selected the important features using this algorithm and for enhancing the accuracy of classification, these features are given as input to the next step.

3.3 Classification using fuzzy AlexNet neural network

AlexNet architecture

AlexNet is an 8-layer system, with its superior classification advantages. The first two layers use methods for calculation like convolution (Conv), ReLU, max-pooling, and normalization. The second layer produces a conversion operation involving 256 feature vectors, with a kernel size of 5, and a stage of 1, and the remaining parameters are the same as the first layer. Only convolution and ReLU operations are performed on the third and fourth layers in this research. It carries out the same procedure as the first on the fifth layer, but somehow doesn't standardize it. The fifth layer results in a long vector and input into the conventional neural network, In which the first two layers of complete link

Algorithm 2: Improved cuckoo search algorithm

Data: Abalone dataset
Result: Selected features

```
Initialize Objective function f(x);
Initialize Initial population of 'n' host nests x_1, x_2, ..., x_dT;
Initialize Parameters: pa_min, pa_max, α_min, α_max, λ_val, N_i;
i = 1;
while i ≤ N_i do
    Update parameters P_a, α, and K using Eqs. (6)-(8);
    generation = 1;
    while generation ≤ Max_Generation do
        Randomly obtain a cuckoo (say i) using Levy flights;
        Evaluate its fitness F_i
        Randomly select a nest among n (say j);
        if F_i > F_j then
            Replace nest j with the new solution;
        Abandon a fraction (p_a) of worst nests and build new ones at new locations;
        Maintain the best solutions (or quality solutions);
        Compute the current best solution and rank the solutions;
        Increment generation by 1;
    Increment i by 1;
Perform result processing;
```

kernels are all 4096. The last layer produces 1000 nodes and uses the regression function SoftMax to get the label value [4, 31, 32].

Fuzzy AlexNet neural network

Inside the classic backpropagation network structure, the input layer is the first layer of the model, while the flexible input layer (fuzzy layer), hidden layer, and output layer are the second, third, and fourth layers, respectively. This FNN model inputs ten fuzzy truth values and produces two fuzzy truth outcomes [2]. The output layer (i.e. fifth layer) decrypts the value of fuzzy truth to the correct value which is the actual p_i^n of i^{th} nodes n^{th} prediction. The FNN model makes use of to denote the input of the i^{th} artificial neuron in the i^{th} level layer, o_j^i to denote the output of the i^{th} artificial neuron in the i^{th} level layer, w^i to indicate the weight of relations between layer two and layer three to represent the weight of the 3rd- to 4th-layer links and $w_i^{''}$ to denote connection weights between the 4th and 5th layers.

Input layer

The input layer gathers a node's states in real-time and generates state feedback values as input parameters when a prediction of p_i^n is required. The input layer simply imports state feedback variables into the FNN in real-time, and the formula for the input-output is as follows:

$$o_i^1 = I_i^1 = x_i, \tag{9}$$

where x^i equates the values of F, M, P, UF and UM in respectively. This equation (9) represents the first-layer membership function for input data. It states that the degree of membership o_i^1 for input data I_i^1 is equal to the actual input value(x_i).

Fuzzy layer

Using a membership function, the fuzzy layer converts the proper values into fuzzy truth values. The formulae for input and output are shown as follows:

$$I_i^2 = o_i^1, \tag{10}$$

$$o_i^2 = U_A(I_i^2), o_i^2 \in [0, 1]. \tag{11}$$

The equation (10) represents the output of the first layer (I_i^2) in the Fuzzy AlexNet neural network. It states that the output of the first layer is equal to the degree of membership (oi1) calculated in Equation (9). The equation(11) represents the second-layer membership o_i^1 function. It states that the degree of membership (o_i^2) for the output of the first layer (I_i^2) is determined by applying a membership function U_A to the value I_i^2. The membership function maps the output of the first layer to a degree of membership in the range $[0, 1]$. The specific form of the membership function U_A and its parameters would be defined in the context of the Fuzzy AlexNet neural network. Here $U_A(x)$ represents the membership. The sigmoid activation function was selected for this study out of a set of membership functions since its "S" type of curve gracefully defines the node's computation complexity variations.

Hidden layer

The more hidden layers that can be equipped, the more complicated functions that can be fitted. And it may also result in the disadvantages of a large amount of computation and overfitting. One hidden layer, in particular, should be able to fulfill all of the criteria for predictive purposes. As a consequence, there is only one hidden layer in this load balancing model. The formulas for input and output are defined as follows.

$$I_i^3 = \sum_{j=1}^{n} w_i O_j^2 - \theta_j, \tag{12}$$

$$O_i^3 = \phi(I_i^3). \tag{13}$$

This equation (12) represents the weighted sum of the membership degrees of the second layer o_j^2 for each neuron j, multiplied by the corresponding weight w_i. The result is then subtracted by a threshold value θ_i. The sum is denoted as θ_i and represents the input to the third layer of the network. The equation (13) represents the membership function (o_i^3) for the third layer of the Fuzzy AlexNet neural network. It applies a membership function ϕ to the input value I_i^3 to determine the degree of membership or activation level of neuron i in the third layer. Here outputs of 10 artificial neurons on the 2nd level layer are represented by o_j^2,θ_i is a minimum value, and $\phi(x)$ is the artificial neuron activation process.

Output layer

The neural network produces fuzzy numerical values for the "high" and "low" fuzzy subsets of p_i^n The following are the formulas for input and output.

$$I_i^4 = \sum_{j=1}^{m} w_j^{'} O_j^3 - \theta_i^{'}, \tag{14}$$

$$O_i^4 = \phi(I_i^4). \tag{15}$$

The equation (14) represents the weighted sum of the membership degrees of the third layer (o_j^3) for each neuron j, multiplied by the corresponding weight $w_j^{'}$. The result is then subtracted by a threshold value $\theta_i^{'}$. The sum is denoted as o_i^4 and represents the input to the fourth layer of the network. The equation (15) represents the membership function (o_i^4) for the fourth layer of the Fuzzy AlexNet neural network. Here, m is the number of artificial neurons on the hidden layer (i.e., the third level layer),$\theta_i^{'}$ is a threshold value, and $\phi(x)$ has the same definition as equation (14).

Decode layer

In this network, using the fuzzy weighted average approach, the decoding layer is used to transform the fuzzy truth values into the correct value of p_i^n using the fuzzy weighted average approach. The following are the formulas for input and output.

$$I_i^4 = \sum_{i=2}^{5} w_i^{''} O_i^4, \tag{16}$$

$$p_i = O^5 = I^5 / \sum_{i=1}^{2} O_i^4. \tag{17}$$

This equation (16) represents the weighted sum of the membership degrees of the fourth layer o_i^4 for each neuron i, multiplied by the corresponding weight $W_i^{''}$.It represents the input to the decode layer of the network. This equation (17) calculates the normalized outputs ($p_i = 0^5$) of the decode layer in the Fuzzy AlexNet neural network.

3.4 Hybrid particle swarm optimization with ant colony (HASO) based parameter fine tuning

This article uses the HASO [27] technique to tune improved fuzzy neural network parameter values. During the proper method's initialization application, an effective technique is required for fine-tuning derived parameters and local search [10, 22]. Four different parameter groups in other domains are utilized to define the proposed structure. Each category of parameters has a parameter vector $\alpha_C(c = 1; 2; 3; 4)$ that includes all of the parameters in that category. For FNN, with R fuzzy rules, parameter vectors are specified by $\infty_1, \infty_2, \infty_3$ and ∞_4. Following are the parameters of n dimension input space structure. It includes transformation matrix parameter (T), fuzzy rule center parameter in new spaces (ψ), consequent part parameters (W), and regulator parameters (β). Despite the fact that PSO and ACO algorithms have been created for the classification task, we do not think that any of these algorithms is the best option for this issue for the reasons listed below [23, 26, 27].

A classification task requires a very long period of computing time when many attributes [29] and classes are involved. This is because every inner iteration of the typical ACO algorithm necessitates a computationally expensive rule-pruning step. Additionally, continuous characteristics must be discretized beforehand for the method to work with them, which might reduce classification accuracy and lengthen computation time at this preprocessing stage. Compared to ACO, a normal PSO algorithm does not naturally handle categorical variables. To overcome those issues this work introduces a hybrid of Particle swarm optimization with Ant Colony Optimization for parameter tuning.

Steps of HASO:

In the Hybridization of Particle swarm optimization with Ant Colony Optimization (PACO), major steps include, solution construction, local search, update of Gbest and Pbest, and update of pheromone and pheromone disturbance. In starting of depot, m ants are considered. Every ant selects solutions. For solution construction, applied ACO state transition rule. After ants complete solution construction, local search is used to optimize the top r best ants solution using Gbest and Pbest solutions of PSO. Then compare the best solutions of ACO and PSO then take the solution as best which has the higher fitness function among them. All ants update short-term memory. PACO updates pheromone trails, Gbest solutions and Pbest solutions according to elite ants and PACO performs pheromone disturbance by modifying the current pheromone trail, if there is no enhancement in Gbest in w successive iterations. PACO resets some ants Pbest solutions. On similar routes, pheromone laying can be avoided. The PSO and ant colony optimization flowchart is shown in Figure 2.

4 Results and discussion

The experimental output of the proposed model is examined in this section. The experimental setup includes Matlab 2016a software with Intel core processor. Further the training and testing phase has been conducted with 80% and 20% of samples. For the Abalone age prediction dataset (UCI Machine learning Repository), MATLAB is used to run the model in terms of F-measure, accuracy, precision, and recall values. Existing Bat Hybrid Ant Colony Particle Swarm Optimization (BA-HASO), Hybrid Ant Colony Particle Swarm Optimization (HASO), Hybridization of Ant Colony Optimization Genetic Algorithm (HACOGA) [**?**] and Intuitive, Interpretable, and Correlated-Contours fuzzy rules (IC-FNN) techniques are compared with proposed mrthodology [9]. Here CS is Easier to application, PSO produces computational efficiency and ACO has Inherent parallelism so we can get more accurate results while using these methods than other optimization models. F-ANNS technique. Ring counting count, cone cutting shell, and staining are used to define the abalone age. To predict the age of abalone, other measurements are used. To solve the problem, other information such as weather patterns and location (therefore food availability) may be needed. This work uses 10 fold cross-validation technique for the validation of results. Moreover, ANOVA statistical test is performed for this data.

Figure 3. shows the performance comparison for Convergence study of proposed ICSO and BAT optimization method for feature selection. From the Figure 3 it is concluded that this proposed ICSO has high-speed convergence and produces optimal features.

Precision performance metric comparisons between proposed F-ANNS and existing BA-HASO, HASO, HACOGA, IC-FNN are shown in Figure 4. It shows the proposed technique's effectiveness. proposed AlexNet neural networks pooling layer handles the uncertain and imprecise data by which precision is improved. On X-axis of the Figure 4, various techniques are represented and precision values are represented in Y-axis. Around 74.727% recall result value is produced in the proposed F-ANNS model, which is the highest value, whereas BA-HASO produces 58.8844%, HASO produces 56.4926%, HACOGA produces 52.3210% and IC-FNN produces 52.0303%.

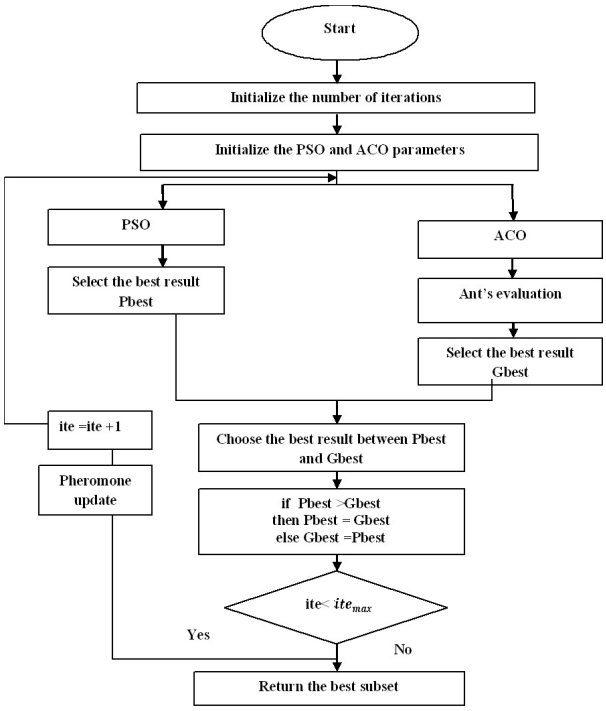


Figure 2: Flow chart of hybrid particle swarm optimization with ant colony

The performance metric comparison between proposed F-ANNS and existing BA-HASO, HASO, HACOGA, IC-FNN are shown in Figure 5. it shows the proposed techniques effectiveness. In X axis of the Figure 5, various techniques are represented and recall values are represented in Y axis. Around 97.834% recall result value is produced in proposed F-ANNS model, which is a highest value, whereas, BA-HASO produces 94.0335%, HASO produces 93.027%, HACOGA produces 89.9220 % and IC-FNN produces 80.5319%.

The accuracy performance metric comparison between proposed F-ANNS and existing BA-HASO, HASO, HACOGA, IC-FNN are shown in FigURE 6. it shows the proposed techniques effectiveness. For every feature, improved cuckoo search optimization fitness evaluation and highly relevant features are selected, so that classification accuracy is enhanced. Around 97.387% accurate results are produced in proposed F-ANNS model, which is a highest value, whereas, BA-HASO produces 97.7470%, HASO produces 93.7389%, HACOGA produces 87.4940 % and IC-FNN produces 82.8551%.

Figure 8. shows the Computation time performance metric comparison between proposed F-ANNS and existing BA-HASO, HASO, HACOGA, IC-FNN. From the results it is concluded that the proposed F-ANNS methods consume lower computation time results which is 1.25 (s) whereas, BA-HASO produces1.95(s) HASO produces 2.18(s) HACOGA

Table 2: Attribute information

Features or attributes	**Type of data**	**Measurement**	**Description**
sex	Nominal	-	Male, Female, and I (infant)
Abalone length	Cont.	millimeter	Measurement of the longest shell
Abalone diameter	Cont.	millimeter	Normal to the length
Abalone height	Cont.	millimeter	containing meat in a shell
Abalone whole weight	Cont.	gm	entire abalone
Abalone shucked weight	Cont.	gm	Abalones weight
Abalone viscera weight	Cont.	gm	gut weight (after bleeding)
Abalone shell weight	Cont.	gm	after it has been dried
Abalone rings	Int.	-	+1.5 gives the age in years

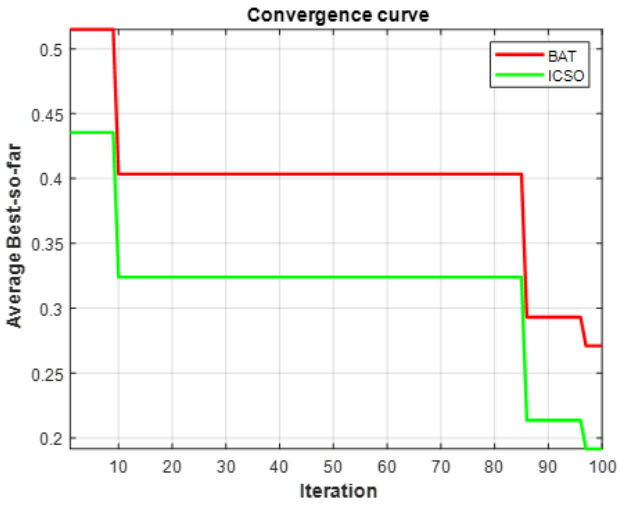


Figure 3: Convergence study for different optimization methods

Table 3: Performance Comparison Result Value

Methodologies	**Precision**	**Recall**	**Accuracy**	**F-measure**
IC-FNN	52.0309	80.5319	82.8551	63.2176
HACOGA	52.3210	89.9220	87.4940	66.9736
HASO	56.4926	93.0257	93.7389	69.3910
BA-HASO	58.8844	94.0355	94.7470	71.4910
F-ANNS	74.727	97.834	97.387	84.733

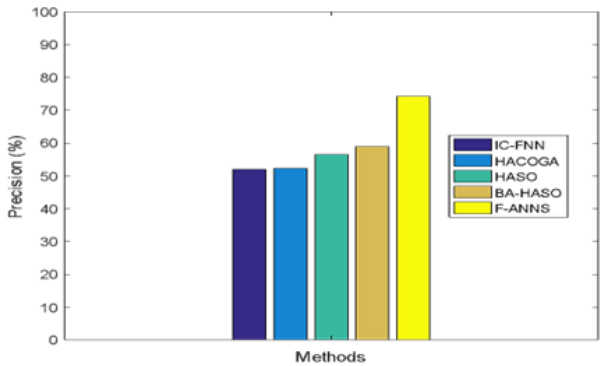


Figure 4: Precision results vs. classification methods

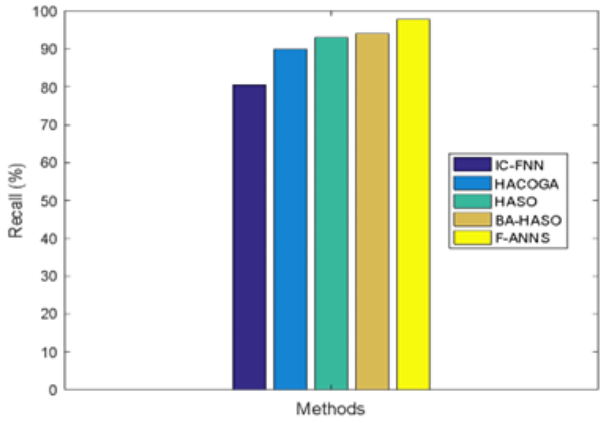


Figure 5: Recall results vs. classification methods

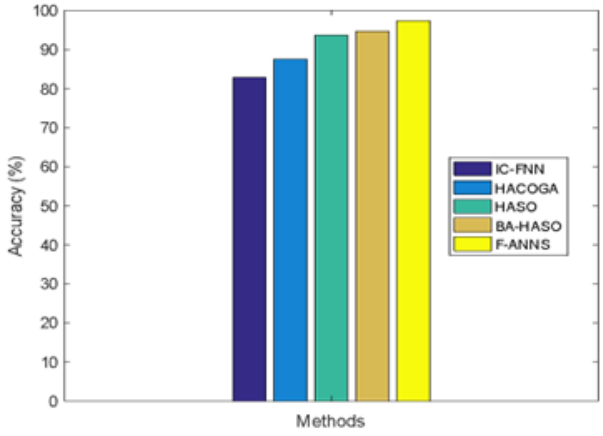


Figure 6: Accuracy results vs. classification methods

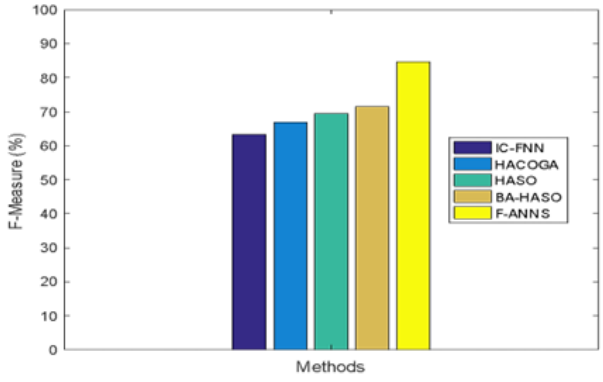


Figure 7: F-measure results vs. classification methods

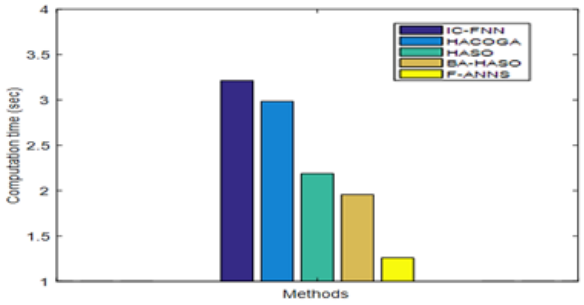


Figure 8: Computation time results vs. classification methods

2.98(s) produces and IC-FNN produces3.21(s).

From the above figures 4, 5, 6, 7, 8 it is clearly shown that the proposed F-ANNS model provides better accuracy, precision, recall, f-measure as compared to the other existing BA-HASO, HASO, HACOGA, IC-FNN methods.

5 Conclusions

The predictive capacity of the Artificial neural network training algorithm is unified using an intelligent model and the models linguistic interpretation possibility is produced through fuzzy systems, which allows the fuzzy neural network creation. For classification, an enhanced framework is introduced. In this work, data is pre-processed using a missing data interpolation technique based on clustering. Features are selected using an enhanced cuckoo search optimization technique. At last, classification is done using an enhanced classifier based on fuzzy logic and AlexNet neural network structure (F-ANNS). The hybrid Ant Colony Particle Swarm Optimization algorithm presents a promising approach for solving optimization problems. By combining the strengths of both Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), the hybrid ACPSO algorithm leverages the exploration capabilities of ACO and the exploitation abilities of PSO, leading to improved search performance and convergence speed. The revised conclusion provides a comprehensive summary of the strengths, experimental results, robustness, and scalability of the hybrid algorithm. The hybrid Ant Colony Particle Swarm Optimization (HASO) technique is used for optimizing classifier parameters in the training phase. Experimental results show that the proposed F-ANNS technique produces 97.38 % accuracy which is better than other methods. However, for the local optimal solution and the slow rate of convergence other methodologies of feature selections may be used in future work. Hybrid HASO algorithm can be further improved, refined, and expanded to address more complex optimization problems and deliver enhanced performance, scalability, and applicability across diverse domains.

Acknowledgement

The authors wish to express their appreciation for several excellent suggestions for improvements in this paper made by the referees.

References

[1] M. Abdolrazzaghi, M. H. Zarifi, W. Pedrycz, M. Daneshmand, *Robust ultra-high resolution microwave planar sensor using fuzzy neural network approach*, IEEE Sensors Journal, **17**(2) (2016), 323-332.

[2] A. S. Alturki, *Traffic sign detection and recognition using adaptive threshold segmentation with fuzzy neural network classification*, In 2018, International Symposium on Networks, Computers and Communications (ISNCC), (2018), 1-7.

[3] V. V. Borisov, K. P. Korshunova, *Multiclass classification based on the convolutional fuzzy neural networks*, In Russian Conference on Artificial Intelligence, Springer, Cham, (2019), 226-233.

[4] D. Chitara, K. R. Niazi, A. Swarnkar, N. Gupta, *Cuckoo search optimization algorithm for designing of a multimachine power system stabilizer*, IEEE Transactions on Industry Applications, **54**(4) (2018), 3056-3065.

[5] C. T. Chu, C. C. Ho, H. W. Li, *Proximity capacitive gesture recognition for an adaptive fuzzy neural network*, In 2018, 7th International Symposium on Next Generation Electronics, (ISNE), (2018), 1-4.

[6] Y. Deng, Z. Ren, Y. Kong, F. Bao, Q. Dai, *A hierarchical fused fuzzy deep neural network for data classification*, IEEE Transactions on Fuzzy Systems, **25**(4) (2016), 1006-1012.

[7] S. Ding, W. Song, D. G. Wang, H. X. Li, *A classification method by using fuzzy neural network and ensemble learning*, In International Conference on Fuzzy Information and Engineering, (2017), 133-142.

[8] L. Du, Y. Du, M. C. F. Chang, *A Reconfigurable 64-dimension K-means clustering accelerator with adaptive overflow control*, IEEE Transactions on Circuits and Systems II: Express Briefs, **67**(4) (2020), 760-764.

[9] M. M. Ebadzadeh, A. Salimi-Badr, *IC-FNN: A novel fuzzy neural network with interpretable, intuitive, and correlated-contours fuzzy rules for function approximation*, IEEE Transactions on Fuzzy Systems, **26**(3) (2018), 1288-1302.

[10] J. E. Espinosa, S. A. Velastin, J. W. Branch, *Vehicle detection using AlexNet and faster R-CNN deep learning models: A comparative study*, In International Visual Informatics Conference, (2017), 3-15.

[11] H. G. Han, Z. Y. Chen, H. X. Liu, J. F. Qiao, *A self-organizing interval type-2 fuzzy-neural-network for modeling nonlinear systems*, Neurocomputing, **290** (2018), 196-207.

[12] H. G. Han, J. M. Li, X. L. Wu, J. F. Qiao, *Cooperative strategy for constructing interval type-2 fuzzy neural network*, Neurocomputing, **365** (2019), 249-260.

[13] M. Han, K. Zhong, T. Qiu, B. Han, *Interval type-2 fuzzy neural networks for chaotic time series prediction: A concise overview*, IEEE Transactions on Cybernetics, **49**(7) (2018), 2720-2731.

[14] P. Jana, S. Ghosh, S. K. Bera, R. Sarkar, *Handwritten document image binarization: An adaptive K-means based approach*, In 2017, IEEE Calcutta Conference (CALCON), (2017), 226-230.

[15] S. D. Kermany, M. Joorabian, S. Deilami, M. A. Masoum, *Hybrid islanding detection in microgrid with multiple connection points to smart grids using fuzzy-neural network*, IEEE Transactions on Power Systems, **32**(4) (2016), 2640-2651.

[16] I. Khan, Z. Luo, J. Z. Huang, W. Shahzad, *Variable weighting in fuzzy k-means clustering to determine the number of clusters*, IEEE Transactions on Knowledge and Data Engineering, **32**(9) (2020), 1838-1853.

[17] G. Khodabandelou, M. M. Ebadzadeh, *Fuzzy neural network with support vector-based learning for classification and regression*, Soft Computing, (2019), 12153-12168.

[18] K. P. Korshunova, *A convolutional fuzzy neural network for image classification*, In 2018 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC), (2018), 1-4.

[19] S. Z. Mirjalili, H. Zhang, S. Chalup, N. Noman, S. Mirjalili, *A novel U-shaped transfer function for binary particle swarm optimisation*, Soft Computing for Problem Solving, (2019), 241-259. DOI:10.1007/978-981-15-3290-0_19.

[20] T. K. B. Mudiyanselage, X. Xiao, Y. Zhang, Y. Pan, *Deep fuzzy neural networks for biomarker selection for accurate cancer detection*, IEEE Transactions on Fuzzy Systems, **28**(12) (2019), 3219-3228.

[21] W. Nawaz, S. Ahmed, A. Tahir, H. A. Khan, *Classification of breast cancer histology images using AlexNet*, In International Conference Image Analysis and Recognition, (2018), 869-876.

[22] P. Patro, K. Kumar, G. S. Kumar, *Cellular neural network, fuzzy cellular neural network and its applications*, International Journal of Control Theory and Applications, **10** (2017), 161-168.

[23] P. Patro, K. Kumar, G. S. Kumar, *Applications of three layer CNN in image processing*, Journal of Advanced Research in Dynamical and Control Systems, **01** (2018), 510-512.

[24] P. Patro, K. Kumar, G. S. Kumar, *Various classifiers performance based machine learning methods*, International Journal of Recent Technology and Engineering (IJRTE), **8**(3S3) (2019), 305-310.

[25] P. Patro, K. Kumar, G. S. Kumar, *Neuro fuzzy system with hybrid ant colony particle swarm optimization (HASO) and robust activation*, Journal of Advanced Research in Dynamical and Control Systems, **02** (2020), 741-750.

[26] P. Patro, K. Kumar, G. S. Kumar, *Optimized hybridization of ant colony optimization and genetic algorithm (HACOGA) based IC-FNN classifier for abalone*, Journal of Computational and Theoretical Nanoscience, **17** (2020), 2756-2764.

[27] N. M. Ranjan, R. S. Prasad, *LFNN: Lion fuzzy neural network-based evolutionary model for text classification using context and sense-based features*, Applied Soft Computing, **71** (2018), 994-1008.

[28] UCI Machine learning repository, https://archive.ics.uci.edu/ml/index.php.

[29] J. Xu, J. Han, K. Xiong, F. Nie, *Robust and sparse fuzzy K-means clustering*, Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-16, (2016), 2224-2230.

[30] R. Zhang, X. Li, H. Zhang, F. Nie, *Deep fuzzy K-means with adaptive loss and entropy regularization*, IEEE Transactions on Fuzzy Systems, **28**(11) (2020), 2814-2824.

[31] J. Zhao, S. Liu, M. Zhou, X. Guo, L. Qi, *Modified cuckoo search algorithm to solve economic power dispatch optimization problems*, IEEE/CAA Journal of Automatica Sinica, **5**(4) (2018), 794-806.

[32] G. Zhou, L. Lv, X. Qiao, L. Jin, *Hierarchical attention-based fuzzy neural network for subject classification of power customer service work orders*, In 2019, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), (2019), 1-6.

[33] S. Zobeidi, M. Naderan, S. E. Alavi, *Effective text classification using multi-level fuzzy neural network*, In 2017, 5th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), (2017), 91-96.