



Extreme Learning Machine Based Pattern Classifiers for Symbolic Interval Data

N. Emami*^a, M. Kuchaki Rafsanjani^b

^aDepartment of Computer Science, Faculty of Engineering and Basic Sciences, Kosar University of Bojnord, Iran

^bDepartment of Computer Science, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran

PAPER INFO

Paper history:

Received 28 April 2021

Received in revised form 11 September 2021

Accepted 28 September 2021

Keywords:

Interval Data

Classification

Extreme Learning Machine

Data Analysis

ABSTRACT

Interval data are usually applied where inaccuracy and variability must be considered. This paper presents a learning method for Interval Extreme Learning Machine (IELM) in classification. IELM has two steps similar to well known ELM. At first weights connecting the input and the hidden layers are generated randomly and in the second step, ELM uses the Moore–Penrose generalized inverse to determine the weights connecting the hidden and output layers. In order to use Moore–Penrose generalized inverse for determining second layer weights in IELM, this paper proposes four classification methods to handle symbolic interval data based on ELM. The first one uses a midpoint of intervals for each feature value then it applies a classic ELM. The second one considers each feature value as a pair of quantitative features and implements a conjoint for classic extreme learning machine. The third one represents interval features by their vertices and performs a classic extreme learning machine as well. The fourth one takes each interval as a pair of quantitative features after that two separated classic extreme learning machines are performed on these features and combines the results accordingly. Algorithms are tested on the synthetic and real datasets. A synthetic dataset is applied to determine the number of hidden layer nodes in an IELM. The classification error rate is considered as a comparison criterion. The error rate obtained for each proposed methods is 19.167%, 15% , 6.536% and 18.333% respectively. Experiments demonstrate the usefulness of these classifiers to classify symbolic interval data.

doi: 10.5829/ije.2021.34.11b.17

NOMENCLATURE

a^L	Lower limit of interval	L	Hidden nodes
a^U	Upper limit of interval	w_i	First layer weight vector
a^M	Midpoint of interval	β_i	Second layer weight vector
k	Positive scalar	b_i	Bias
K	Number of classes	G(.)	Activation function
F	Increasing function	H	Hidden layer output matrix of the neural network
i/j	Index	$\hat{\beta}$	β approximation
N	Number of sample	H^+	Moore-Penrose inverse of matrix H.
x_i	The i^{th} sample	f_{ij}^L	The lower limit of the j^{th} feature of the i^{th} sample
t_j	The j^{th} target	f_{ij}^U	The upper limit of the j^{th} feature of the i^{th} sample
o_j	The j^{th} output	f_{ij}^M	The midpoint of the j^{th} feature of the i^{th} sample
p	Number of features	γ_i	Length of interval

1. INTRODUCTION

In real-life situations, there is imprecise and incompleteness in the feature values [1-6]. It is suitable

to apply interval feature value for data [7-11]. Interval data offer a way of representing the available information where uncertainty or variability must be taken into account [12]. Analyzing and modeling for interval data have raised in the field of Symbolic Data Analysis (SDA) [13]. It is introduced as a new domain in multivariate analysis, pattern recognition and

*Corresponding Author Institutional Email: nasibeh.emami@kub.ac.ir
(N. Emami)

artificial intelligence scope. SDA aims to provide suitable methods (clustering, factorial techniques, decision trees, etc.) for managing aggregated data described by multi-valued variables, where the cells of the data table contain the sets of categories, intervals or weight (probability) distributions [14,15].

Several clustering methods have been proposed for interval data. A fuzzy clustering method is used for analyzing interval-valued data which is introduced by D'Urso et al. [16]. A preferential interval-valued fuzzy c-means algorithm for remotely sensed imagery classification is presented by Feng et al. [17]. A method for dealing with hierarchical clustering for interval-valued data has been proposed by Galdino and Maciel [18]. A new interval possibilistic fuzzy c-means (IPFCM) clustering method is proposed for clustering symbolic interval data [19]. A multivariate outlier detection method for interval data is proposed by Silva et al. [20] that makes use of a parametric approach to model the interval data. A simulation study demonstrates the usefulness of the robust estimates for outlier detection, and new diagnostic plots allow gaining deeper insight into the structure of real world interval data. A robust partitioning fuzzy clustering algorithm for interval-valued data based on adaptive city-block distance that takes into account the relevance of the variables according to the boundaries is proposed [21]. This distance changes at each iteration of the algorithm and is different from one cluster to another. The method optimizes an objective function by alternating three steps to compute the representatives of each group, the fuzzy partition, and the relevance weights for the interval-valued variables for each boundary is investigated.

Several supervised classification methods are directed toward developing efficient tools related to interval data. A symbolic classifier as a region-oriented approach for the quantitative, categorical, interval, and categorical multi-valued data was introduced by Rizo Rodríguez and de Assis Tenório de Carvalho [22]. This approach is an adaptation of the concept of mutual neighbors to define the concepts of mutual neighbors between symbolic data and Mutual Neighbourhood Graph (MNG) between groups. At the end of the learning step, the symbolic description of each group is obtained through the use of an approximation of a MNG and a symbolic join operator. In order to reduce the complexity of the learning step without compromising the classifier performance with regarding to the prediction accuracy another MNG approximation is proposed [23]. A region-oriented approach in which each region is determined by the convex hull of the objects belonging to a class was introduced by D'Oliveira et al. [24]. A generalization of binary decision trees to predict the class membership of symbolic data is presented in literature [25]. A novel

approach by Singh and Huang [26], in order to solve the problems of classification and decision-making by employing the interval-valued fuzzy sets, rough sets and granular computing (GrC) concepts. A novel approach which is introduced in literature [27] is a generalization of probabilistic neural network for interval data processing that can be used in classifying interval information. Generalized multi-perceptions to work with interval data are mentioned in literature [28]. The fuzzy radial basis function network to work with symbolic data was introduced by mali and Mitra [29]. A lazy-learning approach that extends K-Nearest Neighbor classification to modal and interval data is stated in literature [30]. A new model from multilayer perceptron based on interval arithmetic where inputs and outputs are considered as interval values but weights and biases are considered as single-values introduced in literature [12]. Different pattern classifiers for interval data based on the logistic regression methodology have been presented by De Souza et al. [31].

Lately, the Extreme Learning Machine (ELM) has been proposed by Huang [32-34]. It is derived from the single-hidden layer feed forward neural networks (SLFNs) and has an input and hidden layer. There are two steps for computing weights in ELM. In the first step, the weights are generated randomly between the input and hidden layer. The second step applies the Moore-Penrose generalized inverse to specify the weights connecting between the hidden and output layer.

We proposed four ELM based pattern classifiers for symbolic interval data. Inputs are vectors with interval components and output are crisp. Also, the weights and biases are real numbers. The first one uses the midpoint of intervals for each feature and uses a classic ELM. The second one considers each feature values as a pair of quantitative features and applies a conjoint classic ELM. The third one is represented by its vertices and performs classic ELM. The fourth one takes each interval as a pair of quantitative features and then it performs two separate classic ELM on these features and combines the results in suitable way. The main contributions of this paper are:

- Classification of aggregated data described by multi-valued features.
- Providing some ways in which interval data can be compatible for Moore-Penrose inverse calculation in the second layer of ELM.
- Reducing the classification error rate.

The rest of the paper is organized as follows: section 2 points out some important preliminaries, section 3 introduces proposed pattern classifiers for interval data based on ELM. The performance of these classifiers is based on the prediction error rate Experimental data and results have been presented in section 4 and the performance results on the car interval dataset are

shown. Discussion over the proposed methods can be found in section 5. Finally, section 6 concludes the paper.

2. PRELIMINARIES

This section describes preliminaries about interval arithmetic and original ELM.

2.1. Interval Arithmetic Interval arithmetic was early introduced as a technique for considering uncertainty, inaccuracy or variability. It works by expressing every uncertainty quantity as a range of possible values. The size of this range (or interval) expresses the uncertainty associated with the quantity [7, 35, 36]. An interval number A is a closed interval $[a^L, a^U] \subset \mathbb{R}$ of all real numbers that including the end points a^L and a^U , such that $a^L \ll a^U$. If $a^L = a^U$ then interval is called to be degenerated, thin or even point interval. Let interval number $A = [a^L, a^U]$ and $B = [b^L, b^U] \subset \mathbb{R}$. Intervals are produced for each arithmetic operation in Equations (1)-(6).

$$[a^L, a^U] + [b^L, b^U] = [a^L + b^L, a^U + b^U] \quad (1)$$

$$[a^L, a^U] - [b^L, b^U] = [a^L - b^L, a^U - b^U] \quad (2)$$

$$[a^L, a^U] \times [b^L, b^U] = [\min(a^L b^L, a^L b^U, a^U b^L, a^U b^U), \max(a^L b^L, a^L b^U, a^U b^L, a^U b^U)] \quad (3)$$

$$k \times [a^L, a^U] = [ka^L, ka^U] \quad (4)$$

$$F([a^L, a^U]) = [F(a^L), F(a^U)] \quad (5)$$

$$a^M = \frac{a^L + a^U}{2} \quad (6)$$

2.2. Extreme Learning Machine (ELM) The mathematical modeling of the ELM describes here [33]. Consider N arbitrary samples (x_i, t_i) , where $x_i = [x_{i1}, x_{i2}, \dots, x_{ip}]^T \in \mathbb{R}^p$ and $t_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in \mathbb{R}^m$, a standard Single Layer Feed Forward Neural Network (SLFN) with L hidden nodes can approximate these N samples with zero error means $\sum_{j=1}^L \|o_j - t_j\| = 0$; i.e., there exist (w_i, b_i) and β_i such that

$$\sum_{i=1}^L \beta_i G(w_i, b_i, x_j) = t_j, \quad j = 1, \dots, N \quad (7)$$

Equation (7) can be written in compact form as $H\beta = T$ where

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} G(w_1, b_1, x_1) & \dots & G(w_L, b_L, x_1) \\ \vdots & \dots & \vdots \\ G(w_1, b_1, x_N) & \dots & G(w_L, b_L, x_N) \end{bmatrix}_{N \times L}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \quad \text{and} \quad T = \begin{bmatrix} T_1^T \\ \vdots \\ T_N^T \end{bmatrix}_{N \times m}$$

The i^{th} column of H is the i^{th} hidden node output with respect to inputs x_1, x_2, \dots, x_N .

Figure 1 shows the overall scheme of ELM. weights are selected randomly in the input layer but weights in the hidden layer need to be adjusted based on the training samples. Therefore, we have the linear system $H\beta = T$ to find a least squares solution $\hat{\beta}$ such that

$$\|H\hat{\beta} - T\| = \min_{\hat{\beta}} \|H\hat{\beta} - T\| \quad (8)$$

The smallest norm least square solution of Equation (8) is $\hat{\beta} = H^+T$.

3. PROPOSED ELM CLASSIFIERS FOR INTERVAL DATA

In this section, MELM, JELM, VELM and LUEL pattern classifiers based on ELM for interval data are presented. In all these methods, the weights are real. The first layer weights are selected randomly and the second layer weights are learned by training data. Afterwards, the learned weights are used to assign new samples to the classes.

Suppose $(x_i, t_i), i = 1, \dots, N$, be a training symbolic sample set with K class labels. Sample i^{th} presents by interval features $\{f_{i1}, f_{i2}, \dots, f_{ip}\}$ which $f_{ij} = [f_{ij}^L, f_{ij}^U], j = 1, 2, \dots, p$ and a categorical discrete variable $t_i \in \{1, 2, \dots, K\}$.

Let p, \tilde{N} and m to be the number of neurons for the input, hidden and output layers, respectively. The weight vector connecting the input and the i^{th} hidden layers is denoted by $w_i = [w_{i1}, w_{i2}, \dots, w_{ip}]^T \in \mathbb{R}^p$ and $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T \in \mathbb{R}^m, i = 1, 2, \dots, \tilde{N}$ and b_i denotes the threshold of the i^{th} hidden node for $i = 1, 2, \dots, \tilde{N}$. An activation function $g(\cdot)$ is used for the hidden and output layers.

3.1. MELM This method uses the midpoint of the intervals in the representation of interval data. That means a feature $f_{ij} = [f_{ij}^L, f_{ij}^U]$ is represented by $f_{ij}^M = \frac{f_{ij}^L + f_{ij}^U}{2}$. Therefore each symbolic interval training sample i has a vector of p features midpoint $x_i^M = [f_{i1}^M, f_{i2}^M, \dots, f_{ip}^M]$ that are fed as an input to the network. Then weights are selected randomly in the input layer and the second layer weights are earned by solving the linear system $H\beta = T$. The smallest norm least square solution of the linear system is $\hat{\beta} = H^+T$. So $\hat{\beta}$ coefficient learned by training samples is being applied in the classification of interval data. Algorithm1 summerises the proposed MELM.

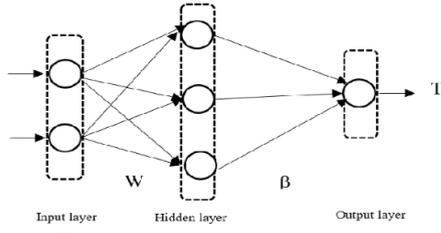


Figure 1. Overall Scheme of ELM

3. 2. JELM Here, pattern classifier is introduced which utilises the lower and upper bounds of the intervals conjointly. Sample interval training data are $x_i = [[f_{i1}^L, f_{i1}^U], [f_{i2}^L, f_{i2}^U], \dots, [f_{ip}^L, f_{ip}^U]]$. In order to consider the lower and upper bounds of the intervals conjointly, each sample has been represented by $2p$ feature values $x_i = [f_{i1}^L, f_{i1}^U, f_{i2}^L, f_{i2}^U, \dots, f_{ip}^L, f_{ip}^U]$. These vectors are being fed as inputs to the ELM.

The first layer weights are being produced randomly and for the second layer we need to solve the linear system $H\beta = T$. The smallest norm least square solution of the linear system comes from training samples and we have equation $\tilde{\beta} = H^+T$. After we find the approximate weights by training data, classification can be done. Algorithm 2 summarises the proposed JELM.

3. 3. VELM This subsection introduces a method based on ELM by employing the vertices of the hypercube for symbolic interval data. Suppose a symbolic interval vector is shown by $x_i = [[f_{i1}^L, f_{i1}^U], \dots, [f_{ip}^L, f_{ip}^U]]$ that have 2^p vertices in R^p space. It can be described by a matrix

$$M = \begin{pmatrix} f_{i1}^L & \dots & f_{ip}^L \\ f_{i1}^U & \dots & f_{ip}^U \\ \vdots & \ddots & \vdots \\ f_{i1}^U & \dots & f_{ip}^L \\ f_{i1}^L & \dots & f_{ip}^U \end{pmatrix}$$

of all the vertices of hypercube in R^p

space. Therefore each symbolic interval training sample is a matrix $2^p \times p$ corresponds to all possible combinations of the limits of intervals. Class variable for each row of matrix M is similar to the original representation of the samples. If training sample dataset have N samples then the size of training sample becomes $N \times 2^p$ rows and p column in this representation.

For instance, let $N=2, p=2$ and suppose the sample $x_i = ([f_{i1}^L, f_{i1}^U], [f_{i2}^L, f_{i2}^U])$. matrix of vertices of the hypercubes for this sample is:

$$M = \begin{pmatrix} f_{i1}^L & f_{i2}^L \\ f_{i1}^L & f_{i2}^U \\ f_{i1}^U & f_{i2}^L \\ f_{i1}^U & f_{i2}^U \end{pmatrix}$$

and the symbolic interval training dataset

$$M = \begin{pmatrix} f_{11}^L & f_{12}^L & y_1 \\ f_{11}^L & f_{12}^U & y_1 \\ f_{11}^U & f_{12}^L & y_1 \\ f_{11}^U & f_{12}^U & y_1 \\ f_{21}^L & f_{22}^L & y_2 \\ f_{21}^L & f_{22}^U & y_2 \\ f_{21}^U & f_{22}^L & y_2 \\ f_{21}^U & f_{22}^U & y_2 \end{pmatrix}$$

changed to a new dataset M . By constructing the M -matrix, the ELM inputs are prepared for the symbolic interval training data, and the weights of the first layer are randomly selected. Then the weights of the second layer are approximated by calculating the Moore-Penrose generalized inverse of H -matrix. Algorithm 3 summarises the proposed VELM.

3. 4. LUELM A pattern classifier for symbolic interval data based on ELM is defined by the lower and upper bounds of the interval separately. In this method, we will perform one ELM based on lower bounds and another ELM is based on the upper bounds. In each ELM, first layer weights are produced randomly and second layer weights are calculated by solving the linear system $H\beta = T$. So an approximation of β coefficient is $\tilde{\beta} = H^+T$.

In order to label the samples, the weighted average output of the networks is calculated and passed through the discretized function. Algorithm 4 summarises the proposed LUELM.

ALGORITHM 1: Pseudocode of MELM

MELM algorithm
Input: Given interval training set $x_i = ([f_{i1}^L, f_{i1}^U], \dots, [f_{ip}^L, f_{ip}^U]), T_i, i = 1, \dots, N$
Output: class label
1. Calculate the midpoint of intervals $x_i^M = [f_{i1}^M, \dots, f_{ip}^M] \in R^p$
2. Randomly assign the input weights w_i and bias b_i
3. Calculate the hidden layer output matrix H
4. Calculate the output weights matrix as $\tilde{\beta} = H^+T$
5. Calculate the class label based on $\tilde{\beta}$ coefficient

ALGORITHM 2: Pseudocode of JELM

JELM algorithm
Input: Given interval training set. $x_i = ([f_{i1}^L, f_{i1}^U], \dots, [f_{ip}^L, f_{ip}^U]), T_i, i = 1, \dots, N$
Output: class label
1. Consider lower and upper bounds of the intervals conjointly $x_i = [f_{i1}^L, f_{i1}^U, f_{i2}^L, f_{i2}^U, \dots, f_{ip}^L, f_{ip}^U] \in R^{2p}, i = 1, \dots, N$
2. Randomly assign the input weights w_i and bias b_i
3. Calculate the hidden layer output matrix H
4. Calculate the output weights matrix as $\tilde{\beta} = H^+T$
5. Calculate the class label based on $\tilde{\beta}$ coefficient

ALGORITHM 3: Pseudocode of VELM

VELM algorithm
Input: Given interval training set $x_i = ([f_{i1}^L, f_{i1}^U], \dots, [f_{ip}^L, f_{ip}^U]), T_i, i = 1, \dots, N$
Output: class label
1. Represent interval data by vertices of the hypercube $x_i \in R^p, i = 1, \dots, N \times 2^p$
2. Randomly assign the input weights w_i and bias b_i
3. Calculate the hidden layer output matrix H

-
4. Calculate the output weights matrix as $\hat{\beta} = H^+T$
 5. Calculate the class label based on $\hat{\beta}$ coefficient
-

4. EXPERIMENTAL AND RESULTS

In this section, the proposed methods for interval extreme learning machines have been applied for three examples. Example one is a synthetic interval data with a low degree of class overlapping. Example two is a synthetic interval data too; however, it has a moderate degree of class overlapping. These synthetic data sets are being used to determine the number of layers for each model. Example three is a real dataset.

Each synthetic interval data has three classes. One of the classes has an ellipse shape of size 200 and other classes have spherical shapes of size 150 and 100. Each class in these quantitative datasets was drawn according to two independent normal distributions. Each data point (z_1, z_2) of each one of this synthetic quantitative dataset is a seed of a vector of intervals (rectangle) defined as $([z_1 - \gamma_1/2, z_1 + \gamma_1/2], [z_2 - \gamma_2/2, z_2 + \gamma_2/2])$.

Example one is constructed according to the following parameters:

- class1: $\mu_1 = 50, \mu_2 = 25, \delta_1^2 = 9$ and $\delta_2^2 = 36$
- class2: $\mu_1 = 45, \mu_2 = -2, \delta_1^2 = 25$ and $\delta_2^2 = 25$
- class3: $\mu_1 = 38, \mu_2 = 40, \delta_1^2 = 9$ and $\delta_2^2 = 9$

It has low degree of class overlapping. Figures 2 and 3 display the quantitative and interval form of data in example one respectively.

Example two has moderate degree of class overlapping and it is constructed according to the following parameters:

- class1: $\mu_1 = 50, \mu_2 = 25, \delta_1^2 = 9$ and $\delta_2^2 = 36$
- class2: $\mu_1 = 45, \mu_2 = 5, \delta_1^2 = 25$ and $\delta_2^2 = 25$
- class3: $\mu_1 = 45, \mu_2 = 40, \delta_1^2 = 9$ and $\delta_2^2 = 9$

Figures 4 and 5 show the quantitative and interval form of data in example two, respectively [22].

Car dataset is a real symbolic interval dataset. It is widely used to compare classification methods of the literature of SDA [14, 15, 31, 37-39]. It contains a total of 33 car models described by eight interval features included price, engine capacity, top speed, acceleration, step, length, width and height. The nominal variable of Car category places each car in two classes. Class

ALGORITHM 4: Pseudocode of LUELML

LUELML algorithm

Input: Given interval training set

$$x_i = ([f_{i1}^l, f_{i1}^u], \dots, [f_{ip}^l, f_{ip}^u]), T_i, i = 1, \dots, N$$

Output: class label

Step one:

1. Consider the lower bounds of the interval $x_i = [f_{i1}^l, f_{i2}^l, \dots, f_{ip}^l] \in R^p, i = 1, \dots, N$
 2. Randomly assign the input weights w_i and bias b_i
-

-
3. Calculate the hidden layer output matrix H
 4. Calculate the output weights matrix as $\hat{\beta} = H^+T$
 - 5.

Step two:

6. Consider the upper bounds of the interval $x_i = [f_{i1}^u, f_{i2}^u, \dots, f_{ip}^u] \in R^p, i = 1, \dots, N$
7. Randomly assign the input weights w_i and bias b_i
8. Calculate the hidden layer output matrix H
9. Calculate the output weights matrix as $\hat{\beta} = H^+T$

Step three:

10. Calculate the average of step one and two
 11. Calculate the class label based on average $\hat{\beta}$ coefficient
-

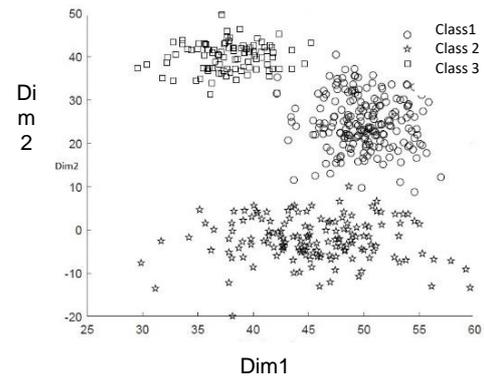


Figure 2. The quantitative form of data in example 1

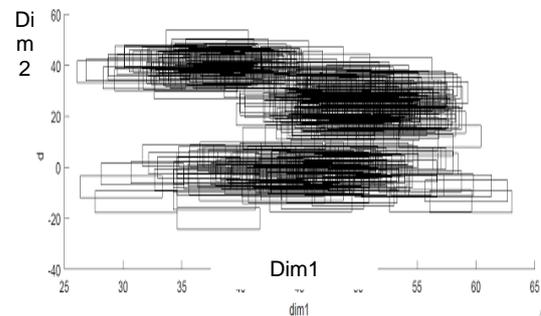


Figure 1. Interval form of data in example 1

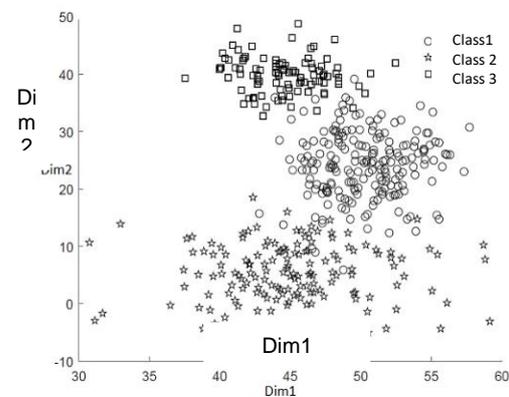


Figure 4. The quantitative form of data in example 2

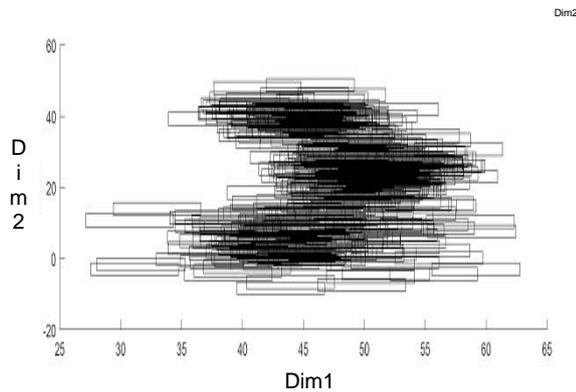


Figure 5. Interval form of data in example 2

one (Utilitarian and Berlina) has 18 car models and class two (Sporting and Luxury) has 13 car models. (see Table 1).

In order to evaluate the performance of the proposed classifiers, 75% of the original dataset is selected randomly as learning set and 25% of the original dataset is selected as test dataset.

The MELM, JELM, VELM and LUEL M classifiers were applied to the synthetic interval example datasets one and two. The error rate of the classification is computed on the test data. The estimated error rate of classification corresponds to the average of the error rates is found among the 100 replications of the test set for each synthetic interval example datasets one and two.

The average classification error rate for each proposed classifier on synthetic interval example datasets one and two are shown in Tables 2 and 3 respectively. The standard deviation of the results is enclosed in parentheses. γ_1 and γ_2 are selected randomly from [1,10], [1,20], [1,30], [1,40], [1, 50] and the number of nodes in hidden layer (M) for each classifier is placed in a separate column. Interval data in this configuration shows lower degree of classification difficulty. Therefore, the best average rate of the JELM is slightly affected by the higher range of intervals.

TABLE 1. Car dataset feature and class values

	Price	Engine Capacity	Height	Category
Alfa 145	[27806, 33596]	[1370, 1910]	[143, 143]	Utilitarian
Alfa 156	[41593, 62291]	[1598, 2492]	[142, 142]	Berlina
Alfa 166	[64499, 88760]	[1970, 2959]	[142, 142]	Luxury
Aston Martin	[260500, 460000]	[5935, 5935]	[124, 132]	Sporting
.
.
.
passat	[39676, 63455]	[1595, 2496]	[146, 146]	Luxury

TABLE 2. Average classification error rate on synthetic interval examples one

Proposed methods	γ	Synthetic example one	
		M=5	M=7
MELM	[1,10]	1.67(1.09)	12.92(27.69)
	[1,20]	1.82(1.32)	18.84(33.51)
	[1,30]	1.40(0.96)	16.34(29.05)
	[1,40]	1.11(0.80)	19.08(33.25)
	[1,50]	1.49(3.57)	18.10(28.15)
JELM	[1,10]	1.04(1.02)	0.48(0.69)
	[1,20]	1.85(1.78)	0.71(0.86)
	[1,30]	1.90(1.46)	0.48(0.80)
	[1,40]	1.61(1.88)	0.74(0.85)
	[1,50]	1.07(0.95)	0.71(0.83)
VELM	[1,10]	37.21(1.89)	34.79(1.99)
	[1,20]	39.72(3.33)	37.04(2.76)
	[1,30]	40.64(4.77)	32.30(2.29)
	[1,40]	42.30(2.85)	34.12(2.60)
	[1,50]	40.22(4.86)	37.96(4.76)
LUEL M	[1,10]	2.5(1.18)	4.43(18.11)
	[1,20]	1.04(1.15)	1.16(1.10)
	[1,30]	5.18(13.46)	5.71(14.09)
	[1,40]	2.86(1.43)	7.05(14.41)
	[1,50]	3.27(1.58)	10.06(18.44)

TABLE 3. Average classification error rate on synthetic interval examples two

Proposed methods	γ	Synthetic example two	
		M=5	M=7
MELM	[1,10]	8.57(2.63)	9.11(2.94)
	[1,20]	8.78(2.03)	20.68(26.00)
	[1,30]	8.93(2.30)	26.93(30.14)
	[1,40]	8.63(2.20)	26.96(28.72)
	[1,50]	9.11(2.94)	31.40(31.97)
JELM	[1,10]	8.63(2.89)	4.94(2.63)
	[1,20]	7.59(3.14)	4.32(2.10)
	[1,30]	8.96(4.15)	4.91(3.15)
	[1,40]	8.75(3.22)	6.70(2.46)
	[1,50]	8.39(3.44)	7.05(2.94)
VELM	[1,10]	37.70(3.46)	33.91(1.81)
	[1,20]	38.98(2.49)	34.07(2.23)
	[1,30]	36.09(2.06)	35.44(2.93)
	[1,40]	38.88(2.46)	35.81(4.00)
	[1,50]	38.03(4.02)	37.21(4.31)

	[1,10]	7.92(2.57)	16.67(27.93)
	[1,20]	7.20(2.39)	15.65(22.18)
LUELM	[1,30]	8.81(7.41)	14.88(18.10)
	[1,40]	7.20(4.10)	26.49(30.65)
	[1,50]	12.42(21.88)	22.56(25.48)

The worst average performance is obtained by the VELM that uses vertex to represent data. The size of data increases in VELM, therefore it is expected that as the size of data increments, the number of hidden nodes also becomes large. As a result the classification error rate in VELM with 7 nodes in hidden layer for synthetic example one and two is better than VELM with 5 nodes in hidden layer for synthetic example one and two.

Tables 2 and 3 show that MELM and LUELm with 5 nodes in the hidden layer produce better results than MELM and LUELm with 7 nodes in the hidden layer. JELM and VELM prefer 7 nodes in the hidden layer to lead to significant results. However, some high dimensional dot product operations appear in the training process. Eventually, it causes increasing of the computational complexity and training time.

In order to better demonstrate and compare the classification error rates of the proposed methods, the results are shown in Figures 6 to 13. The horizontal axis shows degree of classification difficulty and the vertical axis shows classification error rate on the proposed methods.

Figure 6 demonstrates MELM classification error rate on synthetic example one. It shows that as the degree of classification difficulty increases the results do not change significantly in MELM with 5 hidden nodes but, with the surge of neurons in the hidden layer, the classification error has an almost upward trend. Classification error rate on synthetic example one obtained from JELM is shown in Figure 7. Trends on Figure 7 illustrates that increasing the number of hidden layer neurons in JELM results in decreasing the average classification error rate. The trend of the average error rate based on the degree of classification difficulty is almost constant. Trends on VELM classification error rate on synthetic example one is demonstrated in Figure 8. In VELM, the number of data is increased, so when the number of hidden layer neurons increases, better results are obtained. The trend which is based on degree of classification difficulty is almost constant in this case.

Figure 9 illustrates the results on LUELm method. It shows that by increasing the number of hidden layer neurons, the classification error is increased and the degree of classification difficulty is related to an increasing trend.

Classification error rate diagram on synthetic example two is shown in Figure 10. Trends on this

diagram show that the increasing in the degree of classification difficulty has no effect on the classification error rate of the MELM with 5 hidden nodes but it has an increasing effect on the classification error rate of the MELM with 7 hidden nodes. Figure 11 illustrates JELM classification error rate in synthetic example two. This proposed method tends to have fewer neurons in the hidden layer, and the degree of classification difficulty of the data does not have much effects on the classification accuracy.

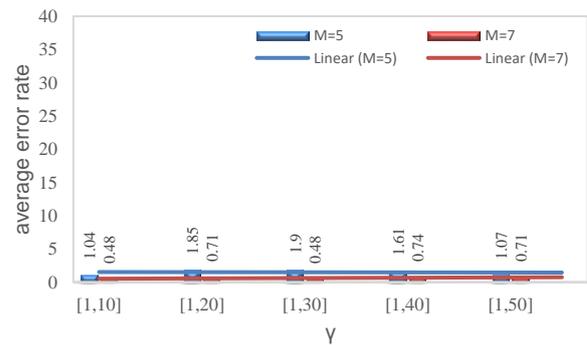


Figure 7. JELM classification error rate in synthetic example one

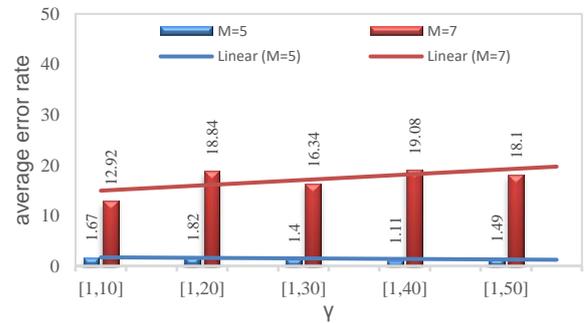


Figure 6. MELM classification error rate in synthetic example one

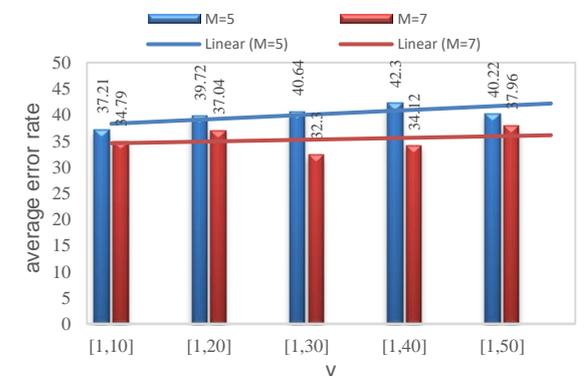


Figure 8. VELM classification error rate in synthetic example one

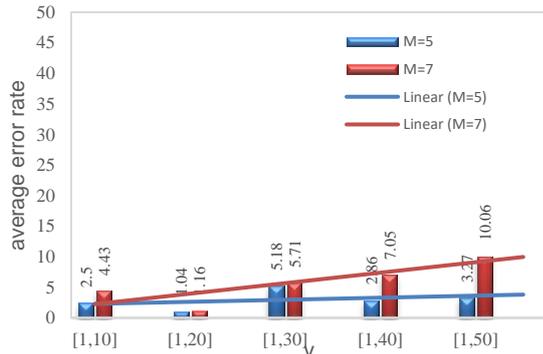


Figure 9. LUELM classification error rate in synthetic example one

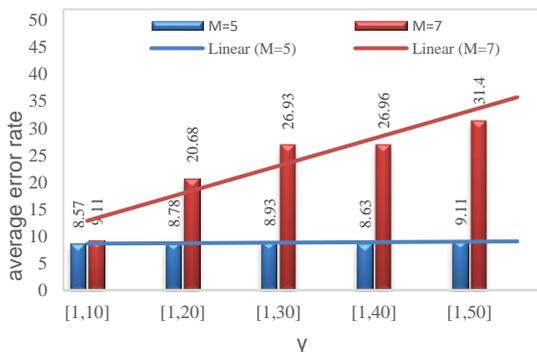


Figure 10. MELM classification error rate on synthetic example two

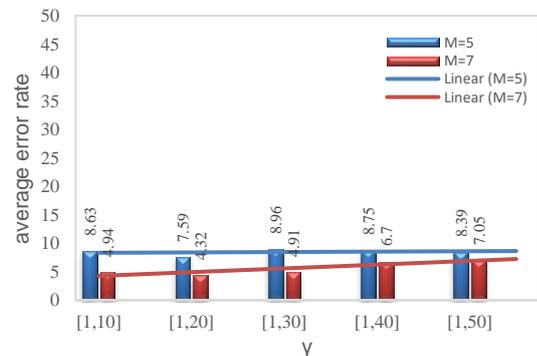


Figure 11. JELM classification error rate in synthetic example two

VELM classification error rate in synthetic example two is shown in Figure 12. Unlike other proposed algorithms in this paper, VELM offers interesting results. Here the best error rate is achieved for more neurons in hidden layer. As expected, with increasing the number of the data, the number of the hidden layer neurons increases. Also, the degree of classification difficulty of the data does not have significant effect on

the VELM, but when VELM has a smaller number of neurons, the error increases along with increasing the degree of classification difficulty of the data.

Figure 13 illustrates LUELM classification error rate on synthetic example two. LUELM with 5 hidden neurons has significant error rate and it is as the degree of classification difficulty of the data. There is a slight increase in the average error rate, as the number of neurons increases, On the other hand, the classification error increases as the number of neurons increases.

The proposed methods are also tested on the a real Car dataset as an application. Data is divided into training and test data randomly such that 75% of the original dataset has been selected as the learning set and 25% of the original dataset has been selected as the test dataset Table 4 shows the results of the average error rate among the 100 replications of the proposed methods for the Car dataset. In real car dataset, MELM and LUELM with 5 neurons in the hidden layer demonstrate less error rate than MELM and LUELM with 7 neurons. Also the JELM and VELM with 7

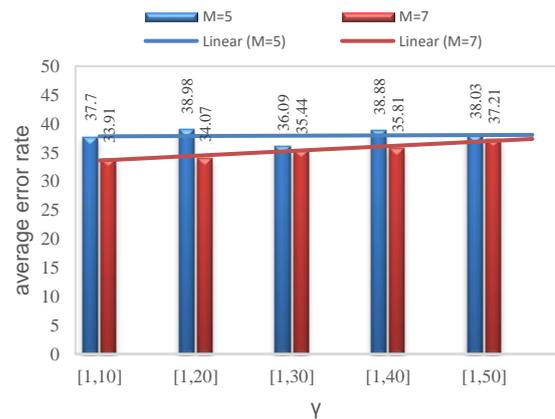


Figure 12. VELM classification error rate in synthetic example two

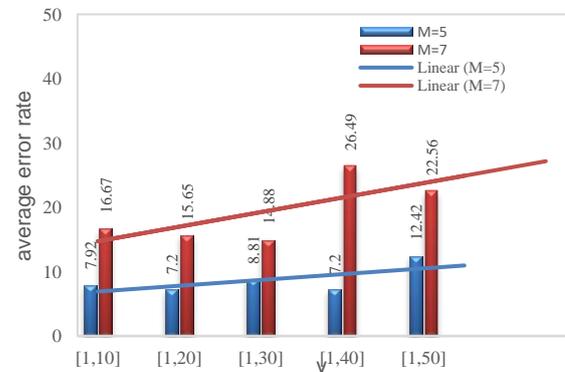


Figure 13. LUELM classification error rate in synthetic example two

neurons show less error rate than the JELM and VELMs with 5 neurons. Table 5 is created to compare average classification error rate on car dataset with previous works [31]. IDPCs methods are based on logistic regression (LR). Average error rate on Table 5 shows that proposed methods based on the ELM result in significant improvement and also among the proposed methods VELM has the best performance.

5. DISCUSSION

We tried to classify aggregated data described by multi-valued features. The essence of ELM is that the hidden layer of SLFNs does not need to be tuned. More specifically, this paper provides methods in which interval data can be compatible for Moore-Penrose inverse calculation in second layer of ELM and reduce classification error rate.

LR based methods and proposed algorithms have one problem in common: finding the optimum value for their parameters. an iterative optimization method is used in LR based methods to minimize the cost function, but the iterative optimization of network weights is avoided in proposed methods and they use

TABLE 4. Average error rate for car dataset

Proposed methods	M=5	M=7
MELM	19.17	20.83
JELM	20.83	15
VELM	7.92	6.54
LUELM	18.33	19.58

TABLE 5. Comparing proposed methods with other methods on car dataset

Methods	Error rate(%)
IDPC_sp1(second application) [31]	57.57
IDPC_sp(first application) [31]	48.48
KNN [30]	45
IDPC_CSP [31]	36.36
IDPC_VSP(maxrule) [31]	36.4
IDPC_VSP(minrule) [31]	30.3
IDPC_VSP(averagerule) [31]	30.3
IDPC_pp [31]	27.2
MELM	19.1667
LUELM	18.3333
JELM	15
VELM	6.5385

the randomization and Pseudo inverse to determine the network. On the other hand the results show that proposed methods have a significant classification error rate than LR based methods.

Training Time Complexity in logistic regression, means solving the optimization problem. and it is estimated as $O(N \times p)$.

IDPC-CSP classifier that utilized mid-point representation has $O(N \times p)$ computational complexity.

In IDPC-SP classifiers, data are defined by the lower and upper bounds of the intervals conjointly. computational complexity for this method is: $O(N \times 2 \times p)$

In IDPC-VSP classifier Each symbolic interval training sample is a matrix $2^p \times p$ corresponding to all possible combinations of the limits of intervals. So, the size of input matrix is $p \times (2^p \times p \times N)$. Considering $\alpha = (2^p \times p)$ as a constant value. So, computational complexity of IDPC-VSP is $O(2 \times ((2^p \times p \times N \times p)))$

IDPC-PP classifier is defined by the lower and upper bounds of the intervals separately. The analysis in this method consists of fitting two logistic binary regressions for each class. computational complexity for this method is sum of the computational complexity of lower bound and upper band based classifiers. So the estimation for computational complexity is $O(2 \times (N \times p))$.

There are two fundamental issues in neuro-computation: The first one is learning algorithm development and the second one is the network topology design. In fact, these two issues are closely related with each other. The learning ability of a neural network is not only a function of time, but also it is a function of the network structure.

A typical neural network contains an input layer, an output layer, and one or more hidden layers. The number of outputs and the number of inputs are usually fixed while the number of hidden layers and number of hidden neurons in each hidden layer are parameters that can be specified for each application [40].

ELM is a network which has a single hidden layer with L neurons. The time complexity of ELM is the sum of the calculations performed to obtain the weights between the input layer and the hidden layer and the weights between the hidden layer and the output layer. Assume that the size of the input matrix is $p \times N$ and the size of weights matrix between the input layer and the hidden layer is $L \times p$. In this case, the complexity of the matrix multiplication performed at this step is $O(L \times p \times N)$.

To calculate the weights in the second layer, ELM uses a Moore Penrose pseudo inverse that has a time complexity equal to $O(2 \times L \times N^2 + 2 \times L^3)$ for a matrix size of $L \times N$ and applying the common Singular Value Decomposition (SVD) method [41]. Therefore,

the associate computational complexity can be estimated as: $O(L \times p \times N + 2 \times L \times N^2 + 2 \times L^3)$

The proposed methods in this paper have a structure similar to the original ELM however their representations are different from each other. In MELM method the size of the input matrix is $p \times N$. Therefore, computational complexity equals to the basic ELM.

JELM method represents the data in the size of $2p \times N$. As a result, its computational complexity is:

$$O(L \times 2p \times N + 2 \times L \times N^2 + 2 \times L^3).$$

Each symbolic interval training sample is a matrix $2^p \times p$ corresponding to all possible combinations of the limits of intervals in VELM method. So the size of the input matrix is $p \times (2^p \times p \times N)$. Considering $\alpha = (2^p \times p)$ as a constant value. So, computational complexity of VELM is $O(L \times p \times \alpha N + 2 \times L \times N^2 + 2 \times L^3)$.

Input matrix in LUELM method is $p \times N$. In this method we have two networks which perform their calculations independently. Therefore, their computational complexity is added together and estimated as $O(2 \times (L \times p \times N + 2 \times L \times N^2 + 2 \times L^3))$.

6. CONCLUSION

In this paper, four new models of ELM are proposed to handle symbolic interval data. They have the architecture of a standard ELM with single-valued weights and biases, but the way interval data entered the network is different. In MELM, each interval is represented by the midpoints of intervals. JELM uses a pair of conjoint intervals. The vertices of intervals which has been used in VELM and LUELM is considered as the lower and upper bounds of the interval separately. Two Interval synthetic data and error rate criteria are used in order to determine the number of hidden layer nodes in each proposed pattern classifier model. The results show that MELM and LUELM produce significantly better results with five hidden layer nodes, while the JELM and VELM prefer seven hidden layer nodes to produce significant results. Proposed classifiers also used car interval dataset as a real synthetic dataset. Afterwards the results was compared with other methods and showed that the proposed methods have a better performance in comparison to other methods.

7. REFERENCES

- Hira, Z. M., and Gillies, D. F., "A review of feature selection and feature extraction methods applied on microarray data", *Advances in Bioinformatics*, Vol. 2015, (2015). DOI: 10.1155/2015/198363

- Dai, J., Liu, Y., Chen, j., and Liu, X., "Fast feature selection for interval-valued data through kernel density estimation entropy", *International Journal of Machine Learning and Cybernetics*, Vol. 11, (2020), 2607-2624. DOI: 10.1007/s13042-020-01131-5
- Yang, L. f., Liu, C., Long, H., Ashfaq, R. A. R., He, Y. L., "Further improvements on extreme learning machine for interval neural network", *Neural Computing & Applications*, Vol. 29, (2018), 311-318. DOI: 10.1007/s00521-016-2727-4
- Safaria, A., Hosseini, R., Mazinani, M., "A novel type-2 adaptive neuro fuzzy inference system classifier for modelling uncertainty in prediction of air pollution disaster", *International Journal of Engineering, Transactions B: Applications*, Vol. 30, (2017), 1746-1751. DOI: 10.5829/ije.2017.30.11b.16
- Mousavi, S. M., Makui, A., Raissic, S., Mojtahed, S.M.H., "A multi-criteria decision-making approach with interval numbers for evaluating project risk responses", *International Journal of Engineering, Transactions B: Applications*, Vol. 25, No. 2, (2012) 121-129. DOI: 10.5829/idosi.ije.2012.25.02b.05
- Taheri, A. A., Taghilou, M., "Towards a Uncertainty Analysis in Thermal Protection using Phase-change Micro/Nano Particles during Hyperthermia", *International Journal of Engineering, Transactions A: Basics*, Vol. 34, No. 1, (2021), 263-271. DOI: 10.5829/ije.2021.34.01a.29
- Moore, R. E., Interval analysis. Prentice-Hall, Englewood Cliffs, 1966.
- Sunaga, T., "Theory of an interval algebra and its applications to numerical analysis", *Japan Journal of Industrial and Applied Mathematics*, Vol. 26, (2009), 125-143. DOI: <https://doi.org/10.1007/BF03186528>
- Yeung, D. S., Ng, W. W.Y., Wang, D. F., Tsang, E. C. C., Wang, X. Z., "Localized generalization error model and its application to architecture selection for radial basis function neural network", *IEEE Transactions on Neural Networks*, Vol.18, (2007), 1294-1305. DOI: 10.1109/TNN.2007.894058
- Tsang, E. C. C., Wang, X. Z and Yeung, D. S., "Improving learning accuracy of fuzzy decision trees by hybrid neural networks", *IEEE Transactions on Fuzzy Systems*, Vol. 8, (2000), 601-614. DOI: 10.1109/91.873583
- Kameli, A., Javadian, N., Daghbandan, A., "Multi-period and Multi-objective Stock Selection Optimization Model Based on Fuzzy Interval Approach", *International Journal of Engineering, Transactions C: Aspects*, Vol. 32, (2019), 1306-1311. DOI: 10.5829/ije.2019.32.09c.11
- Roque, A. M. S., Mate, C., Arroyo, J., and Sarabia, A. N., "IMLP: Applying multi-layer perceptrons to interval-valued data", *Neural Processing Letters*, Vol. 25, (2007), 157-169. DOI: 10.1007/s11063-007-9035-z
- Bock, H. H., and Diday, E., Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data. Springer: Berlin Heidelberg, 2000.
- Carvalho, F. D. A. T. D., "Fuzzy c-means clustering methods for symbolic interval data", *Pattern Recognition Letters*, Vol. 28, (2007), 423-437. DOI: 10.1016/j.patrec.2006.08.014
- Carvalho, F. d. A. T. d., Souza, R. M. C. R. d., and Bezerra, L. X. T., "A Dynamical Clustering Method for Symbolic Interval Data Based on A Single Adaptive Euclidean Distance," in 9th Brazilian Symposium on Neural Networks (SBRN'06), Ribeirao Preto, Brazil., (2006). DOI: 10.1109/SBRN.2006.2
- D'Urso, P., Massari, R., Giovanni, L. D., Cappelli, C., "Exponential distance-based fuzzy clustering for interval-valued data", *Fuzzy Optimization and Decision Making*, Vol. 16, (2017), 51-70. DOI: 10.1007/s10700-016-9238-8

17. Feng, G., Ni, M., Yan, W., Xu, J., "A Preferential Interval-Valued Fuzzy C-Means Algorithm for Remotely Sensed Imagery Classification", *International Journal of Fuzzy Systems*, Vol. 21, (2019), 2212-2222. DOI: 10.1007/s40815-019-00706-x
18. Galdino, S., Maciel, P., "hierarchical cluster analysis of interval-valued data using width of range euclidean distance", in IEEE Latin American Conference on Computational Intelligence (LACCI) 2019, (2019). DOI: 10.1109/LA-CCI47412.2019.9036754.
19. Jeng, I. T., Chen, C. M., Chang, S. C., Chuang, C. C., "IPFCM clustering algorithm under euclidean and hausdorff distance measure for symbolic interval data", *International Journal of Fuzzy Systems*, Vol. 21, (2019), 2102-219. DOI: 10.1007/s40815-019-00707-w
20. Silva, A. P. D., Filzmoser, P., Brito, P., "Outlier detection in interval data", *Advances in Data Analysis and Classification*, Vol. 12, (2018), 785-822. DOI: 10.1007/s11634-017-0305-y
21. Rizo Rodríguez, S. I., de Assis Tenório de Carvalho, F., "A new fuzzy clustering algorithm for interval-valued data based on City-Block distance," in IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), (2019). DOI: 10.1109/FUZZ-IEEE.2019.8859017.
22. Ichino, M., Yaguchi, H., Diday, E., A fuzzy symbolic pattern classifier. In: Ordinal and symbolic data analysis. Diday E., et al. (Eds.) Springer, Berlin, 1996.
23. De Carvalho, S. R. M. C., and Frery, A. C., "Symbolic approach to SAR image classification", in IEEE International Geoscience and Remote Sensing Symposium (IGARSS'99), Hamburg, Germany, (1999). DOI: 10.1109/IGARSS.1999.774617.
24. D'Oliveira, S., De Carvalho, F. A. T., Souza, R. M. C. R. D., "Classification of SAR Images Through a Convex Hull Region Oriented Approach", in 11th International Conference on Neural information processing, (2004). DOI: 10.1007/978-3-540-30499-9_118
25. Ciampi, A., Diday, E., Lebbe, J., Perinel, E., and Vignes, R., "Growing a tree classifier with imprecise data", *Pattern Recognition Letters*, Vol. 21, (2000), 787-803. DOI: 10.1016/S0167-8655(00)00040-4
26. Singh, P., Huang, Y. P., "A four-way decision-making approach using interval-valued fuzzy sets, rough set and granular computing: a new approach in data classification and decision-making", *Granular Computing*, Vol. 5, (2020), 397-409. DOI: 10.1007/s41066-019-00165-7
27. Kowalski, P. A., Kulczycki, P., "Interval probabilistic neural network", *Neural Computing and Applications*, Vol. 28 (2017), 817-834. DOI: 10.1007/s00521-015-2109-3
28. Rossi, F., and Conan-Guez, B., "Multi-layer perceptron interval data", In: Classification, Clustering, and Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization, Jajuga K., Sokołowski A., Bock H. H., (Eds.) Springer, Berlin, Heidelberg., (2002). DOI: 10.1007/978-3-642-56181-8_47
29. Mali, K., and Mitra, S. "Symbolic classification, clustering and fuzzy radial basis function network", *Fuzzy Sets and Systems*, Vol. 152, (2005), 553-564. Doi: <https://doi.org/10.1016/j.fss.2004.10.001>
30. Appice, A., D'Amato, C., Esposito, F., and Malerba, D. "Classification of symbolic objects: A lazy learning approach", *Intelligent Data Analysis*, Vol. 10, (2006), 301-324. DOI: 10.3233/IDA-2006-10402
31. De Souza, R. M. C. R., Queiroz, D. C. F. and Cysneiros, F. J. A., "Logistic regression-based pattern classifiers for symbolic interval data", *Pattern Analysis and Applications*, Vol. 14, (2011), 273-282. DOI: 10.1007/s10044-011-0222-1
32. Huang, G. B., Zhu, Q. Y., Siew, and C. K., "Extreme learning machine: theory and applications", *Neurocomputing*, Vol. 70, (2006), 489-501. DOI: 10.1016/j.neucom.2005.12.126
33. Huang, G. B., Wang, D. H., and Lan, Y., "Extreme learning machines: A survey", *International Journal of Machine Learning and Cybernetics*, Vol. 2, (2011), 107-122. DOI: 10.1007/s13042-011-0019-y
34. Huang, G., Zhou, H., Ding, X., and Zhang, R., "Extreme learning machine for regression and multiclass classification", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 42, (2012), 513-529. DOI: 10.1109/TSMCB.2011.2168604.
35. Chetwynd, D., Worden, K., and Manson, G., "An Application of Interval-Valued Neural Networks to a Regression Problem", in Proceedings of the Royal Society A, Mathematica, Physical and Engineering Sciences, (2006). DOI: <https://doi.org/10.1098/rspa.2006.1717>
36. Adam, S. P., Likas, A. C., and Vrahatis, M. N., "Evaluating generalization through interval-based neural network inversion", *Neural Computing and Applications*, Vol. 31, (2019), 9241-9260. DOI: 10.1007/s00521-019-04129-5
37. Carvalho, F. A. T., Brito, P., and Bock, H. H., "Dynamic clustering for interval data based on L2 distance", *Computational Statistics*, Vol. 21, (2006), 231-250. DOI: <https://doi.org/10.1007/s00180-006-0261-z>
38. De Carvalho, J. T., Pimentel, L. X., Bezerra T., de Souza, R. M. C. R., "Clustering Symbolic Interval Data Based on a Single Adaptive Hausdorff Distance", in IEEE International Conference on Systems, Man and Cybernetics, Montreal, QC, Canada, (2007). DOI: 10.1109/ICSMC.2007.4413616.
39. De Souza, R. M. C. R., De Carvalho, F. A. T., Pizzato, D. F., "A Partitioning Method for Mixed Feature-Type Symbolic Data Using a Squared Euclidean Distance," in 29th Annual German Conference on Artificial Intelligence (KI2006), Bremen (Alemanha). Lecture notes on artificial intelligence—LNAI, vol 4314. Springer, Berlin, (2006). DOI: 10.1007/978-3-540-69912-5_20
40. Yu, H., "Network complexity analysis of multilayer feedforward artificial neural networks", *Studies in Computational Intelligence*, Vol. 268, (2010), DOI: 10.1007/978-3-642-10690-3_3
41. Ahmadian, A. M., Zirwas, W., Ganesan, R. S., Panzner, B., "Low Complexity Moore-Penrose Inverse for Large CoMP Areas with Sparse Massive MIMO Channel Matrices", in IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), (2016), DOI: 10.1109/PIMRC.2016.7794773.

Persian Abstract

چکیده

داده‌های فاصله‌ای معمولاً در موقعیت‌هایی مورد استفاده قرار می‌گیرند که عدم صحت و تغییرپذیری وجود دارد. در این مقاله یادگیری شبکه عصبی ELM برای طبقه‌بندی داده‌های بازه‌ای ارائه شده است. IELM مانند ELM، دو مرحله دارد. در مرحله اول، وزن‌های اتصال لایه ورودی و لایه پنهان به طور تصادفی تولید می‌شوند و در مرحله دوم، برای تعیین وزن‌های بین لایه پنهان و لایه خروجی به کمک شبه معکوس، از روش Moore-Penrose استفاده می‌کند. در این مقاله چهار روش طبقه‌بندی برای مدیریت داده‌های فاصله‌ای مبتنی بر شبکه عصبی ELM پیشنهاد شده است. مورد اول از یک نقطه میانی فواصل برای هر مقدار ویژگی استفاده می‌کند سپس شبکه عصبی ELM کلاسیک طبقه‌بندی را انجام می‌دهد. مورد دوم هر مقدار ویژگی را به عنوان یک جفت ویژگی کمی در نظر می‌گیرد و از یک شبکه عصبی ELM کلاسیک برای طبقه‌بندی استفاده می‌کند. مورد سوم از طریق رئوس آن ویژگی‌های فاصله را نشان می‌دهد و همچنین یک شبکه عصبی ELM کلاسیک برای طبقه‌بندی بکار می‌رود. مورد چهارم هر بازه را به عنوان یک جفت ویژگی کمی در نظر می‌گیرد، بعد از آن دو شبکه عصبی ELM جداگانه بر اساس حد بالا و حد پایین آموزش می‌بینند و سپس نتایج را به طور مناسب ترکیب می‌کند. الگوریتم‌ها روی مجموعه داده‌های مصنوعی و واقعی آزمایش شده‌اند. مجموعه داده‌های مصنوعی برای تعیین تعداد گره‌های لایه پنهان در شبکه عصبی ELM اعمال می‌شود. میزان خطای طبقه‌بندی به عنوان معیار مقایسه در نظر گرفته شده است. میزان خطای به دست آمده برای هر چهار روش پیشنهادی به ترتیب 19.167% ، 15% ، 6.536% و 18.333% است. آزمایش‌ها سودمندی این طبقه‌بندها را برای طبقه‌بندی داده‌های بازه‌ای نشان می‌دهد.