applied sciences

MDPI

Article

Fuzzy Neural Network with Ordered Fuzzy Numbers for Life Quality Technologies

Łukasz Apiecionek [1,*], Rafał Moś [2] and Dawid Ewald [1]

[1] Department of Computer Science, Kazimierz Wielki Univesrity, 85-064 Bydgoszcz, Poland
[2] IST Software Sp. z o.o. Aleje Jerozolimskie 162A, 02-342 Warszawa, Poland
* Correspondence: lapiecionek@ukw.edu.pl

Abstract: The general goal of the research in this article is to devise an artificial neural network that requires less computational power than an ordinary one for assessing overall life satisfaction—a term often referred to as quality of life (QoL). The development of the mentioned ANN was possible due to the application of fuzzy logic, especially ordered fuzzy numbers (OFN). Research on the appliance of OFN aims at different issues such as the detection of an attack on a computer network, the anticipation of server load, management of multiplexing of data transmission paths, or transmission error rate forecasting that allows the improvement of the quality of life. It occurs due to, for instance, reduced energy demand, savings through better data transmission, and the distribution of computers' power used in the cloud. Finally, the application of OFN on single neurons of a deep ANN allows achieving a network that is able to solve the same problem as a normal network, but with a lower number of neurons. Such networks in the future may be implemented easier in small solutions, such as solutions for the Internet of Things to improve the quality of human life. This approach is unique and has no equivalent in the literature. Due to the application of OFN in an ANN, fewer requirements for network architecture were needed to solve the same problems, and as a result, there is less demand for processor power and RAM.

Keywords: fuzzy logic; quality of life; life satisfaction; artificial neural network

Citation: Apiecione, Ł.; Moś, R.; Ewald, D. Fuzzy Neural Network with Ordered Fuzzy Numbers for Life Quality Technologies. *Appl. Sci.* **2023**, *13*, 3487. https://doi.org/10.3390/app13063487

Academic Editors: Piotr Prokopowicz, Katarzyna Węgrzyn-Wolska and Maciej Piechowiak

Received: 30 December 2022
Revised: 29 January 2023
Accepted: 14 February 2023
Published: 9 March 2023

1. Introduction

At the present time, many IT solutions are being implemented with the aim of improving the quality of social life. These solutions for steering processes and data gathering are used to facilitate decision making or even making the decisions instead of humans, but at a significantly faster rate. It contributes to the reduction of electric power consumption and better use of resources, and, above all, allows users to save time. The foregoing solutions are implemented under the concept of the Internet of Things. For instance, having connected home devices, smart homes are then constructed. As a result, from an economical point of view, money is saved; from the psychological perspective, people have better life satisfaction. The main contribution of this paper is the proposal of the solution to the scientific problem. These days, artificial intelligence, especially deep artificial neural networks, are used in many tasks. Unfortunately, these solutions cannot always be implemented due to their requirements in terms of processor power or RAM. The authors of this paper hypothesize that an artificial neural network can be constructed by means of fuzzy logic, i.e., ordered fuzzy numbers (OFN) in its neurons. In this paper, a short introduction to OFN is presented, with the examples of their applications, such as the detection of attacks on computer networks, anticipation of server load, management of multiplexing connections, or forecasting transmission error rate. The results of these studies led to the conclusion that OFN contributes to faster responses of developed algorithms, and thus improves their quality. Moreover, the increase in the quality of solutions impacts the quality of users' lives positively. Therefore, we conducted research on our hypothesis, which is based on a comparison of the implemented network with an ordinary one. Today, millions of computers with an infinite variety of programs are being used in everyday life, which means the occurrence of numerous lines of code with potential mistakes. These mistakes may generate two kinds of issues, including working and security problems. First, stopping the work of computers may result in preventing the completion of tasks performed by people. The second issue is connected with security. Some mistakes in the code may allow hackers to gain access to the system without permission and take over the data. That is why devices and programs should be monitored to verify the proper operation. When we consider the variety of devices, including small ones with low computational power, it is necessary to have some algorithms for simple access and control. These small systems provide no possibility for installing solutions that require high computational power and lots of memory. To solve the problem of security and access that affects the user quality of life, the authors propose a fuzzy neural network with ordered fuzzy numbers, which does not require high computational power and can be used to detect and predict problems with the devices and programs.

2. Current Fuzzy Neural Networks Review

A lot of research dealing with fuzzy neural networks has been conducted. Some use the McCulloch–Pitts model of a neuron and change the activation function to the fuzzy function, which makes the neuron more general [1]. Some authors [2] change the neuron architecture to use triangular fuzzy weights. This approach uses fuzzy vectors and real numbers as inputs. The outputs of such networks are fuzzy vectors. Such solutions are possible by using the extension principle of Zadeh. Most papers use fuzzy weights and signals in the network [3]. Paulo Vitor et al. [4] propose a fuzzy neuron using the idea of null-uninorm, called null-unineuron. It allows building an evolving neuro-fuzzy model. It leads to the extraction of advanced fuzzy rules and/or connections of antecedents. The authors used a three-layer model with a weighted fuzzification approach based on incremental data partitioning concepts for knowledge extraction. The main use of fuzzy networks is to build controllers that will work better than ordinary ones [5–16]. There are some papers in which the authors describe how to use a fuzzy network algorithm for labeling images [17]. Wane et al. [18] provide information on how to use an unsupervised multilayer fuzzy neural network, which could be used for image clustering. The authors extend fuzzy systems to unsupervised tasks by proposing manifold representation. There are also some works on using fuzzy neural networks for medical image recognition. The authors defined the problem that medical images can often blur object boundaries, and it is impossible to describe the uncertainty of these blurred boundaries using traditional neural network models. They propose a new fuzzy metric to characterize the uncertainty of pixels and design a fuzzy hierarchical fusion attention neural network based on multiscale guided learning. The authors also proposed data transformation to convert the image into a fuzzy domain, which allows using fuzzy rules to deal with the uncertainty of the pixels. Then, the results are connected with the result of the convolution in the neural network [19]. Sami Ben Jabeur and Vanessa Serret proposed a combined method for bankruptcy prediction based on fuzzy set qualitative comparative analysis and convolutional neural networks [20]. Some authors propose using an adaptive neural network for solving some real problems, e.g., the problem of the Takagi–Sugeno fuzzy model-based state estimation and sliding mode control for nonlinear systems through an adaptive neural network. In this proposition, the system parameters follow the Markovian switching rules [21]. One of the reasons for using a fuzzy theorem with a neural network is to learn the neural network faster [22]. Liu et al. [23] present a fixed-time synchronization for fuzzy inertial neural networks with time-varying coefficients and time delays. The authors propose their criteria to ensure fixed-time synchronization for the drive–response fuzzy inertial neural networks via state feedback control methods. Gong et al. propose a delay-independent nonlinear fuzzy control for solving the synchronization problem for T-S fuzzy memristive neural networks with time delay [24]. In [25], a novel adaptive fuzzy neural network-aided progressive

Gaussian filter is proposed to solve the problem of uncertainty of prior information in nonlinear filtering, which could be used in navigation and positioning systems. In [26], a deep fuzzy neural network is used for controlling the integrity of the reactor core, reactor coolant system, or containment. In [27], the authors propose fuzzy reinforced polynomial neural networks. The results provided show that the proposed architecture achieves at most an accuracy 43.6% higher in comparison with the accuracy produced by some classical models reported in the literature. In [28], the authors propose an architecture of a fuzzy neural network based on self-organizing direction-aware data using stochastic processes. It also achieves better results than ordinary neural networks. In [29], the authors proposed a novel multi-functional recurrent fuzzy neural network. Its architecture consists of two fuzzy neural networks with Takagi–Sugeno–Kang fuzzy rules. This network is used for calculating the output and the system's state. The feedback connection between these networks is also provided. In [30], the authors describe problems of traditional neural networks with low accuracy and convergence. To solve these problems, the authors proposed a fuzzy deep wavelet neural network inversion method and some hybrid learning algorithm. In [31], the author proposes a family of neural network operators of fuzzy n-cell number valued functions with sigmoidal functions used for neuron activation. In [32], the authors describe a problem for small solution systems that are sensitive to the initial weights. This also requires high computation. That is why the authors propose new learning algorithms that are not sensitive to the provided problems.

In [33], the author proposes a family of multivariate fuzzy neural network interpolation operators activated by sigmoidal functions. In [34], the authors provide some results of using a fuzzy network with different activation functions, which give different results of course. In [35], the authors also propose hierarchical polynomial-based fuzzy neural networks that could improve the prediction accuracy of the model. Finally, there are also some papers that present a review of fuzzy systems and artificial neural networks as [36]. In [37], the author proposes the adaptive network-based fuzzy interface system (ANFIS), which uses logical rules and connections in the network layer, not only between the close connected layers. In this paper, the authors propose a different solution that could not be found in the literature. We use an ordinary deep neural network and change the neuron into a fuzzy neuron with the Ordered Fuzzy Numbers arithmetic. It lets us to use not only the triangle form of fuzzy numbers (as in the ANFIS model), but also the trapezoid one.

3. Ordered Fuzzy Numbers

When the precise amount of something could not be given, and is presented by such words as "little", "more", "some", we could use the term. The branch of artificial intelligence that deals with this problem is called fuzzy logic. It was described by Lofti A. Zadeh in the paper "Fuzzy sets" in Information and Control journal in 1965 [38]. In this paper, he defines the term of a fuzzy set. This fuzzy set allows us to describe imprecise data. These data are presented by means of values from the interval from 0 to 1. A number that is assigned to the set defines its degree of membership in this set. The author used a three-valued logic defined by Jan Łukasiewicz [39]. The use of this theory, the L-R representation of fuzzy numbers, was presented by D. Dubois and H. Prade [40–42]. Another method for fuzzy set presentation was developed by W. Kosiński [43] and his team. This method is called Ordered Fuzzy Numbers. The advantage of this presentation is that it allows us to link the change of trend to a fuzzy number. It gives more opportunities for arithmetic operation on fuzzy numbers, which was described in the next publications [44–47]. The following definitions and equations provide basic information about Ordered Fuzzy Numbers, which will be used in the proposed fuzzy network. It is worth noting that this arithmetic is very simple, which means that it may be calculated in a fast and easy way.

Definition 1. *A is an ordered fuzzy number A when it is an ordered pair of functions*

$$A = (x_{up}, x_{down}) \tag{1}$$

where $x_{up}, x_{down} : [0, 1] \longrightarrow R$ *are continuous functions.*

Figure 1 presents the respective parts of the functions called: up and $down$. The OFN presentation of fuzzy numbers is presented in Figure 2.

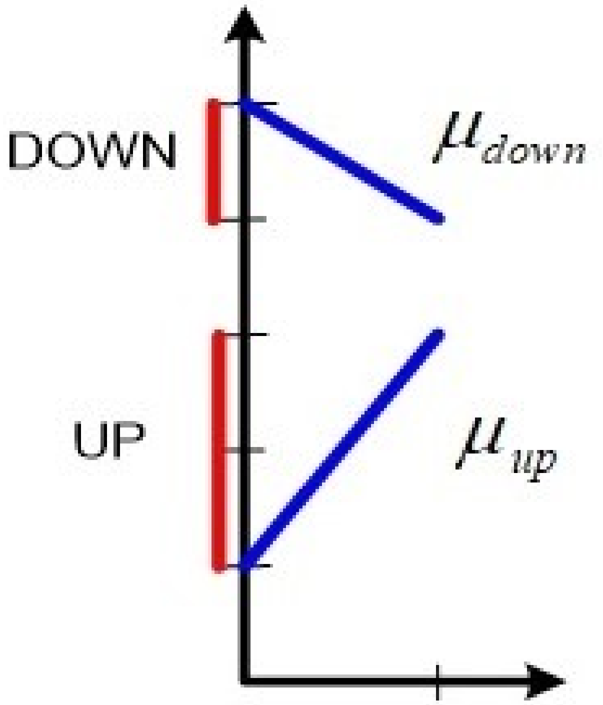

Figure 1. The functions up and down in Ordered Fuzzy Number representation.

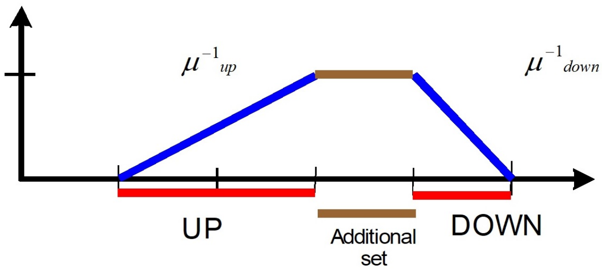

Figure 2. A fuzzy representation of Ordered Fuzzy Number.

As it could be noticed, the continuity of the two parts (UP and DOWN) shows that they are limited in the following values: $UP = (l_A, 1_A^-)$ and $DOWN = (1_A^+, p_A)$. When both functions within the OFN are continuous, then their inverse functions x_{UP}^{-1} and x_{down}^{-1} are defined by their limits UP and DOWN. This condition provides the following equations:

$$l_A : x_{UP}(0), 1_A^- : x_{UP}(1), 1_A^+ : x_{down}(1), p_A : x_{down}(0) \tag{2}$$

If a constant function equal to 1 is added within the interval $[1_A^-, 1_A^+]$ are defined by their limits UP and DOWN. On this assumption, the following equations are valid:

$$l_A : x_{UP}(0), 1_A^- : x_{UP}(1), 1_A^+ : x_{down}(1), p_A : x_{down}(0) \tag{3}$$

when a constant function (equal to 1) is added within the interval $[1_A^-, 1_A^+]$, then the result is functions UP and DOWN in one range (Figure 2, where:$\mu_{down} = x_{down}, \mu_{up} = x_{up}$). This result could be treated as a carrier. Then, the function of membership in the fuzzy set is defined on the R set by the following equations:

$$\mu_A(x) = 0 \text{ for } x \notin [l_A, p_A]\ ,$$
$$\mu_A(x) = x_{UP}^{-1}\ for\ x \in UP, \tag{4}$$
$$\mu_A(x) = x_{DOWN}^{-1} \text{ for } x \in DOWN\ ,$$

The fuzzy set defined in the OFN way acquires an additional parameter order, whereas the following interval is the carrier:

$$UP \cup [1_A^-, 1_A^+] \cup DOWN \tag{5}$$

The following conditions defined the limit values for UP and DOWN functions in such way:

$$\begin{aligned} \mu_A(l_A) &= 0 \\ \mu_A(l_A^-) &= 1 \\ \mu_A(l_A^+) &= 1 \\ \mu_A(p_A) &= 0 \end{aligned} \tag{6}$$

OFN can be described by four real following values:

$$A = (l_A, l_A^-, l_A^+, p_A)$$

Figure 3 presents an example of OFN, including their characteristic points in two orders: positive and negative.

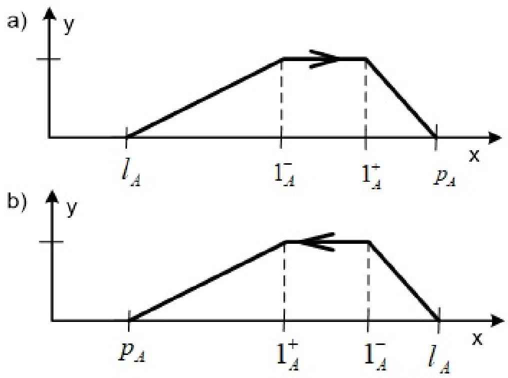

Figure 3. Fuzzy number that is ordered (**a**) positively or (**b**) negatively.

Functions f_A, g_A correspond to parts $up_A, down_A \subseteq R^2$, which gives:

$$up_A = \{(f_A(y), y) : y \in [0,1]$$
$$down_A = \{(g_A(y), y) : y \in [0,1] \tag{7}$$

Orientation corresponds to the order of graphs f_A and g_A.

Definition 2. *A membership function of an OFN A is the function $\mu_A : R \longrightarrow [0,1]$ defined for $x \in R$ in the following way:*
if

$$x \notin supp_A \Rightarrow \mu_A(x) = 0$$
$$x \in (1_A^-, 1_A^+) \Rightarrow \mu_A(x) = 1 \tag{8}$$
$$x \in supp_A \wedge x \notin (1_A^-, 1_A^+) \Rightarrow \mu_A(x) = max(f_A^{-1}(x), g_A^{-1}(x))$$

Such a definition of the membership function can be applied in the OFN notation for setting control rules in the same way as in classic fuzzy numbers.

Definition 3. *The replacement of the part UP (function f_A) with the part DOWN (function g_A) is a reversal of the OFN. This could be defined as follows:*

$$B = A \mid^{-} \Leftrightarrow g_B = f_A \wedge f_B = g_A \tag{9}$$

where:

- *A is OFN (described by the pair of functions (f_A, g_A)),*
- *B is a result of the operation of reversal of the OFN A,*
- *sign "$|^{-}$" is a symbol that describes a reversed orientation of OFN,*
- *The result obtained in this operation is called a reversed OFN or a reversed orientation number.*

A set of key points of OFN is very important for number interpretations. This set is defined as follows:

$$[f(0), f(1), g(1), g(0)]$$

The basic arithmetic operations for A = (f, g) and B = (e, h) could be defined as another pair of affine functions, according to the following formulas:

- addition $A + B = (f + e, g + h) = C$

$$C \longrightarrow [f(0) + e(0), f(1) + e(1), g(1) + h(1), g(0) + h(0)] \tag{10}$$

- scalar multiplication $C = \lambda A = (\lambda f, \lambda g)$,

$$C \longrightarrow [\lambda f(0), \lambda f(1), \lambda g(1), \lambda g(0)] \tag{11}$$

- subtraction $A - B = (f - e, g - h) = C$,

$$C \longrightarrow [f(0) - e(0), f(1) - e(1), g(1) - h(1), g(0) - h(0)] \tag{12}$$

- multiplication $A \times B = (f \times e, g \times h) = C$,

$$C \longrightarrow [f(0) \times e(0), f(1) \times e(1), g(1) \times h(1), g(0) \times h(0)] \tag{13}$$

Ordered Fuzzy Numbers in practice
Ordered Fuzzy Numbers are applicable in many everyday life aspects, such as:

- Smart home systems,
- Data transmission control,
- Servers management in cloud systems,
- Detection of attacks on IT systems.

The application of the OFN solution accelerates the control of the heating system. The tests performed in the climatic chamber have confirmed that the results of data analysis with fuzzy logic with OFN algebra, flowing through the controller speeds up reaction so that the temperature can be kept closer to the set one [48].

In the data transmission that uses multiplexing of data transmission paths in the technology Multi Path Transmission Control, the application of an algorithm to predict the error rate in a given path, using OFNs allowed for a faster response and the choice of the different transmission path. With such solution, it is possible to decrease the number of data retransmissions in the network. Following up, there is the possibility to save electric power–transmitting. The same data with a lower retransmissions number saves the power required to send the data. This algorithm can be applied in today's networks, including even 5G networks [49].

Similar possibilities of saving energy necessary in everyday life can be achieved by using algorithms for predicting the demand for virtual machines in cloud systems. The proposed cloud algorithm to forecast servers use with the OFN's application was characterized by the possibility to shut down machines faster at a time when the need for computing power to handle incoming connections is decreased [50]. The faster shutdown of servers also means lower power consumption.

One of the present Internet problems is the Denial of Service attack type, including Distributed Denial of Service [51]. Such an attack can lock each system available in the network, and lead to an immediate decline in the quality of life. The lock of bank systems can exclude the payments by bank cards (credit, debt) and paralyze trade. These days, such attacks are elements of hybrid wars. Therefore, the quick detection and neutralization of attacks are essential. The algorithms using OFN in the fast detection of predicted attacks can be helpful in the comparison to historical values. The implementation of such algorithms allows edge devices to react faster and block unwanted network traffic [52]. As it can be noted, in many aspects affecting the quality of life, the use of algorithms using OFN logic has an impact on faster reactions, resulting in tangible benefits.

4. Fuzzy Neural Networks

Currently, Artificial Neural Networks, including Deep Networks, are widely used solutions in the field of Artificial Intelligence. The neuron is the single element to build these networks. There are many neuron implementations. One of them is the McCullocha-Pitts neuron presented in Figure 4.

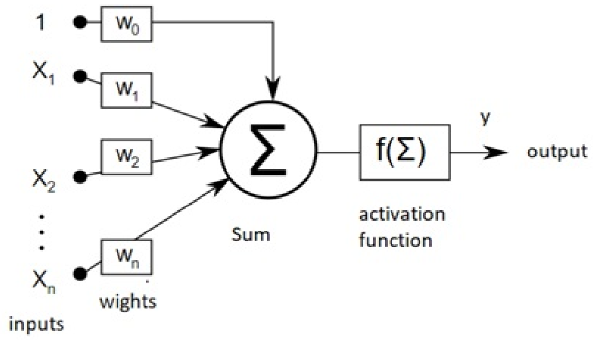

Figure 4. McCullotch-Pitts neuron.

This neuron model assumes the possibility to have many inputs and one output. Each input is related (tied) to weights, which are real numbers. The value at the neuron output is calculated in the following steps:

Step 1: the value of the product's sum of input values and the weights assigned to them is calculated:

$$s = w_0 + \sum_{n}^{i=1} x_i w_i \tag{14}$$

Step 2: the value of the activation function for the calculated sum is calculated on the output, and the value of the activation function f (s) for the calculated sum is given:

$$y = f(x) \tag{15}$$

Then, with the help of such neurons, their network is built, which may have a different number of inputs, layers of neurons and a different number of outputs. In the literature, one can find publications containing an overview of fuzzy networks [36]. However, none of them use the Ordered Fuzzy Numbers arithmetic. In order to build such a fuzzy network, a neuron, according to McCulloch-Pitts scheme, was proposed. However, in this case, the inputs, weights, and outputs are numbers in OFN notation. Then the arithmetic in OFN notation is carried out in the following steps.

Step 1: the value of the product's sum of input values and the weights assigned to them is calculated:

$$s = W_0 + \sum_{n}^{i=1} X_i W_i \tag{16}$$

Step 2: the arithmetic in OFN notation is calculated in the following steps.

$$Y = f(X) \tag{17}$$

where the individual components of S, W, X and Y are values in OFN notation. This approach requires:

- Fuzzification of the input data to the network,
- Defuzzification of the output data from the network,
- Development of network training algorithms that operate in OFN arithmetic.

The scheme of operation of the developed deep network is presented in Figure 5.

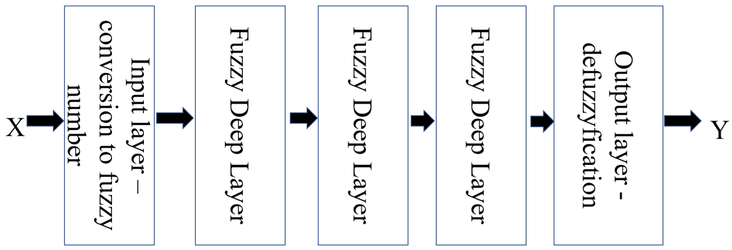

Figure 5. Fuzzy network with OFN.

A number of methods [53] has already been developed for fuzzification and defuzzification OFNs, and it was not necessary to develop new ones. However, it was crucial to develop network training algorithms.

During the works on the training algorithms of the constructed network, 4 (four) training algorithms were developed and it was checked which one would achieve the best results in the shortest time, i.e., it would be characterized by the best anomaly detection ratio in terms of the time necessary to train the network.

There were the followings algorithms:

- Algorithm 1 with back error propagation in variant 1 $\rightarrow$ fuzzification of input data, result calculation, fuzzification of correct output value, value comparison with calculated result, and percentage distribution of back error propagation (Figure 6).

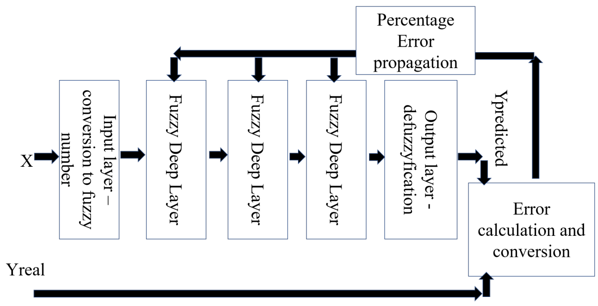

Figure 6. First learning algorithm.

- Algorithm 2 with back error propagation in variant 2 $\rightarrow$ fuzzification of input data, result calculation, fuzzification of correct output value, defuzzification of the result, value comparison with the result, error calculation, error fuzzification, and percentage distribution of back error propagation (Figure 7).

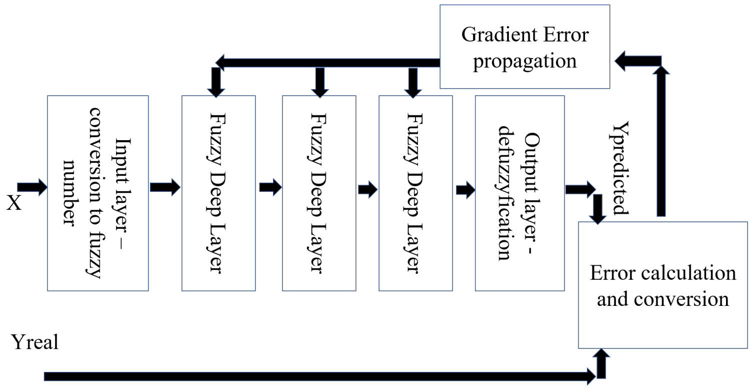

Figure 7. Second learning algorithm.

- Algorithm 3 with back error propagation in variant 3 → fuzzification of input data, result calculation, fuzzification of the correct output value, comparison of the value with the calculated result, and back error gradient propagation(Figure 8).

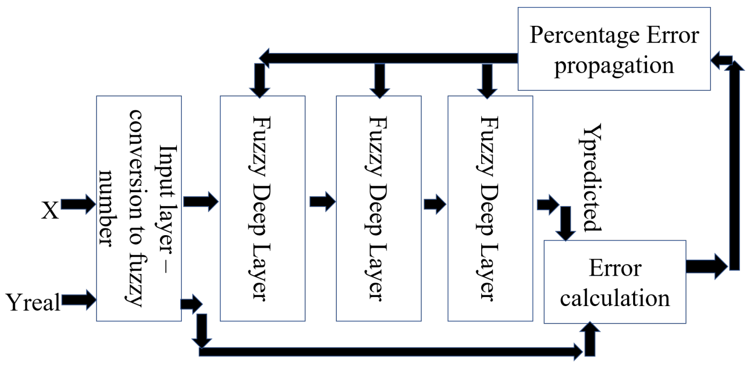

Figure 8. Third learning algorithm.

- Algorithm 4 with back error propagation in variant 4 → fuzzification of input data, result calculation, fuzzification of the correct output value, defuzzification of result, comparison of the value with result, error fuzzification, and back error gradient propagation (Figure 9).

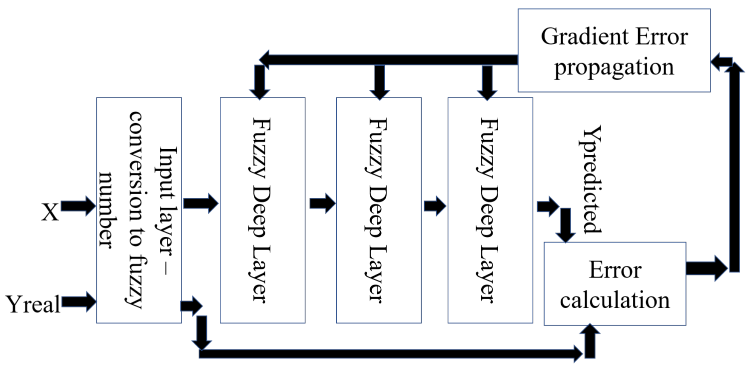

Figure 9. Fourth learning algorithm.

After the algorithms were coded, research on the effectiveness of network training and the quality of the acquired results (event prediction) started. For this purpose, based on the prepared data training files, the chosen anomalies were selected (7 types) and different data structures were prepared for them. The training trials of different net sizes (number of neurons in layers) have been performed. The correctness of the net implementation has been noted based on:

- Increase the efficiency (accuracy) of training for increase in the number of epochs,

- Extension of training time along with the increase in the number of epochs.

The tests have been performed for a minimum of 25 trials, for each algorithm. Then the average results for each algorithm were calculated. Based on the assessment of the net training effectiveness, algorithm 4 was selected. It implements the full idea of the training algorithm with back error propagation and distribution of error, using a gradient in the weights of neurons in the network. This algorithm acquired the highest speed of network training while keeping the same training efficiency. In the frame of comparison, experiments consisting of training the deep network for the same input data have been performed, comparing the effectiveness of the results for the same number of learning epochs. Algorithm 4 acquired the best learning results for the maintained speed.

5. Network Comparison

In the frame of network operation scrutiny, its operation was compared with the deep network. The idea of the test was to check what size of the network will be necessary to solve the same problem. As a research problem, the detection of the iris flower was chosen (Iris) [54]. The test set consisted of the description of the iris flowers using four attributes. The database contained 150 samples and has been limited to two flower classes. The structures of the tested networks were as follows: Deep Neural Network 1: input layer, four neurons; deep layer, eight neurons; and output layer, two neurons.

- Fuzzy Neural Network with OFN 1: input layer, 4 neurons; deep layer, 1 neuron; output layer, 2 neurons,
- Fuzzy Network with OFN 2: input layer, 4 neurons; deep layer, 3 neurons; output layer, 2 neurons,
- Fuzzy Network with OFN 3: input layer, 4 neurons; deep layer, 4 neurons; output layer, 2 neurons,
- Fuzzy Network with OFN 4: input layer, 4 neurons; deep layer, 8 neurons; output layer, 2 neurons.

Overall, 500 epochs were used when training the regular network, and 550 when training the fuzzy network. Iris flower detection error value is presented in Table 1. The conducted test was to recognize the flower. So, the possible results were 0 or 1. Finally, using the test data, the amount of 0 and 1 was calculated. The percentage of the right results was calculated and provided in the table.

The iterating network learning process is presented in the pictures below (Figures 10–14).

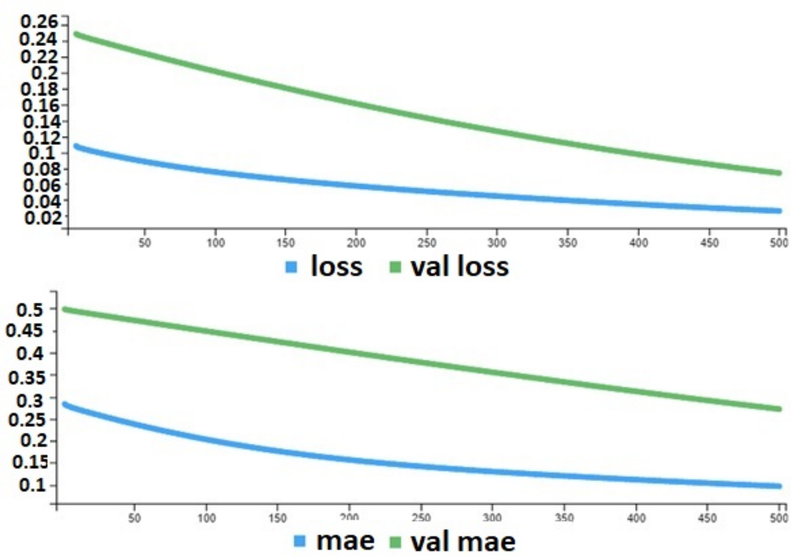

Figure 10. Deep neural network—learning process.

Table 1. Network error.

No.	Network	% Error
1	Deep Neural Network 1	0.04220253
2	Fuzzy Neural Network with OFN 1	0.00036679
3	Fuzzy Neural Network with OFN 2	0.00610502
4	Fuzzy Neural Network with OFN 3	0.00284964
5	Fuzzy Neural Network with OFN 4	0.00612133

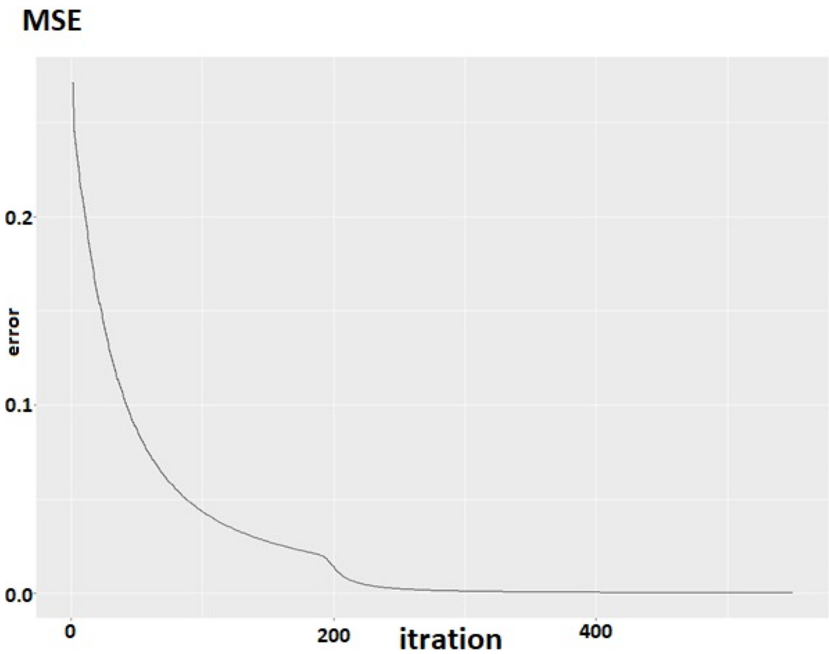

Figure 11. Fuzzy network 1—learning process.

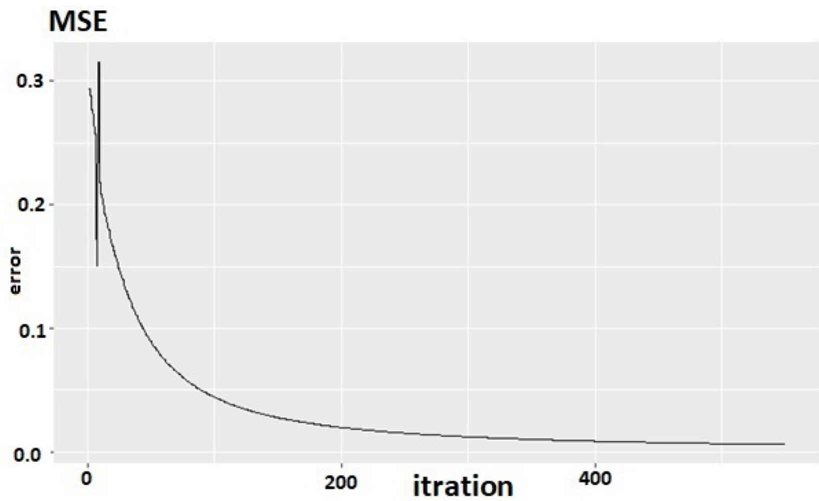

Figure 12. Fuzzy network 2—learning process.

As can be noted from the table, the acquired network accuracy results are promising. Whereas, the network learning process can be considered varied. The initial large changes in the error value are noticeable. The developed algorithm allowed us to acquire satisfying results; however, the developed algorithm still needs a lot of work. When we consider many devices used by people in their life, including a number of small ones with low computational power, there is a necessity to possess some algorithms to control devices and programs' status in a simple way. In such places there is no possibility to install solutions that require high computational power and large amounts of memory. Then, possible problems with the devices may affect the quality of life. The authors propose a fuzzy neural network with ordered fuzzy numbers that does not require high computational power and can be used to detect and predict problems with the devices and programs. The results of the conducted tests using an ordinary network and a proposed fuzzy network prove that the proposed network requires less neurons. Therefore, it requires less computational power. So, it can be implemented in small devices used by people to recognize (predict) problems, and consequently raise the quality of life.

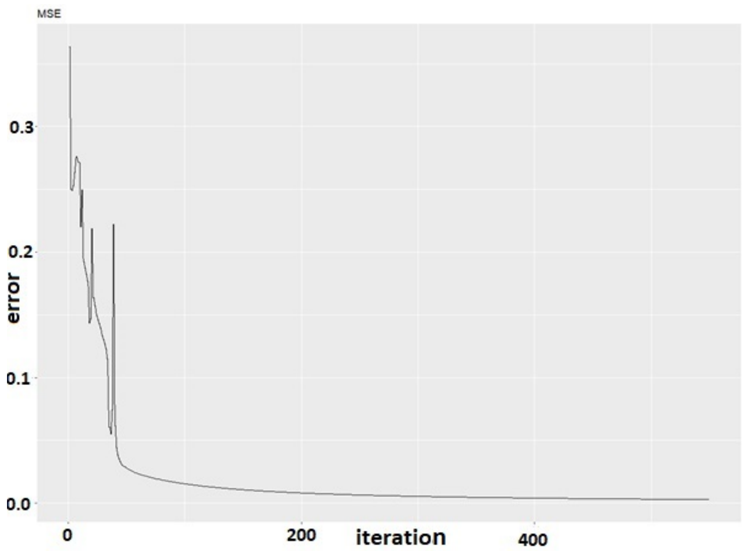

Figure 13. Fuzzy network 3—learning process.

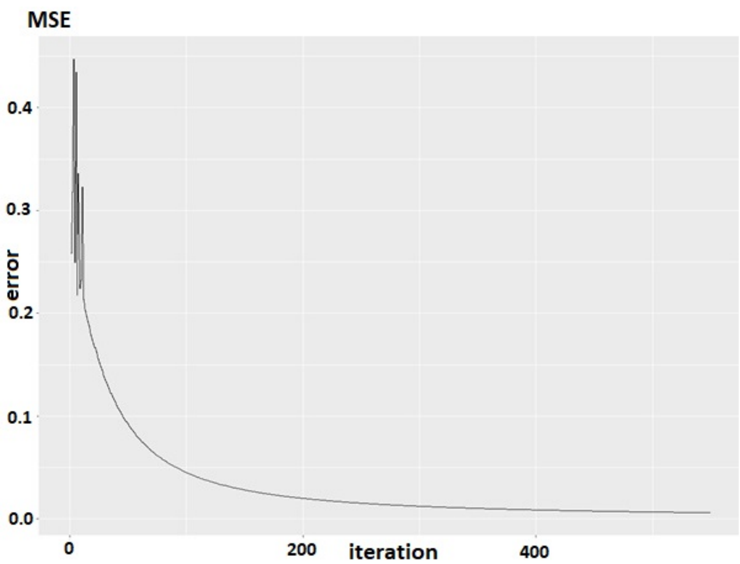

Figure 14. Fuzzy network 4—learning process.

6. Discussion

The application of the fuzzy neuron network with fuzzy neuron, using OFN, led to the detection of Iris flower. The test has proven that such a network works and the developed algorithm allows us to perform the learning process. Of course, not all the networks learned perfectly, but ultimately the process turned out to be successful. It is worth noting that even a network with a smaller number of neurons in the hidden layer allowed the iris flower to be recognized with a satisfactory result. Of course, not all networks learned perfectly, but in the final effect, the process was successful. It is worth noting that even a network with a smaller number of neurons in the hidden layer enabled recognized the Iris flower with a satisfactory result. It is worth noting that even the network with a lower number of neurons in the hidden layer allowed for detection of the Iris flower with a satisfactory result. As expected, the fuzzy network requires a lower number of neurons. Obviously, this is only one test example; in other cases, the results may not be satisfactory, so the network requires further research.

There are many places in which the proposed solutions can be used. These places are small businesses and small devices in which there is still small computational power. It could be used in a small security center, because it can be used in a virtual machine to collect logs and predict future problems. There are many places in which AI with a neural network can be implemented, but it was not used before. It requires a lot of computational power. Such places are small Internet of Things devices. On the condition the management software is installed on such devices, it will not be required to transfer large amounts of data to the management center. It will allow analyzing the data in the place where it operates. So, it may help to optimize the energy consumption by the devices or heating systems. Taking into account to the fuel and resource usage, it is very crucial.

7. Summary

This article presents the idea of using OFN arithmetic to improve the quality of everyday human life through better resource management. As it has been shown, the use of OFNs allows for faster detection of attacks on networks or devices, anticipating errors in transmission, which makes possible to switch the interface faster and save the energy needed for data retransmission. Finally, OFNs can be used to construct a fuzzy artificial neural network, and the applications of AI are very broad. It is only a matter of time before they will be commercially applicable and everyone will be able to come in touch with this technology. The main contribution of this paper is a novel fuzzy neural network with Ordered Fuzzy Numbers used in weights (and the OFN arithmetic for results calculations). This network can use lower amounts of neurons, which means that it will use less computational power to calculate the result. Therefore, this solution can be used in small devices such as Internet of Things components to analyze the data. The possibility to analyze the data in these devices allows avoiding the transfer of all data to the management center. Then, for example, smart home systems can use it to calculate optimal heating or smart city lights using AI can be more adaptive to the amount of traffic. Of course, there is a limited usability of the proposed network, e.g., according to time series analysis. The authors are working on preparing a framework to tensorflow, which is an open-source solution. This will let to use other kinds of networks, such as long short-term memory (LSTM) or convolutional neural network (CNN).

Author Contributions: Conceptualization, Ł.A. and R.M.; methodology, Ł.A.; software, Ł.A. and D.E.; validation, Ł.A. and D.E.; formal analysis, Ł.A; investigation, Ł.A.; resources, Ł.A. and D.E.; data curation, Ł.A. and D.E.; writing—original draft preparation, Ł.A.; writing—review and editing, Ł.A.; visualization, Ł.A. and D.E.; supervision, Ł.A. and R.M.; project administration, R.M.; funding acquisition, R.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the the project entitled "Smart anomaly prediction system in software environments" founded by IST Software and The Polish National Centre for Research and Development.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lee, S.C.; Lee, E.T. Fuzzy Neural Networks. *Math. Biosci.* **1975**, *23*, 151–177. [CrossRef]
2. Ishibuchi, H.; Kwon, K.; Tanaka, H. A learning algorithm of fuzzy neural networks with triangular fuzzy weights. *Fuzzy Sets Syst.* **1995**, *71*, 277–293. [CrossRef]
3. Buckley, J.J.; Hayashi, Y. Fuzzy neural networks: A survey. *Fuzzy Sets Syst.* **1994**, *66*, 1–13. [CrossRef]
4. de Campos Souza, P.V.; Lughofer, E. EFNN-NullUni: An evolving fuzzy neural network based on null-uninorm. *Fuzzy Sets Syst.* **2022**, *449*, 1–31. [CrossRef]
5. Liu, X.; Zhao, T.; Cao, J.; Li, P. Design of an interval type-2 fuzzy neural network sliding mode robust controller for higher stability of magnetic spacecraft attitude control. *ISA Trans.* 2023, *in press*. [CrossRef] [PubMed]
6. Zheng, K.; Zhang, Q.; Hu, Y.; Wu, B. Design of fuzzy system-fuzzy neural network-backstepping control for complex robot system. *Inf. Sci.* **2021**, *546*, 1230–1255. [CrossRef]
7. Zhang, Y.; Ma, H.; Xu, J. Neural network-based fuzzy vibration controller for offshore platform with random time delay. *Ocean. Eng.* **2021**, *225*, 108733. [CrossRef]
8. Yang, M.; Sheng, Z.; Yin, G.; Wang, H. A recurrent neural network based fuzzy sliding mode control for 4-DOF ROV movements. *Ocean. Eng.* **2022**, *256*, 111509. [CrossRef]
9. Zhang, R.; Gao, L. The Brushless DC motor control system Based on neural network fuzzy PID control of power electronics technology. *Optik* **2022**, *271*, 169879. [CrossRef]
10. Zhang, Q.; Wai, R. Distributed secondary control of islanded micro-grid based on adaptive fuzzy-neural-network-inherited total-sliding-mode control technique. *Int. J. Electr. Power Energy Syst.* **2022**, *137*, 107792. [CrossRef]
11. Wang, P.; Li, X.; Wang, N.; Li, Y.; Shi, K.; Lu, J. Almost periodic synchronization of quaternion-valued fuzzy cellular neural networks with leakage delays. *Fuzzy Sets Syst.* **2022**, *426*, 46–65. [CrossRef]
12. Hou, G.; Xiong, J.; Zhou, G.; Gong, L.; Huang, C.; Wang, S. Coordinated control system modeling of ultra-supercritical unit based on a new fuzzy neural network. *Energy* **2021**, *234*, 121231. [CrossRef]
13. Pang, M.; Zhang, Z.; Wang, X.; Wang, Z.; Lin, C. Fixed/Preassigned-time synchronization of high-dimension-valued fuzzy neural networks with time-varying delays via nonseparation approach. *Knowl.-Based Syst.* **2022**, *255*, 109774. [CrossRef]
14. Van, M. Higher-order terminal sliding mode controller for fault accommodation of Lipschitz second-order nonlinear systems using fuzzy neural network. *Appl. Soft Comput.* **2021**, *104*, 107186. [CrossRef]
15. Salari, A.H.; Mirzaeinejad, H.; Mahani, M.F. Tire normal force estimation using artificial neural networks and fuzzy classifiers: Experimental validation. *Appl. Soft Comput.* **2023**, *132*, 109835. [CrossRef]
16. Yadav, P.K.; Bhasker, R.; Upadhyay, S.K. Comparative study of ANFIS fuzzy logic and neural network scheduling based load frequency control for two-area hydro thermal system. *Mater. Today Proc.* **2022**, *56*, 3042–3050. [CrossRef]
17. Khan, S.; Asim, M.; Khan, S.; Musyafa, A.; Wu, Q. Unsupervised domain adaptation using fuzzy rules and stochastic hierarchical convolutional neural networks. *Comput. Electr. Eng.* **2023**, *105*, 108547. [CrossRef]
18. Wang, Y.; Ishibuchi, H.; Er, M.J.; Zhu, J. Unsupervised multilayer fuzzy neural networks for image clustering. *Inf. Sci.* **2023**, *622*, 682–709. [CrossRef]
19. Wang, C.; Lv, X.; Shao, M.; Qian, Y.; Zhang, Y. A novel fuzzy hierarchical fusion attention convolution neural network for medical image super-resolution reconstruction. *Inf. Sci.* **2023**, *622*, 424–436. [CrossRef]
20. Jabeur, S.B.; Serret, V. Bankruptcy prediction using fuzzy convolutional neural networks. *Res. Int. Bus. Financ.* **2023**, *64*, 101844. [CrossRef]
21. Wu, Z.; Jiang, B.; Gao, Q. State estimation and fuzzy sliding mode control of nonlinear Markovian jump systems via adaptive neural network. *J. Frankl. Inst.* **2022**, *359*, 8974–8990. [CrossRef]
22. Pham, P.; Nguyen, L.T.T.; Nguyen, N.T.; Kozma, R.; Vo, B. A hierarchical fused fuzzy deep neural network with heterogeneous network embedding for recommendation. *Inf. Sci.* **2023**, *620*, 105–124. [CrossRef]
23. Liu, J.; Shu, L.; Chen, Q.; Zhong, S. Fixed-time synchronization criteria of fuzzy inertial neural networks via Lyapunov functions with indefinite derivatives and its application to image encryption. *Fuzzy Sets Syst.* **2022**. [CrossRef]
24. Gong, S.; Guo, Z.; Wen, S. Finite-time synchronization of T-S fuzzy memristive neural networks with time delay. *Fuzzy Sets Syst.* **2022**. [CrossRef]
25. Lai, X.; Tong, S.; Zhu, G. Adaptive fuzzy neural network-aided progressive Gaussian approximate filter for GPS/INS integration navigation. *Measurement* **2022**, *200*, 111641. [CrossRef]
26. Koo, Y.D.; Jo, H.S.; Na, M.G.; Yoo, K.H.; Kim, C. Prediction of the internal states of a nuclear power plant containment in LOCAs using rule-dropout deep fuzzy neural networks. *Ann. Nucl. Energy* **2021**, *156*, 108180. [CrossRef]
27. Huang, W.; Oh, S.; Pedrycz, W. Fuzzy reinforced polynomial neural networks constructed with the aid of PNN architecture and fuzzy hybrid predictor based on nonlinear function. *Neurocomputing* **2021**, *458*, 454–467. [CrossRef]
28. de Campos Souza, P.V.; Lughofer, E.; Guimaraes, A.J. An interpretable evolving fuzzy neural network based on self-organized direction-aware data partitioning and fuzzy logic neurons. *Appl. Soft Comput.* **2021**, *112*, 107829. [CrossRef]
29. Nasiri, H.; Ebadzadeh, M.M. MFRFNN: Multi-Functional Recurrent Fuzzy Neural Network for Chaotic Time Series Prediction. *Neurocomputing* **2022**, *507*, 292–310. [CrossRef]
30. Dong, L.; Jiang, F.; Wang, M.; Li, X. Fuzzy deep wavelet neural network with hybrid learning algorithm: Application to electrical resistivity imaging inversion. *Knowl.-Based Syst.* **2022**, *242*, 108164. [CrossRef]

31. Kadak, U. Neural network operators of fuzzy n-cell number valued functions and multidimensional fuzzy inference system. *Knowl.-Based Syst.* **2022**, *258*, 110018. [CrossRef]
32. Kuo, R.J.; Zulvia, F.E. The application of gradient evolution algorithm to an intuitionistic fuzzy neural network for forecasting medical cost of acute hepatitis treatment in Taiwan. *Appl. Soft Comput.* **2021**, *111*, 107711. [CrossRef]
33. Kadak, U. Multivariate fuzzy neural network interpolation operators and applications to image processing. *Expert Syst. Appl.* **2022**, *206*, 117771. [CrossRef]
34. Kong, Y.; Chen, S.; Jiang, Y.; Wang, H.; Chen, H. Zeroing neural network with fuzzy parameter for cooperative manner of multiple redundant manipulators. *Expert Syst. Appl.* **2023**, *212*, 118735. [CrossRef]
35. Zhang, C.; Oh, S.; Fu, Z. Hierarchical polynomial-based fuzzy neural networks driven with the aid of hybrid network architecture and ranking-based neuron selection strategies. *Appl. Soft Comput.* **2021**, *113*, 107865. [CrossRef]
36. de Campos Souza, P.V. Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. *Appl. Soft Comput.* **2020**, *92*, 106275. [CrossRef]
37. Jang, J.-S. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685. [CrossRef]
38. Zadeh, L. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
39. Łukasiewicz, J. O logice trójwartościowej. *Stud. Filoz.* **1988**, 270–280.
40. Dubois, D.; Prade, H. Operations on fuzzy numbers. *Int. J. Syst. Sci.* **1978**, *9*, 613–626. [CrossRef]
41. Dubois, D.; Prade, H. Fuzzy elements in a fuzzy set. In Proceedings of the IFSA, Beijing, China, 28–31 July 2005; Volume 5, pp. 55–60.
42. Dubois, D.; Prade, H. Gradual elements in a fuzzy set. *Soft Comput.* **2008**, *12*, 165–175. [CrossRef]
43. Kosiński, W. On fuzzy number calculus. *Int. J. Appl. Math. Comput. Sci.* **2006**, *16*, 51–57.
44. Kosiński, W.; Słysz, S. Fuzzy numbers and their quotient space with algebraic operations. *Bull. Polish Acad. Sci. Ser. Tech. Sci.* **1993**, *41*, 285–295.
45. Kosiński, W.; Prokopowicz, P.; Ślęzak, D. On algebraic operations on fuzzy numbers. In *Intelligent Information Processing and Web Mining*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 353–362.
46. Kosiński, W.; Prokopowicz, P.; Ślęzak, D. Ordered fuzzy numbers, Bulletin of the Polish Academy of Sciences. *Ser. Sci. Math.* **2003**, *51*, 327–338.
47. Kosiński, W.; Prokopowicz, P.; Frischmuth, K. On Algebra of Ordered Fuzzy Numbers. In *Soft Computing Foundations and Theoretical Aspects*; Krassimir, J.K., Atanassow, T., Hryniewicz, O., Eds.; Springer: Warszawa, Poland, 2004.
48. Apiecionek, Ł.; Dobrosielski, W.; Ewald, D. Ordered Fuzzy Numbers for IoT Smart Home Solution. In *Advances and New Developments in Fuzzy Logic and Technology, Proceedings of the IWIFSGN'2019—The Eighteenth International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets, Warsaw, Poland, 24–25 October 2019*; Atanassov, K.T., Atanassova, V., Kacprzyk, J., Kałuszko, A., Krawczak, M., Owsiński, J.W., Sotirov, S.S., Sotirova, E., Szmidt, E., Zadrożny, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2021.
49. Apiecionek, Ł. Limiting Energy Consumption by Decreasing Packets Retransmissions in 5G Network. *Mob. Inf. Syst.* **2017**, *2017*, 4291091. [CrossRef]
50. Apiecionek, L.; Czerniak, J.; Dobrosielski, W.; Ewald, D. Fuzzy logic load balancing for cloud architecture network—A simulation test. In *Advances in Fuzzy Logic and Technology 2017, Proceedings of the EUSFLAT-2017—The 10th Conference of the European Society for Fuzzy Logic and Technology, Warsaw, Poland, 11–15 September 2017*; Springer International Publishing: Cham, Switzerland, 2017; pp. 43–54.
51. Czerniak, J.; Apiecionek, Ł.; Zarzycki, H. Protection Tool for Distributed Denial of Services Attack. In Proceedings of the Beyond Databases, Architectures and Structures: 10th International Conference, BDAS 2014, Ustron, Poland, 27–30 May 2014; Kozielski, S., Mrozek, D., Kasprowski, P., Małysiak-Mrozek, B., Kostrzewa, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 405–414.
52. Apiecionek, L.; Czerniak, J.; Dobrosielski, W.; Ewald, D. New proposal of fuzzy observation of ddos attack. In *Uncertainty and Imprecision in Decision Making and Decision Support: CrossFerti-lization, New Models, and Applications, Proceedings of the BOS-2016 and IWIFSGN-2016, Warsaw, Poland, 12–14 October 2016*; Atanassov, K.T., Kacprzyk, J., Kałuszko, A., Krawczak, M., Owsiński, J., Sotirov, S., Sotirova, E., Szmidt, E., Zadrożny, S., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 240–250.
53. Prokopowicz, P.; Czerniak, J.M.; Mikołajewski, D.; Apiecionek, Ł. *Theory and Applications of Ordered Fuzzy Numbers A Tribute*; Kosiński, W., Ed.; Springer: Berlin/Heidelberg, Germany, 2017.
54. Machine Learning Repository. Available online: http://archive.ics.uci.edu/ml/datasets/Iris (accessed on 4 November 2022).