

Autonomous Data Density pruning fuzzy neural network for Optical Interconnection Network

Paulo Vitor de C. Souza and Eduardo A. Soares and Augusto J. Guimaraes and Vanessa S. Araujo and Vinicius Jonathan S. Araujo and Thiago S. Rezende

Abstract Traditionally, fuzzy neural networks have parametric clustering methods based on equally spaced membership functions to fuzzify inputs of the model. In this sense, it produces an excessive number calculations for the parameters' definition of the network architecture, which may be a problem especially for real-time large-scale tasks. Therefore, this paper proposes a new model that uses a non-parametric technique for the fuzzification process. The proposed model uses an autonomous data density approach in a pruned fuzzy neural network, wich favours the compactness of the model. The performance of the proposed approach is evaluated through the usage of databases related to the Optical Interconnection Network. Finally, binary patterns classification tests for the identification of Temporal Distribution (Asynchronous or Client-Server) were performed and compared with state-of-the-art fuzzy neural-based and traditional machine learning approaches. Results demonstrated that the proposed model is an efficient tool for these challenging classification tasks.

Paulo Vitor C. Souza
CEFET-MG and UNA-Betim, Av. Amazonas, 5.253, Nova Suia, Belo Horizonte, MG, Brazil,
30.421-169, Phone: +55 (31) 3319-7000, e-mail: goldenpaul@informatica.esp.ufmg.br

Eduardo A. Soares
Department of Engineering, Federal University of Lavras(UFLA), Lavras, Minas Gerais, Brazil e-mail: e.almeidasoares@lancaster.ac.uk

Augusto J. Guimaraes, Vanessa S. Araujo, Vinicius Jonathan S. Araujo, Thiago S. Rezende
UNA-Betim, Av. Gov. Valadares, 640 - Centro, Betim - MG, Brazil, 32.510-010 e-mail: augustojunioguimaraes, silvarezendethiago@gmail.com, v.souzaaraujo@yahoo.com.br, vinicius.j.s.a22@hotmail.com

1 Introduction

Parallel computing has evolved with the need for more advanced computing processes. Therefore, the need for more efficient algorithms to solve complex problems with adequate time and efficiency. To verify the ability of methods to perform in several parallel procedures that may involve analysis or simulations of communication between devices, the prediction of device behavior is considered.

Works were performed to predict the 2D SOME-Bus multiprocessor architecture using the message transmission programming model [2], for such a database was created for training and testing of intelligent models using OPNET Modeler. It has as the main feature to simulate the 2D SOME-Bus pass-through multiprocessor architecture, thus allowing the collected values to reflect the communication behavior of real systems that communicate in parallel [4].

Intelligent models able to predict factors involved in computer communications databases may be fuzzy neural networks [72]. They operate in a way that allows collected data to be replicated through devices that use the interpretability of fuzzy systems and different methods of training provided by artificial neural networks. These models can predict desired values and extract pertinent information from databases in order to support the construction of expert systems based on IF-THEN rules [72].

Therefore, this paper proposes a pruned fuzzy neural network to define the temporal distribution of parallel connections among computerized devices. The proposed model uses in its fuzzification layer a non-parametric technique for the definition of representative spaces. This approach applies the concepts of data density recently proposed by Angelov and Yager. [6]. An autonomous approach proposed by Gu et al. [33] is used (Autonomous Data Density Clustering-ADDC). Additionally, a pruning approach based on F-scores [28] helps in the definition of the most representative neurons for the problem approached. Finally, an extreme learning machine [38] is used to calculate analytically the weights that connect the first two layers (fuzzy inference system) to the third layer (neural aggregation network). The results obtained by the proposed model are compared with other state-of-the-art intelligent approaches in order to prove the effectiveness of the proposed technique.

The main contribution of this paper is the use of an autonomous intelligent hybrid model based on the density of the data submitted to the model to build compact fuzzy rules to deal with problems related to network connections. **Another relevant factor is to look for a hybrid alternative capable of bringing practical answers to the problem with less time when compared to traditional approaches.**

The paper is organized as follows: In the next section (Sec. 2), a literature review is presented in order to assist the reader in the understanding of the contexts involved. In Section 3, the proposed model is presented and defined. Results and discussions are shown in Section 4. Conclusion and future research directions are given in Section 5.

2 Literature Review

2.1 Optical Interconnection Network

2D SOME-Bus is a robust implementation of computer architecture. The same is a simultaneous two-dimensional optical multiprocessor bus [2]. In the Some-Bus architecture, we have the 2D SOME-Bus and the 1D SOME-Bus, used for data transmission on the bus. Each of them has different performance characteristics, such as the timely execution of the operations. The order of complexity of the two is different. While an all-to-all communication task is performed, where a different message is sent to each processor, the 2D SOME-Bus can execute this on an $O(N^2)$ and the 1D SOME-Bus, in turn, runs on $O(N)$. When performing a barrier synchronization, the 2D SOME-Bus has the time of synchronization of $O(N)$ while the 1D SOME-Bus has a time of $O(1)$. 2D SOME-Bus are nodes connected to a 1D SOME-Bus. An electro-optical converter allows the routing and transmission of messages on a bus [2].

Figure 1¹ presents an example of a 2D SOME-Bus architecture model where O_h represents horizontal output channel, and O_v represents vertical output channel, while I_h represents horizontal input queue, and I_v represents vertical input queue. Bus_h0/Bus_h1 represents horizontal SOME-Bus of number zero and one, respectively.

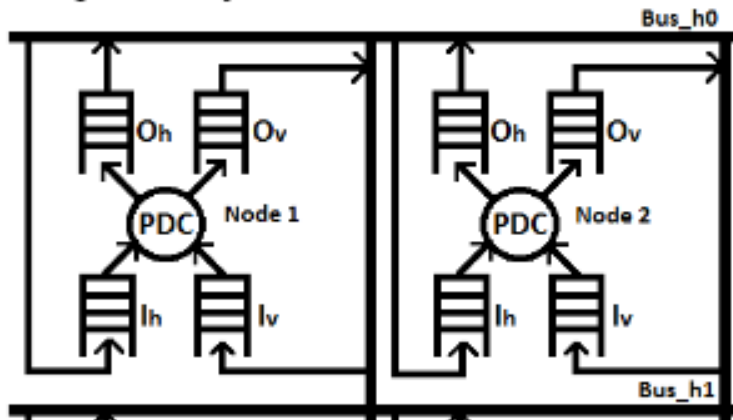


Fig. 1 2D SOME-Bus architecture

¹ Based on Abasikeles [1]

2.2 *Related works*

The work involving optical interconnection network activities have been carried since before the 1990s, as in the work of Sampsell et al. [64] and Johnson et al. [41]. During the '90s, works such as those of Louri et al. [50] that work with parallel computing and Giles et al. [29] that uses concepts of computational intelligence. Other reseraches were developed in the 2000s to approach parallel optical architectures for large databases [55]. Researchers in the artificial intelligence field commonly study the techniques to predict performance measures of a multiprocessor architecture. In recent years, artificial intelligence and machine learning techniques have been applied in parallel computing as in Akay et al. [3,4], Zayid et al. [79] and Aci et al. [2]. Other intelligent techniques such as clustering assistance in multiprocessors evaluation [59], a framework proposed by [54], sorting and matrix operations [44], a serial synchronized multistage interconnection [77], Hopfield neural network [27].

However, these approaches are outdated, especially the ones highlighted by Kotilainen et al. [46] and Cavalieri et al. [24], that since the early 1990s have been addressing solutions for parallel computing through the use of intelligent techniques.

Therefore, this research field has been studied for a long time because of its challenging nature.

2.3 *Fuzzy Systems, Artificial Neural Networks, and Hybrid Models*

Intelligent approaches seek the simulation of human characteristics in problem-solving through the replication of everyday learning and experiences. Given this context, the biological neuron modeling [52] has triggered several studies and techniques to simulate human learning abilities in simulated models. The artificial neural networks stand out as models that use the concepts of cerebral synapses and biological neurons to solve problems in the computational environment. These models can measure standards, approximate values of functions, treat aspects related to time series, among others. However, this type of model does not allow an adequate interpretation of the obtained results and lacks more extensive resources to achieve the learning of the database submitted to the model. To counter this kind of situation, in 1965, Professor Zadeh developed the theory of fuzzy sets [78], capable of transforming the nature of a problem into situations more representative of the evaluated problem.

Some intelligent models may use the data itself to extract knowledge about a problem. To combine the concepts of interpretability of fuzzy system concepts, proposed by Zadeh [78] with the mainstream training techniques of the artificial neural networks, hybrid models called fuzzy neural networks were developed [56].

The fuzzy neural networks (FNN) have diverse approaches in the literature, covering the resolution of several complex problems in numerous areas of industry, science, health, and others. The use of hybrid models has been practiced in several

areas, like voice-controlled robot [25], industrial process [51], prediction model of failure depth of mine floor [73], time series forecasting [22, 60] among others [43].

2.4 Fuzzification techniques in Fuzzy Neural Networks

The concepts of the nature of the data are fundamental to understand the clustering technique used in this paper. [8]. This step is crucial for hybrid models because it is a well-elaborated analysis of the problem, where the choice of the partition mode or technique used to fuzzify the database can affect the performance characteristics and even the interpretability of the results. This process allows the creation of regions in the decision space, formed by membership functions that can be triangular, trapezoidal, or Gaussian [63]. In general, there are techniques based on membership functions that are equally spaced or not. Techniques are done in data clustering, density, among others [63].

For Ross et al. [63], the main characteristic of the fuzzification process is to transform numerical values into representative values of the problem, allowing the data to be interpreted according to the choice of representation. A typical example is taking the weight of a group of people and proceeding with the process of interpretability. For the hybrid model, the weight value of a person is only 80 kg for the linguistic representation of this value, in general, using association functions. The value of an obese, regular, or lean person may vary depending on the range of people involved in the problem. If the people weighed weigh mostly 90 kg, the person who weighs 80 is considered to be a regular or even lean person. If the group evaluated is of newborn children, the person weighing 80 kg will be interpreted as obese. Fuzzification presents ways of assessing a value according to its analyzed problem, allowing an evaluated subject to have multiple interpretations depending on the target problem.

In this process, fuzzy rules are constructed with the chosen fuzzification process, allowing the decision processes to be interpretable and logical, especially if there is the aid of a specialist. This process is analysis based and definition of fuzzy rules and the determination of the resulting region for the problem. In general, these rules are conditional (IF B is C THEN Z is J) or nonconditional (B is C). To verify the importance and the representativeness of a rule, we highlight aggregation operations (calculates the relevance of a given fuzzy rule for the analyzed problem) and the composition (derives the influence of each rule on the output variables) [57].

Some techniques used to perform fuzzification processes in hybrid models of neural networks and fuzzy systems. The paper of Kim et al. [45] and Lemos et al. [47–49], Bordignon et al. [13] uses an algorithm based on fuzzy c-means [11].

A model that is used for fuzzification is the ANFIS model proposed by Jang [40], where its versions can generate membership functions that are equally or differently spaced. Several works such as this use this approach like as de Campos Souza et al. [16, 17, 20, 21, 35, 68–71] and Guimaraes et al. [34], allowing the data set to be partitioned in a grid format, allowing for inferences and interpretability about the dataset studied.

Fuzzy neural network models also use the density of training data to create the neurons in the first layer as the Bordignon et al. [12] model.

Recent models work in fuzzy neural networks to use clustering concepts based on Empirical Data Analytics to use a self-organized direction aware data partitioning algorithm for Pulsar detection [23] and binary pattern classification problems [19].

A model of fuzzy neural networks normalized based on incremental clustering of data density was used to assist in predicting efforts for the construction of software [18] it was proposed for De Campos Souza et al.

3 ADDC Pruned Fuzzy Neural Network

In this section, we will present concepts related to the fuzzy neural network structure, its training method, fuzzification, and defuzzification processes. The novelty of this new algorithm and what differs it from other hybrid approaches is that it is entirely based on the data, and their mutual distribution is computed in the data space to create a fuzzification process in the first layer of the model. There is no need for specific predefined limits or any user input in the proposed method to create fuzzy neurons, so the approach is non-parametric, which facilitates its use in fuzzy neural network models.

3.1 Model Architecture

The fuzzy neural network described in this section is composed of three layers. In the first layer, fuzzification is used through the concept of data density. The centers of the *data clouds* formed by ADDC are used to create the fuzzy Gaussian neurons in the first layer. The weights and bias of these neurons are randomly defined in the range of zero to one. In the second layer, the logical neurons of the unineuron [47] type are applied. These neurons have weights and activation functions determined randomly and through t-norms (product) and s-norms (probabilistic-sum) in order to aggregate the neurons of the first layer. To define the weights that connect the second layer with the output layer, the concept of an extreme learning machine [38] is used to act on the neuron with a linear activation function.

Unineuron is used to construct fuzzy neural networks in the second layer to solve pattern recognition problems and bring interpretability to the model.

The second layer is composed by L neuro-fuzzy layer. Each neuron performs a weighted aggregation of some of the first layer outputs. This aggregation is performed using weights w_{il} (for $i = 1 \dots N$ and $l = 1 \dots L$). For each input variable j , only one first layer output a_{jl} is defined as input of the l -th neuron. So that \mathbf{w} is sparse, each neuron of the second layer is associated with an input variable.

Finally, the output layer is composed of one neuron whose activation functions are linear. Figure 2 illustrates the feedforward topology of the fuzzy neural networks considered in this paper.

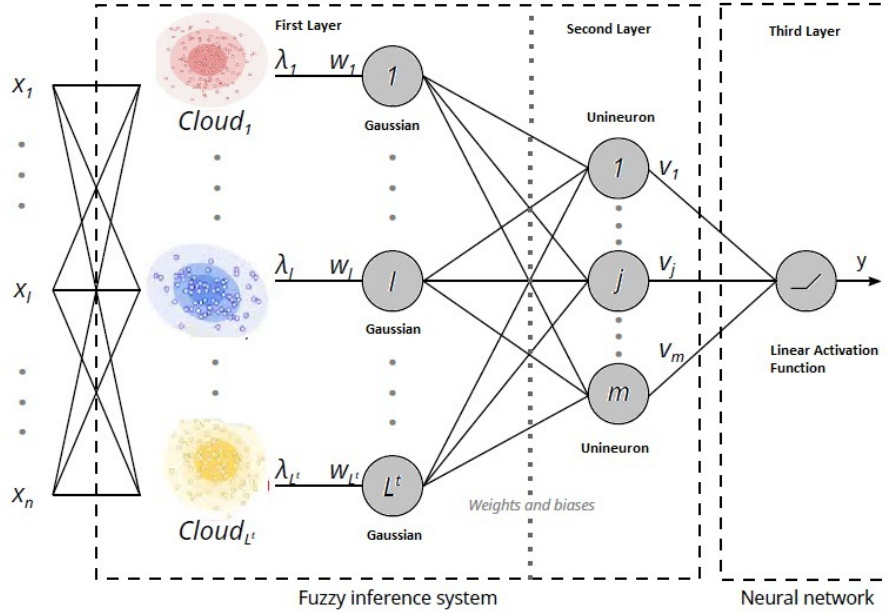


Fig. 2 FNN architecture

3.1.1 First layer- Data Density fuzzification

The first layer is composed of neurons whose activation functions are membership functions of fuzzy sets defined for the input variables. For each input variable x_{ij} , L clusters are defined A_{lj} , $l = 1, \dots, L$ whose membership functions are the activation functions of the corresponding neurons. Thus, the outputs of the first layer are the membership degrees associated with the input values, i.e., $a_{jl} = \mu_l^A$ for $j = 1 \dots N$ and $l = 1 \dots L$, where N is the number of inputs and L is the number of fuzzy sets for each input results by ADDC.

The ADDC method works equally effectively with various types of distance/likeness and an arbitrary number of data dimensions metrics. It starts from scratch, auto-sets the data pattern concerning density, and features a highly accurate clustering performance acting autonomously to find the data patterns and group them in a coherent manner [8].

Efficiently, the model uses several types of distance metrics including Euclidean, Mahalanobis and direction-aware distance metrics due to its *data cloud* nature [7]. In this paper, consider the calculation of Euclidean distances as [8]:

$$\pi_k(x_i) = k((x - \mu_k)^T(x - \mu_k) + X_k - \mu_k^T \mu_k) \quad (1)$$

and Mahalanobis type of distance:

$$\pi_k(x_i) = k((x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) + X_k - \mu_k^T \Sigma_k^{-1} \mu_k) \quad (2)$$

where $X_k = \frac{1}{k} \sum_{x=1}^k x_i^T \sum_k^{-1} x_i$

These two ways of evaluating the distance and sum of measures of dissimilarity between all the points involved in a clustering analysis help to understand the concepts of cumulative proximity, π_k . It represents how a specific data point is close to all other points of data. In this paper, when the cumulative proximity is normalized, it will be considered as Eccentricity and is described as [8]:

$$\xi_k(x) = \frac{2\pi_k(x)}{\sum_{i=1}^k \pi_k(x_i)}, \sum_{i=1}^k \pi_k(x_i) > 0 \quad k > 1 \quad (3)$$

That allows the model to be autonomous and decide the techniques that best enable the data clustering according to its nature. Within its approaches, the model uses to perform the clustering of the training data of the model [8, 9]. Finally, density plays a significant role in the ADDC algorithm. Data density is inversely proportional to the eccentricity. Within this context, it is known that, the closer the particular data point is to other points, the smaller its cumulative proximity is, and the higher its density is [8]. Data density in this paper is considered like [8]:

$$D_k(x_i) = \frac{1}{\varepsilon_k(x_i)} = \frac{E(\pi_k(x))}{2\pi_k(x_i)} \quad (4)$$

Although cumulative proximity and density can both provide sufficient information about the data pattern, density is comparatively superior because of three facts:

- 1) Monotonic;
- 2) The maximum value is 1;
- 3) Asymptotically tends to zero when Cumulative Proximity tends to infinity [8].

Briefly, as advantages of using the ADDC technique, we have [8]:

- No prior knowledge of the dataset is required;
- No user parameter (such as the number of centers) is required; -Entirely based on the data and their mutual distribution in the data space;
- No need to choose a data distribution model. To find the centers of the input data and perform the data fuzzification technique three stages of ADDC method are executed [8]:

- 1) initial center and radius formation;

$$x^{*1} \leftarrow x^{j^*}, j^* = \operatorname{argmax}_{i=1}^k D_k(x_x) \quad (5)$$

where x_1, x_2, \dots, x_k is considered the set of original problem of data submitted to the clustering problem.

2) center and radius updating using The well-known Chebyshev inequality [65];

$$P(\|\mu - x\|^2 \leq n^2 \sigma^2) \geq 1 - \frac{1}{n^2} \quad (6)$$

The equation 6 refers to the calculation of the update of the parameters for the Euclidean distance.

3) clusters final adjustment

$$\text{Cluster_label} = \operatorname{argmin}_{i=1,2,\dots,K} \|x - x_i^*\| \quad (7)$$

where K is the number of clusters.

The final cluster update depends on several factors, for example, when two clusters are defined as double center overlays, or when they are defined as superimposed on the single-center or when the evaluated centers are slightly overlapping.

Studies have been carried out using the technique where papers stand on parallel programming [32,39], autonomous clustering techniques [31], clustering techniques for regression problems and date classification stream [67], prediction of terrorist activities [37], and clustering methods for probabilistic graphs, [36]. **An example of a graphical representation of how the technique identifies centers is given in figure 3.**

Each data group represents a distinct cluster, and the algorithm was able to identify 8 data groups in this example.

3.1.2 Second layer- Logical Neurons fuzzy and fuzzy rules

Second layer of the model is composed by unineurons [47]. Unineurons use the concepts of uninorm [76] to perform more simplified operations according to the activation function of the fuzzy neurons. Its format allows the unineuron to use either concepts of a neuron *and*, or a neuron *or*. The processing of neurons occurs at two levels. At the first level of L_1 locations the input signals are combined individually with the weights. In the second, at global level L_2 , a global aggregation operation is performed on the results of all first-level combinations. The second layer performs the aggregation of the L_c neurons from the first layer through the unineurons proposed by Lemos et al. [47]. In this paper is expressed as follows:

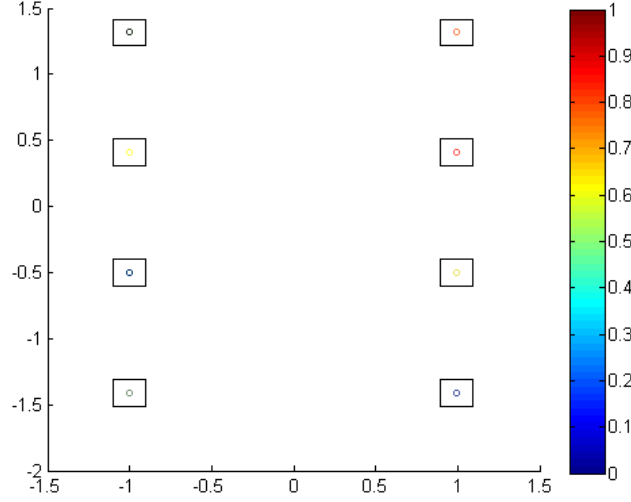


Fig. 3 ADDC Example

$$U(x, y) = \begin{cases} o T(\frac{x}{o}, \frac{y}{o}), \text{if } y \in [0, o] \\ o + (1 - o)S(\frac{x-o}{1-o}, \frac{y-o}{1-o}), \text{if } y \in [o, 1] \\ \delta(x, y), \text{otherside} \end{cases} \quad (8)$$

and

$$\delta(x, y) = \begin{cases} \max(x, y) & \text{if } g \in [0, 0.5] \\ \min(x, y) & \text{if } g \in (0.5, 1] \end{cases}$$

where T is a t -norm, S is a s -norm and o is the activation of the fuzzy neuron. In this paper, we considered the t -norm operator the product and s -norm operator as the probabilistic sum.

The unineuron proposed in Lemos et al. [47] performs the following operations to compute its output:

- 1- each pair (a_i, w_i) is transformed into a single value $b_i = \mathbf{h}(a_i, w_i)$;
- 2- calculate the unified aggregation of the transformed values $\mathbf{U}(b_1, b_2 \dots b_n)$, where n is the number of inputs.

The function p (relevancy transformation) is responsible for transforming the inputs and corresponding weights into individual transformed values. This function fulfills the requirement of monotonicity in value which means if the input value increases the transformed value must also increase. It also meets the requirement zero importance elements should have in effect and the normality of importance of one. Finally, the function p can bring consistency of effect of w_i . A formulation for the p function can be described as [47]:

$$p(w, a, o) = wa + \bar{w}o \quad (9)$$

using the weighted aggregation reported above the unineuron can be written as:

$$\mathbf{z} = UNI(w; a; o) = U_{i=1}^n p(w_i, a_i, o) \quad (10)$$

3.1.3 Third Layer - Aggregation of the Neural Network

The aggregation of the ANN uses the simple concept bias and weights, which in this case, are defined analytically by the extreme machine learning [38].

The output of the model is defined as:

$$y = \beta \sum_{j=0}^l f_{linear}(z_j, v_j) \quad (11)$$

where $z_0 = 1$, v_0 is the bias, and z_j and v_j , $j = 1, \dots, l$ are the output of each fuzzy neuron of the second layer and their corresponding weight, respectively. The β function acts similarly to the sign function, which returns value 1 if the analyzed number is positive and -1 if it is negative. Thus the β function is described as:

$$\beta = \begin{cases} 1, & \text{if } f_{linear}(z_l, v_l) > 0 \\ -1, & \text{if } f_{linear}(z_l, v_l) < 0 \end{cases} \quad (12)$$

This argument is critical for the definition of the binary classification problems, as the one analyzed in this paper.

3.2 Model Training

Subsequently, following the determination of the network topology, the predictions of the evaluation of the vector of weights' from the output layer are performed. In this paper, this vector is considered by the Moore-Penrose pseudo-inverse [68] and is based on the training techniques proposed by Huang et al. [38]:

$$\mathbf{v} = \mathbf{Z}^+ \mathbf{y} \quad (13)$$

where \mathbf{Z}^+ is pseudo-inverse of Moore-Penrose [5], of \mathbf{z} which is the minimum norm of the least-squares solution for the weights of the output layer.

The learning method can be synthesized, as demonstrated in Algorithm 1.

3.2.1 Pruning Technique

Fuzzy neural network models that have a high number of neurons can compromise performance in obtaining answers due to problems when using pseudo-inverse concepts to find the final weights of the network. Another problem that can occur in

Algorithm 1: ADDC-PFNN training

-
- (1) Load data
 - (2) Initialize the ADDC technique and define autonomous radius values, ρ and centre formation.
 - (3) Updating centre and ρ
 - (4) Get L cluster in the first layer using ADDC.
 - (5) Construct L fuzzy neurons with Gaussian membership functions constructed with center values derived from ADDC and σ defined at random.
 - (6) Define the weights and biases of fuzzy neurons at random.
 - (7) Construct L *unineurons* with random weights and bias on the second layer of the network by welding the L fuzzy neurons of the first layer.
 - (8) For all K inputs **do**
 - (8.1) Calculate the mapping $\mathbf{z}(x_i)$
 - end for**
 - (9) Select significant L_ρ neurons using the f-scores.
 - (10) Estimate the weights of the output layer using Eq. (13)
 - (11) Calculate the output of the model using an artificial neuron with linear activation function using Eq. (11).
-

the training of FNN is the possibility of nerve cells that decrease the accuracy of the model. In fuzzy neural networks, the same amount of neurons in the first layer is that of the second layer. To decrease this exponential relationship and facilitate pseudo-inverse calculations. In the models of de Campos Souza et al. [68, 71] Guimaraes et al. [34] and Batista et al. [10], resampling regularization techniques were used. In the model of Campos Souza [14], the concept of pruning was used through the f-scores proposed by Gao et al. [28] initially to prune structures trained with ELM. The f-scores technique has two particularities where its numerator indicates the discrimination between the positive and negative sets, and the denominator is the sum of the deviation within each set of resources. A higher F-score indicates that the support has more discriminative power [28].

The F-score technique was initially applied in an SVM model for feature selection in pattern classification tasks [26]. The promising results allowed its application in trained models based on the extreme learning machine and in neuro-fuzzy hybrid models. Therefore such an approach also applies to problem-solving in this paper, given the nature of pattern classification involved in the Optical Interconnection Network problem. To do so, consider the F-score assessment given by the equation:

$$F(i) = \frac{(\bar{a}_i^{(+)} - \bar{a}_i)^2 + (\bar{a}_i^{(-)} - \bar{a}_i)^2}{\frac{1}{n_+ - 1} \sum_{k=1}^{n_+} (a_{k,i}^{(+)} - \bar{a}_i^{(+)})^2 + \frac{1}{n_- - 1} \sum_{k=1}^{n_-} (a_{k,i}^{(-)} - \bar{a}_i^{(-)})^2}, \quad (14)$$

where $\bar{a}_i^{(+)}$, $\bar{a}_i^{(-)}$ and \bar{a}_i are the mean of the positive, negative and whole samples, respectively, and is the k -th feature value in the i -th feature vector ².

² Similarly, the technique used by Gao et al., instead of analyzing samples, analyzes the membership degree values that make up the neurons of the first layer.

The great advantage of using this feature selection technique is that it is also capable, as well as regularization techniques, of forming subgroups of neurons with higher representative power in solving the problem. So the pruning approach happens quickly, efficiently, and with reduced impact runtime for model training.

Therefore, the L neurons obtained from the first layer are evaluated for power relevance, and the F-score technique estimates 50% of the most relevant neurons through the mean values of f-score, without using parameters in a single step [28]. These pruned neurons are called in this paper by L_p .

Recently, the F-score technique was also used to select significant neurons in a fuzzy neural network that sought to solve health problems, more specifically in the solution of cryotherapy and immunotherapy treatments [42]. The significant difference between the two hybrid architectures lies in the fuzzification method, which, unlike this paper, uses equally spaced membership functions, thus generating exponentially complex fuzzification.

4 Tests

4.1 Database use in the tests

Parallel computing processing was performed using simulations with a software called OPNET Modeler [4]. This software simulates the passage of pensions, the procedure used as the mechanism of communication, in which any processor can send to the network a point-to-point message destined to any other processor. A queue is considered in calculations consisting of a First-Out First-Out buffer with the packet arriving randomly according to a Poisson probability process and a processor that retrieves packets from the buffer at a defined service rate. In all simulations, it is assumed that the processor at each node removes an input queue pack, performing its processing for some time. When the period expires, an output data message is generated. For this type of simulation, the size of each input, a queue has been evaluated as infinite, and a processor is inactive only when all its input queues are empty. The following characteristics were collected from the reported process [2]³: *Node Number*: number of the node's network. In this paper, we use $8 \times 8 = 2$ and $4 \times 4 = 1$.

Thread Number: The number of threads in each node of the simulation.

Spatial Distribution: Uniform (UN=1), Hot Region (HR=2), Bit reverse (BR=3) and Perfect Shuffle (PS=4) traffic models.

Temporal Distribution: Temporal distribution of packet generation were utilized client-server traffic (1) and asynchronous traffic (2).

T/R: Message transfer time (T) and Thread run time (R).

Processor Utilization: Average of the percent of the time that threads are running in the processor.

³ The database can be found in <https://archive.ics.uci.edu/ml/datasets/Optical+Interconnection+Network+>

Channel Waiting Time: Average waiting time of a packet at the output channel.

Input Waiting Time: Average waiting time of a packet until the processor services it.

Network Response Time: The time between a request and received a message at the output channel and in the input queue.

Channel Utilization: The percent of the time that the channel is busy transferring packets to the network.

Statistical data from the database used in the test are presented in Table 1, where the standard deviation of the samples is highlighted in parentheses.

Table 1 Statistical data from the database used in the test. [2]

Predictor Variables	Min.	Max.	Mean
Node number	16	64	40 (24.02)
Thread number	4	10	7 (2.24)
Spatial distribution	1	4	2.50 (1.12)
Temporal Distribution	1	2	1.50 (0.50)
T/R	0.10	1	0.55 (0.29)
Processor utilization	0.20	0.99	0.65 (0.20)
Channel waiting time	0.95	1627.33	377.46 (381.98)
Input waiting time	33.04	892.85	333.25 (233.72)
Network response time	18.98	6065.74	1529.87 (1191.29)
Channel utilization	0.14	0.99	0.72 (0.20)

As can be seen from figure 4, the data is quite scattered, and its nature is extremely complex.

4.2 Settings, evaluation methods and models used in the test.

The result of the cross-validation test evaluated the best value of accuracy. The outputs of the model were normalized to -1 and 1 to aid the correct calculations. The factors evaluated in this paper are as follows:

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (15)$$

$$sensitivity = \frac{TP}{TP + FN} \quad (16)$$

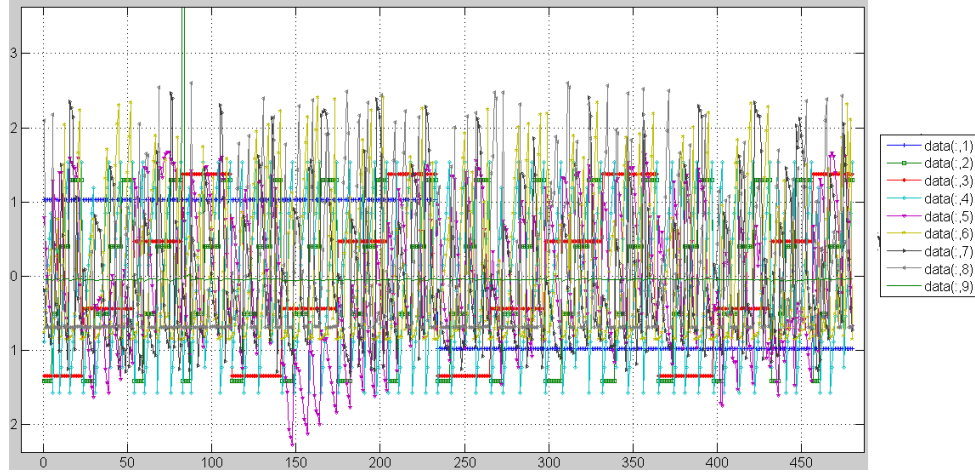


Fig. 4 Graphical representation of database dimensions.

$$specificity = \frac{TN}{TN + TP} \quad (17)$$

$$AUC = \frac{1}{2} (sensitivity + specificity) \quad (18)$$

where, TP = true positive, TN = true negative, FN = false negative and FP = false positive.

The final values of a neuron after pruning of unnecessary neurons are highlighted in Table 2.

The model proposed in this paper was compared with intelligent models that also use pruning and regularization techniques to define the best neurons in the hidden layers of the model. Two models of the fuzzy neural network were included in the test: where one of them is pruned using the concept of F-scores [28], called PFNN [14], a model that selects the best neurons through the bolasso technique (RFNN) [21]. The parameters used in this model were as follows: in the fuzzification, the concept of ANFIS [40] was applied to the generation of equally spaced membership functions. The values of membership functions, M , consensus threshold, α , and the number of bootstraps, bt , were decided through the cross-validation technique [30], choosing the value that maximizes the training accuracy of the models. Values ranges were as follows: $M = 3, 4, 5$, $bt = 16, 32, 64$, and $\alpha = 0.5, 0.6, 0.7$. In the two models, the unineuron in the second layer and a linear activation function in the output neuron were adopted. Other models chosen to be part of the tests are feedforward artificial neural networks that use ELM and different techniques of pruning to update their parameters. The most traditional, OP-ELM [53] uses to select the best neurons the technique Multiresponse Sparse Regression [66]. The model P-ELM [62] selects the most correlated neurons with the output vector according to Chi-Square statistics and considers the number of nodes that minimize

the AIC index. The AF-ELM model [58] that uses the affinity matrix [74] to perform the selection of neurons. Finally, the model CC-ELM [15] selects the most correlated neurons with the expected outputs of the model through the correlation coefficient. In all this models experiments we set the initial number of hidden nodes $k \in 100, 150$ (the best values of k are also selected via cross-validation). The hidden layer parameters were sampled from an uniform distribution $U [0.1, 0.1]$. Sigmoidal activation functions have been used for all neurons.

4.3 Test results

The values in parentheses are standard deviations. Simulations were performed on a Core (TM) 2 Duo CPU, 2.27 GHz, with 3-GB RAM. Time is represented by the sum of training time and test (seconds) in each of the models. Neurons represent the most representative neurons after the pruning or regularization of the models.

Table 2 Results Optical Interconnection database.

Models	Accuracy	AUC	Sens.	Spec.	Neuron	Time
This Paper	80.16 (5.12)	0.8125 (0.0811)	0.8345 (0.1024)	0.7905 (0.1003)	10.15 (0.03)	2.04 (0.03)
PFNN	79.93 (4.20)	0.7969 (0.1011)	0.8244 (0.0427)	0.7695 (0.0592)	106.03 (18.21)	129.94 (16.22)
RFNN	79.22 (4.85)	0.7958 (0.0418)	0.8193 (0.0824)	0.7723 (0.1004)	122.54 (21.54)	130.21 (17.32)
OP-ELM	78.14 (10.22)	0.7764 (0.0360)	0.8026 (0.0555)	0.7502 (0.0490)	69.21 (6.04)	84.62 (7.21)
P-ELM	75.05 (9.67)	0.7490 (0.1243)	0.8026 (0.0555)	0.6954 (0.651)	117.77 (5.21)	106.21 (13.22)
AF-ELM	76.15 (8.22)	0.4543 (0.1213)	0.3687 (0.1786)	0.5399 (0.2142)	86.29 (17.71)	98.27 (16.76)
CC-ELM	74.41 (3.16)	0.7419 (0.0341)	0.8048 (0.0533)	0.6790 (0.0631)	76.12 (7.43)	127.24 (4.43)

The results presented in Table 2 confirm that the nature-based approach can yield more efficient results than the other models submitted to the test. That is due to the primary non-definition of the initial network neurons, a factor that makes the approach less complicated and more autonomous since it assesses the number of neurons according to the fuzzification technique. Thus, it is verified that the model used a much smaller number of fuzzy rules, obtained high indexes of hits in the database, and with a shorter execution time than the other hybrid models or techniques that use the pruning of neurons.

The graphs presented in the figures 5 and 6 show more simply the results obtained by the model proposed in this paper. Regarding accuracy, sensitivity, and specificity, it has similar results to another fuzzy neural network model. However, it is noteworthy that the number of final neurons and the execution time of the algorithm proposed in this paper are much smaller. Therefore, we were able to solve the problem with fewer fuzzy rules and more consistent execution time for everyday decision making. That proves that the model that works with the data nature is feasible to act precisely on this problem type.

Finally, to prove the behavior of the tests, statistical tests were used to assess the final results of the tests, with a primary focus on assessing the models' success

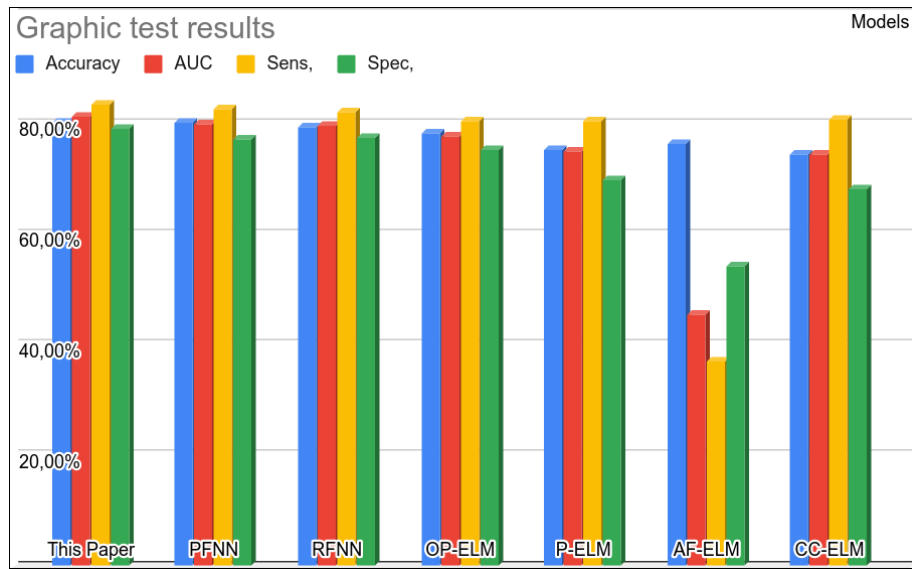


Fig. 5 Graphs on accuracy results.

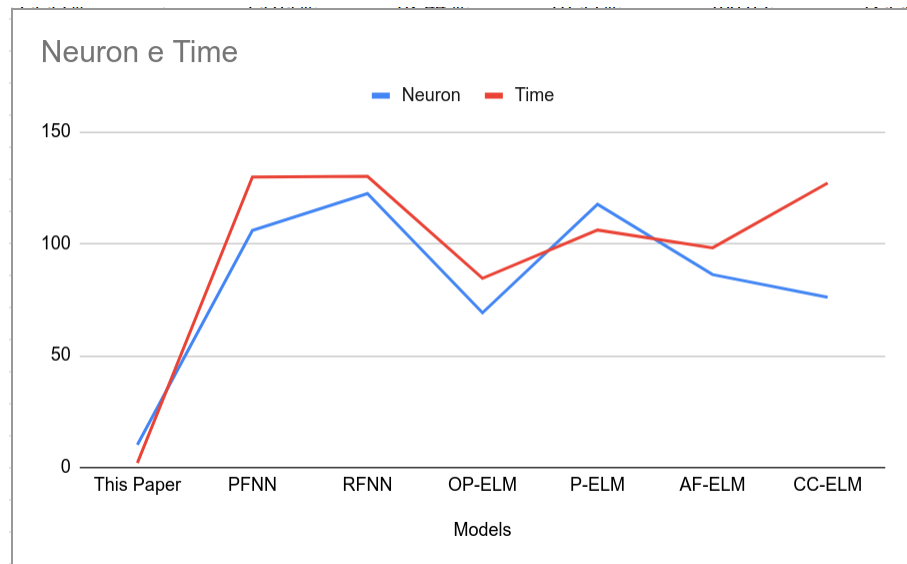


Fig. 6 Relationship between neurons and runtime by algorithm in the test.

rate. To assist in the tests, an online tool ⁴ capable of capturing data from a file and processing them according to the premises existing in the statistical tests was

⁴ <http://tec.citius.usc.es/stac/index.html>

used [61]. The analyzes will take place with the factors Accuracy and AUC since it is possible to identify sensitivity and specificity together. The number of neurons and the time have already been proven that the new model proposed has the best results. To assess the two groups mentioned above, the Non-parametric two groups test (Accuracy and AUC) was chosen. The Null hypothesis (H_0), in this case, is if the medians of the differences between the two group samples are equal. To this end, the Wilcoxon test [75] was applied, which assumes that the differences between samples are symmetrical concerning the median. Moreover, the final results are the negation of the null hypothesis (0.0179). As can be seen in Table 2, the results of the proposed model obtained better values for AUC and accuracy. Therefore, we can conclude that the model presents a satisfactory performance with the accuracy rate and the AUC, being different from the other models submitted to the test.

5 Conclusion

As demonstrated the proposed approach could reach the best result in terms of performance surpassing its state-of-the-art competitors for these challenging tasks. The fuzzification structure of the proposed approach used the data density, which is based on the nature of the data. Therefore, it provided a more compact model which is recommend for highly complex real-time situations as optical interconnections.

Therefore, the proposed approach due to its nonparametric nature and its pruning technique by F-scores could improve the results obtained by traditional approaches offering a less complex approach able to deal with data streams as the identification of optical network interconnections. The limitations of the method are in the choice of some model hyperparameters. Such factors can be addressed in the future by metaheuristics capable of finding the best values for the output of a model. Further work will investigate more complex problems as well as the adaption of the proposed model for different data patterns in a evolving manner.

Acknowledgment

The thanks of this work are destined to CEFET-MG and Faculdade UNA de Betim.

References

1. Abasikeles, I., Akay, M.F.: Using message combining and priority for improving the performance of distributed shared memory multiprocessor systems. In: PROCEEDINGS BOOK OF THE FOURTH INTERNATIONAL CONFERENCE ON, p. 212 (2013)
2. Açı, Ç.İ., Akay, M.F.: A hybrid congestion control algorithm for broadcast-based architectures with multiple input queues. *The Journal of Supercomputing* **71**(5), 1907–1931 (2015)

3. Akay, M.F., Abasıkeleş, I.: Predicting the performance measures of an optical distributed shared memory multiprocessor by using support vector regression. *Expert Systems with Applications* **37**(9), 6293–6301 (2010)
4. Akay, M.F., Aci, İ., Abut, F.: Predicting the performance measures of a 2-dimensional message passing multiprocessor architecture by using machine learning methods. *Neural Network World* **25**(3), 241–265 (2015)
5. Albert, A.: *Regression and the Moore-Penrose pseudoinverse*. Elsevier (1972)
6. Angelov, P., Yager, R.: Simplified fuzzy rule-based systems using non-parametric antecedents and relative data density. In: 2011 IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS), pp. 62–69. IEEE (2011)
7. Angelov, P., Yager, R.: A new type of simplified fuzzy rule-based system. *International Journal of General Systems* **41**(2), 163–185 (2012)
8. Angelov, P.P., Gu, X., Gutierrez, G., Iglesias, J.A., Sanchis, A.: Autonomous data density based clustering method. In: The bi-annual IEEE World Congress on Computational Intelligence (IEEE WCCI), pp. 2405–2413 (2016)
9. Baruah, R.D., Angelov, P.: Dec: Dynamically evolving clustering and its application to structure identification of evolving fuzzy models. *IEEE Transactions on Cybernetics* **44**(9), 1619–1631 (2014). DOI 10.1109/TCYB.2013.2291234
10. Batista, L.O., de Silva, G.A., Araújo, V.S., Araújo, V.J.S., Rezende, T.S., Junio, A., Guimarães, P.V.d.C.S.: Fuzzy neural networks to create an expert system for detecting attacks by sql injection. *International Journal of Forensic Computer Science* **13**(1), 8–21 (2018)
11. Bezdek, J.C., Ehrlich, R., Full, W.: Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences* **10**(2-3), 191–203 (1984)
12. Bordignon, F., Gomide, F.: Extreme learning for evolving hybrid neural networks. In: *Neural Networks (SBRN), 2012 Brazilian Symposium on*, pp. 196–201. IEEE (2012)
13. Bordignon, F., Gomide, F.: Uninorm based evolving neural networks and approximation capabilities. *Neurocomputing* **127**, 13–20 (2014)
14. Vitor de Campos Souza, P.: Pruning fuzzy neural networks based on unineuron for problems of classification of patterns. *Journal of Intelligent & Fuzzy Systems* **35**(2), 2597–2605 (2018)
15. de Campos Souza, P.V., Araujo, V.S., Guimaraes, A.J., Araujo, V.J.S., Rezende, T.S.: Method of pruning the hidden layer of the extreme learning machine based on correlation coefficient. In: 2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI), pp. 1–6 (2018). DOI 10.1109/LA-CCI.2018.8625247
16. de Campos Souza, P.V., Guimaraes, A.J.: Using fuzzy neural networks for improving the prediction of children with autism through mobile devices. In: 2018 IEEE Symposium on Computers and Communications (ISCC), pp. 01,086–01,089. IEEE (2018)
17. de Campos Souza, P.V., Guimaraes, A.J., Araújo, V.S., Rezende, T.S., Araújo, V.J.S.: Fuzzy neural networks based on fuzzy logic neurons regularized by resampling techniques and regularization theory for regression problems. *Inteligencia Artificial* **21**(62), 114–133 (2018)
18. de Campos Souza, P.V., Guimaraes, A.J., Araujo, V.S., Rezende, T.S., Araujo, V.J.S.: Incremental regularized data density-based clustering neural networks to aid in the construction of effort forecasting systems in software development. *Applied Intelligence* pp. 1–14 (2019)
19. de Campos Souza, P.V., Nunes, C.F.G., Guimares, A.J., Rezende, T.S., Araujo, V.S., Araujo, V.J.S.: Self-organized direction aware for regularized fuzzy neural networks. *Evolving Systems* pp. 1–15 (2019)
20. de Campos Souza, P.V., de Oliveira, P.F.A.: Regularized fuzzy neural networks based on null-neurons for problems of classification of patterns. In: 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), pp. 25–30. IEEE (2018)
21. de Campos Souza, P.V., Silva, G.R.L., Torres, L.C.B.: Uninorm based regularized fuzzy neural networks. In: 2018 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), pp. 1–8. IEEE (2018)
22. de Campos Souza, P.V., Torres, L.C.B.: Regularized fuzzy neural network based on or neuron for time series forecasting. In: *North American Fuzzy Information Processing Society Annual Conference*, pp. 13–23. Springer (2018)

23. de Campos Souza, P.V., Torres, L.C.B., Guimarães, A.J., Araujo, V.S.: Pulsar detection for wavelets soda and regularized fuzzy neural networks based on andneuron and robust activation function. *International Journal on Artificial Intelligence Tools* **28**(01), 1950,003 (2019)
24. Cavalieri, S., Di Stefano, A., Mirabella, O., Reitano, G.: On the design of a multiprocessor architecture for neural network simulation. In: 1991., *IEEE International Symposium on Circuits and Systems*, pp. 2538–2541. IEEE (1991)
25. Chatterjee, A., Pulasinghe, K., Watanabe, K., Izumi, K.: A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems. *IEEE Transactions on Industrial Electronics* **52**(6), 1478–1489 (2005)
26. Chen, Y.W., Lin, C.J.: Combining svms with various feature selection strategies. In: *Feature extraction*, pp. 315–324. Springer (2006)
27. Elhadeif, M.: A modified hopfield neural network for diagnosing comparison-based multiprocessor systems using partial syndromes. In: 2011 *IEEE 17th International Conference on Parallel and Distributed Systems*, pp. 646–653. IEEE (2011)
28. Gao, J., Wang, Z., Yang, Y., Zhang, W., Tao, C., Guan, J., Rao, N.: A novel approach for lie detection based on f-score and extreme learning machine. *PLoS one* **8**(6), e64,704 (2013)
29. Giles, C.L., Goudreau, M.W.: Routing in optical multistage interconnection networks: A neural network solution. *Journal of lightwave technology* **13**(6), 1111–1115 (1995)
30. Golub, G.H., Heath, M., Wahba, G.: Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics* **21**(2), 215–223 (1979)
31. Gu, X., Angelov, P.P.: Autonomous data-driven clustering for live data stream. In: *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, pp. 001,128–001,135. IEEE (2016)
32. Gu, X., Angelov, P.P., Gutierrez, G., Iglesias, J.A., Sanchis, A.: Parallel computing teda for high frequency streaming data clustering. In: *INNS Conference on Big Data*, pp. 238–253. Springer (2016)
33. Gu, X., Angelov, P.P., Príncipe, J.C.: A method for autonomous data partitioning. *Information Sciences* (2018)
34. Guimarães, A.J., Araujo, V.J.S., de Campos Souza, P.V., Araujo, V.S., Rezende, T.S.: Using fuzzy neural networks to the prediction of improvement in expert systems for treatment of immunotherapy. In: *Ibero-American Conference on Artificial Intelligence*, pp. 229–240. Springer (2018)
35. Guimaraes, A.J., Arajo, V.J., de Oliveira Batista, L., Souza, P.V.C., Araujo, V., Rezende, T.S.: Using fuzzy neural networks to improve prediction of expert systems for detection of breast cancer. *Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC)* pp. 799–810 (2018). DOI 10.5753/eniac.2018.4468. URL <http://portaldeconteudo.sbc.org.br/index.php/eniac/article/view/4468>
36. Halim, Z., Khattak, J.H.: Density-based clustering of big probabilistic graphs. *Evolving Systems* pp. 1–18 (2018)
37. Hassan, T., Bajwa, I.S., Hassan, S.: Prediction of terrorist activities by using unsupervised learning techniques. *Journal of Applied and Emerging Sciences* **6**(2), pp56–60 (2016)
38. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. *Neurocomputing* **70**(1-3), 489–501 (2006)
39. Iglesias, J.A., Sanchis, A.: Parallel computing teda for high frequency streaming data clustering. In: *Advances in Big Data: Proceedings of the 2nd INNS Conference on Big Data*, October 23-25, 2016, Thessaloniki, Greece, vol. 529, p. 238. Springer (2016)
40. Jang, J.S.: Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics* **23**(3), 665–685 (1993)
41. Johnson, K.M., Surette, M.R., Shamir, J.: Optical interconnection network using polarization-based ferroelectric liquid crystal gates. *Applied Optics* **27**(9), 1727–1733 (1988)
42. Junio Guimares, A., Vitor de Campos Souza, P., Jonathan Silva Arajo, V., Silva Rezende, T., Souza Arajo, V.: Pruning fuzzy neural network applied to the construction of expert systems to aid in the diagnosis of the treatment of cryotherapy and immunotherapy. *Big Data and Cognitive Computing* **3**(2) (2019). DOI 10.3390/bdcc3020022. URL <https://www.mdpi.com/2504-2289/3/2/22>

43. Kangin, D., Angelov, P., Iglesias, J.A.: Autonomously evolving classifier tedaclass. *Information Sciences* **366**, 1 – 11 (2016). DOI <https://doi.org/10.1016/j.ins.2016.05.012>. URL <http://www.sciencedirect.com/science/article/pii/S002002551630336X>
44. Katsinis, C.: Merging, sorting and matrix operations on the some-bus multiprocessor architecture. *Future Generation Computer Systems* **20**(4), 643–661 (2004)
45. Kim, W., Oh, S., Seo, K., Pedrycz, W.: A design of fcm-based interval type-2 fuzzy neural network classifier with the aid of pso. In: 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), pp. 1209–1214 (2013). DOI 10.1109/IFSA-NAFIPS.2013.6608573
46. Kotilainen, P., Saarinen, J., Kaski, K.: Neural network computation in a parallel multiprocessor architecture. In: Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), vol. 2, pp. 1979–1982. IEEE (1993)
47. Lemos, A., Caminhas, W., Gomide, F.: New uninorm-based neuron model and fuzzy neural networks. In: Fuzzy Information Processing Society (NAFIPS), 2010 Annual Meeting of the North American, pp. 1–6. IEEE (2010)
48. Lemos, A., Kreinovich, V., Caminhas, W., Gomide, F.: Universal approximation with uninorm-based fuzzy neural networks. In: Fuzzy Information Processing Society (NAFIPS), 2011 Annual Meeting of the North American, pp. 1–6. IEEE (2011)
49. Lemos, A.P., Caminhas, W., Gomide, F.: A fast learning algorithm for uninorm-based fuzzy neural networks. In: Fuzzy Information Processing Society (NAFIPS), 2012 Annual Meeting of the North American, pp. 1–6. IEEE (2012)
50. Louri, A., Sung, H.: An optical multi-mesh hypercube: a scalable optical interconnection network for massively parallel computing. *Journal of Lightwave Technology* **12**(4), 704–716 (1994)
51. Lu, C.H., Tsai, C.C.: Generalized predictive control using recurrent fuzzy neural networks for industrial processes. *Journal of process control* **17**(1), 83–92 (2007)
52. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* **5**(4), 115–133 (1943)
53. Miche, Y., Sorjamaa, A., Bas, P., Simula, O., Jutten, C., Lendasse, A.: Op-elm: optimally pruned extreme learning machine. *IEEE transactions on neural networks* **21**(1), 158–162 (2010)
54. Mohamad, M., Saman, M.Y.M., Hitam, M.S.: A framework for multiprocessor neural networks systems. In: 2012 International Conference on ICT Convergence (ICTC), pp. 44–48. IEEE (2012)
55. Na, J.W.: A parallel optical computer architecture for large database and knowledge based systems. In: International Conference on Computational and Information Science, pp. 934–939. Springer (2004)
56. Pedrycz, W.: *Fuzzy control and fuzzy systems* (2nd. Research Studies Press Ltd. (1993)
57. Pedrycz, W., Gomide, F.: *An introduction to fuzzy sets: analysis and design*. Mit Press (1998)
58. Pinto, D., Lemos, A.P., Braga, A.P., Horizonte, B., Gerais-Brazil, M.: An affinity matrix approach for structure selection of extreme learning machines. In: Proceedings, p. 343. Presses universitaires de Louvain (2015)
59. Plesser, H.E., Eppler, J.M., Morrison, A., Diesmann, M., Gewaltig, M.O.: Efficient parallel simulation of large-scale neuronal networks on clusters of multiprocessor computers. In: European Conference on Parallel Processing, pp. 672–681. Springer (2007)
60. Rast, M.: Improving fuzzy neural networks using input parameter training. In: 1998 Conference of the North American Fuzzy Information Processing Society - NAFIPS (Cat. No.98TH8353), pp. 55–58 (1998). DOI 10.1109/NAFIPS.1998.715529
61. Rodríguez-Fdez, I., Canosa, A., Mucientes, M., Bugarín, A.: Stac: a web platform for the comparison of algorithms using statistical tests. In: 2015 IEEE international conference on fuzzy systems (FUZZ-IEEE), pp. 1–8. IEEE (2015)
62. Rong, H.J., Ong, Y.S., Tan, A.H., Zhu, Z.: A fast pruned-extreme learning machine for classification problem. *Neurocomputing* **72**(1-3), 359–366 (2008)
63. Ross, T.J.: Membership functions, fuzzification and defuzzification. In: *Fuzzy systems in medicine*, pp. 48–77. Springer (2000)

64. Sampsell, J.B., McDonald, T.G.: Optical fiber interconnection network including spatial light modulator (1989). US Patent 4,856,863
65. Saw, J.G., Yang, M.C., Mo, T.C.: Chebyshev inequality with estimated mean and variance. *The American Statistician* **38**(2), 130–132 (1984)
66. Similä, T., Tikka, J.: Multiresponse sparse regression with application to multidimensional scaling. In: *International Conference on Artificial Neural Networks*, pp. 97–102. Springer (2005)
67. Škrjanc, I., Blažič, S., Lughofer, E., Dovžan, D.: Inner matrix norms in evolving cauchy possibilistic clustering for classification and regression from data streams. *Information Sciences* **478**, 540–563 (2019)
68. Souza, P.V.C.: Regularized fuzzy neural networks for pattern classification problems. *International Journal of Applied Engineering Research* **13**(5), 2985–2991 (2018)
69. Souza, P.V.C., dos Reis, A.G., Marques, G.R.R., Guimaraes, A.J., Araujo, V.J.S., Araujo, V.S., Rezende, T.S., Batista, L.O., da Silva, G.A.: Using hybrid systems in the construction of expert systems in the identification of cognitive and motor problems in children and young people. In: *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–6 (2019). DOI 10.1109/FUZZ-IEEE.2019.8858906
70. Souza, P.V.C., Guimaraes, A.J., Araujo, V.S., Batista, L.O., Rezende, T.S.: An Interpretable Machine Learning Model for Human Fall Detection Systems Using Hybrid Intelligent Models, pp. 181–205. Springer International Publishing, Cham (2020). URL https://doi.org/10.1007/978-3-030-38748-8_8
71. Souza, P.V.d.C., Guimaraes, A.J., Araujo, V.S., Rezende, T.S., Araujo, V.J.S.: Regularized fuzzy neural networks to aid effort forecasting in the construction and software development. *International Journal of Artificial Intelligence Applications (IJAA)* (2018)
72. Toosi, A.N., Kahani, M.: A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. *Computer communications* **30**(10), 2201–2212 (2007)
73. Wang, Z., Zhao, W., Hu, X.: Analysis of prediction model of failure depth of mine floor based on fuzzy neural network. *Geotechnical and Geological Engineering* **37**(1), 71–76 (2019)
74. Weiss, Y.: Segmentation using eigenvectors: a unifying view. In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2, pp. 975–982. IEEE (1999)
75. Wilcoxon, F.: Individual comparisons by ranking methods. In: *Breakthroughs in statistics*, pp. 196–202. Springer (1992)
76. Yager, R.R., Rybalov, A.: Uninorm aggregation operators. *Fuzzy sets and systems* **80**(1), 111–120 (1996)
77. Yamamoto, J., Fujiwara, T., Komeda, T., Kamei, T., Hanawa, T., Amano, H.: Performance evaluation of snail: A multiprocessor based on the simple serial synchronized multistage interconnection network architecture. *Parallel Computing* **25**(9), 1081–1103 (1999)
78. Zadeh, L.A.: A fuzzy-algorithmic approach to the definition of complex or imprecise concepts. In: *Systems Theory in the Social Sciences*, pp. 202–282. Springer (1976)
79. Zayid, E.I.M., Akay, M.F.: Predicting the performance measures of a message-passing multiprocessor architecture using artificial neural networks. *Neural Computing and Applications* **23**(7-8), 2481–2491 (2013)