

# Neural Networks Using Fuzzy Perceptrons in Medical Datasets

Javier Sotelo, Patricia Melin\*, Oscar Castillo

Tijuana Institute of Technology/TecNM,  
Mexico

javier.sotelo@tectijuana.edu.mx, {\*pmelin,octasillo}@tectijuana.mx

**Abstract.** Fuzzy logic is a generalization of traditional logic that works with values more familiar to natural language, allowing problems to be solved that would be very difficult to solve using binary logic. It goes hand in hand with other areas of AI, such as evolutionary computing and neural networks. In the latter, it is thought that the potential of fuzzy logic can be exploited even further, which is why the fuzzy perceptron is proposed. This proposal, a combination of the perceptron with fuzzy logic, consists of applying an activation function with a fuzzy system that determines, through rules, the output of the artificial neuron based on the weighted sum of its inputs and weights. Medical datasets have a wide variety of data compositions, such as different data volumes, data types, organization, etc., which is useful for training neural networks. Therefore, experiments are carried out using various clinical datasets to train neural networks with the proposed activation function and compare them with training a traditional neural network with the aim of achieving improved accuracy in the classification of clinical data.

**Keywords.** Neural networks, fuzzy logic, perceptron, artificial intelligence, neuro-fuzzy.

## 1 Introduction

Neural networks are designed to learn how to classify, identify characteristics, predict, etc. from a data source provided to them. They are widely used today in many sectors of industry, education, health, and even entertainment. They have been around for quite a few years, but today it is very common to hear people talking about artificial intelligence, or see comments about it on the internet or TV. Neural networks are part of that artificial intelligence, more specifically, machine learning, and in some cases, deep learning. The fact that we see artificial intelligence in everyday

matters is largely due to the evolution of hardware, as the devices we use every day, such as smartphones or desktop computers, now have the processing resources necessary to perform neural network training tasks, which previously could only be performed by high-end equipment or equipment specifically dedicated to those tasks. But it's not all about hardware. Since their inception, neural networks have also evolved in terms of architecture, activation functions, performance, etc.

Although current hardware is now better equipped to train neural networks, there are still challenges in this area, for example, when dealing with noisy datasets, class imbalance, insufficient information, among others.

Another limitation lies in activation functions, because the vast majority are fixed, static, designed for general problems, and lack adaptability or freedom to be configured, which may make them less useful in contexts that require more dynamic learning, such as datasets that change or data that needs to be transformed in real time while training.

Unlike functions developed with traditional logic, fuzzy logic allows uncertainty to be handled using smooth approximations, in contrast to the abrupt Boolean values present in classical logic. The combination of fuzzy systems with neural networks has given rise to what is known as neuro-fuzzy systems, which are a very important subject of study in the field of intelligent hybrid systems.

This paper proposes the fuzzy perceptron, a temporary name for a perceptron prototype with a proposed activation function, which consists of a fuzzy system with two rules that evaluate the activation state of the neuron. This fuzzy system, through its parameters, allows the membership

functions to be adapted to the desired problem until the appropriate accuracy is achieved, in contrast to static functions that cannot be parameterized. The activation function developed is tested with three medical datasets of different characteristics and volumes.

The objective is to verify whether it is possible, using fuzzy logic, for an activation function suited to the behavior exhibited by perceptrons with different datasets to perform better in terms of either accuracy or lower loss per epoch than one of the most widely used activation functions: ReLU. Thus, the following section presents the state of the art, followed by a proposal, methodology, results, and conclusions.

## 2 State of the Art

The artificial neuron dates back to 1943 when it was invented by Warren McCulloch and Walter Pitts [1], and later in 1957 it was implemented by Frank Rosenblatt [2] in hardware. Also known as Perceptron, this algorithm consists of a unit that receives a series of inputs  $x_i$  ( $i$  is a value from 1 to  $n$ ), calculates a weighted sum of these inputs, where each input is multiplied by a randomly generated weight and a bias is added [3]. Finally, it passes through an activation function that determines the state of the neuron, i.e., whether the neuron is active or not. If activated, it will transmit the information to the next neuron. The latter occurs only in multilayer perceptron networks that are designed to solve nonlinear problems [4, 5]. Modifications have been made to Rosenblatt's perceptron by various authors, but to this day, the original algorithm remains the most widely used. Some examples include the strategic perceptron [6], voted perceptron [7], second-order perceptron [8], optimistic perceptron [9], and morphological neuron [10]. Multilayer perceptron neural networks are currently used in various applications such as speech recognition, computer vision, pattern recognition, object detection, etc. [11].

The fuzzy logic proposed by Lofti A. Zadeh is an extension of traditional Boolean logic, differing in that it is not limited to yes or no, i.e., 1 or 0, but encompasses a wider range of values. In other words, let's take the temperature on a summer day, which is usually a warm day, depending on the

area, it could be 25 degrees Celsius, 30 degrees Celsius, etc. For traditional logic, this is defined as hot or 1, and the other value would be cold or 0. There are only two possibilities, so it is necessary to define what degree is hot and what degree is cold. Assuming that the threshold is set at 20 degrees Celsius, anything above that is considered hot, and just 1 degree below that is cold, with the difference being barely noticeable in the environment.

For some temperature control systems, it could be complicated when trying to regulate the temperature if fluctuations between 19 and 21 degrees Celsius are detected. Fuzzy logic allows us to recognize similar characteristics as we use them in natural language, for example, very warm, very cold, somewhat warm, somewhat cold, temperate, lukewarm, etc.

This allows better instructions to be assigned to the temperature regulator so that it works according to a set of fuzzy rules that define the temperature ranges that will be considered either cold, hot, very hot, warm, etc., as well as their degrees of membership, where 1 is the complete parameter, i.e., for the cold rule, 1 would be cold and 0 would be the absence of cold, with a middle point between 0 and 1 or even another value, such as 0.5, 0.8, etc., and in turn sharing a degree of membership with another rule, for example, warm, which is the middle point between hot and cold [12, 13].

There are several types of fuzzy logic, such as Type-1 and Interval Type-2, where a degree of uncertainty is added to the fuzzy system, which is useful for measuring different opinions given that, returning to the previous example of temperature, a warm day for one person may be less warm for another [14], generalized Type-2, and more recently Type-3 by intervals, which is still being researched and developed by various authors [15]. Fuzzy logic is an area of research with various applications and notable recent advances, such as in object detection [16], evolutionary algorithms [17-19], water quality prediction [20], photovoltaic systems [21], climate control [22], etc. Activation functions in neural networks serve to transform the neuron's input into an output; otherwise, it would be just a linear regression model. The activation function must be a nonlinear function, as well as differentiable, in order to apply backpropagation,

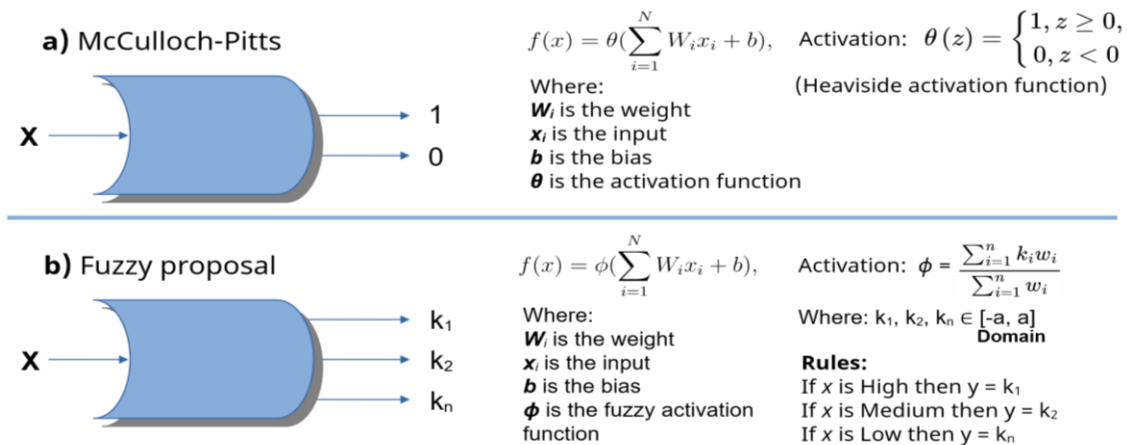


Fig. 1. Comparison of the perceptron and the fuzzy perceptron

which allows for dynamic weight adjustment and improved overall network accuracy. Various activation functions are used, each with different purposes. For example, ReLU is one of the most widely used in hidden layers due to its accuracy and simplicity. Leaky ReLU is a modified version of ReLU, the difference with the function on which it is based being that instead of returning 0 for negative values, leaky ReLU returns a small value close to 0, without being 0.

In output layers, Sigmoid is often used for binary outputs, Softmax for non-binary outputs, and other functions include ELU, Swish, Tanh, etc. [23]. New activation functions continue to be researched in the literature, with objectives such as improving the accuracy of neural networks, reducing epoch losses, and solving more complex problems, among others [24, 25, 26].

Neuro-fuzzy systems are systems that combine neural networks and fuzzy logic. Neural networks enable knowledge acquisition, reasoning, and learning, while fuzzy logic offers reasoning mechanisms for information that may be incomplete or unclear, as it allows for approximation through inference [27].

There are many ways to implement neuro-fuzzy systems, as both separately already have a wide variety of applications. Recent research includes classification of cardiovascular diseases [28],

construction scheduling [29], student performance [30], traffic prediction [31], etc.

### 3 Proposal

A new activation function is proposed, consisting of a fuzzy system that evaluates the neuron's output. This prototype has been given the preliminary name of fuzzy perceptron and has two rules: if the input is high, then the output is high; if the input is low, then the output is low. The Sugeno (TSK) inference method is used, so the outputs are constant. Therefore, the final rules are described as follows: If  $X$  is high, then  $y = 1$ ; if  $X$  is low, then  $y = -1$ , where “ $X$ ” is the neuron input and “ $y$ ” is the result of the weighted sums of the perceptron, i.e., the output. Figure 1 shows a comparison of the proposal with the Heaviside activation function. It can be seen that the rules of the fuzzy system are described in general terms, but as explained previously, only two rules are used. The behavior of the perceptron (inputs, weights, etc.) remains the same.

The reason why fuzzy logic is used is because it makes neural network learning more dynamic, i.e., it allows the activation function to be adapted to the dataset by changing the parameters of the membership functions to match the weights of the neural network.

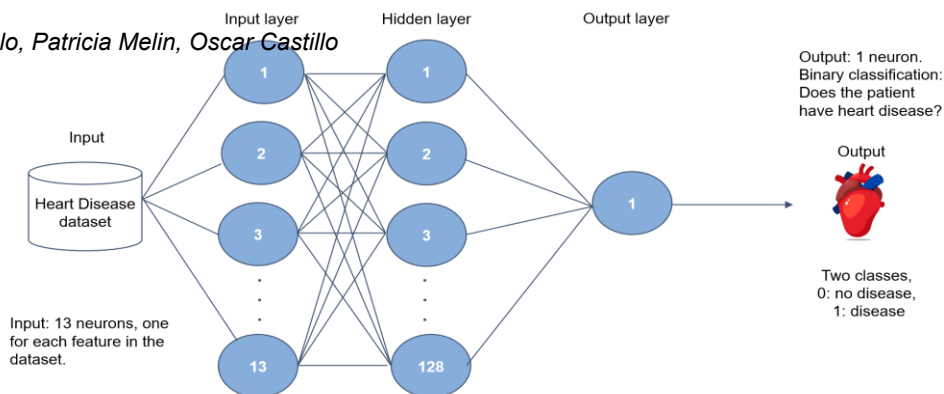


Fig. 2. Neural network architecture

The reason why fuzzy logic is used is because it makes neural network learning more dynamic, i.e., it allows the activation function to be adapted to the dataset by changing the parameters of the membership functions to match the weights of the neural network. In this way, it is thought that better accuracy can be achieved by having an activation function designed specifically for each training session. There may be cases of very similar datasets or those with similar behavior, which would allow them to share the same parameters. However, in most scenarios, they should have some difference in at least one parameter. The main advantage of using fuzzy logic is that it provides some freedom for network adjustment by allowing parameters to be changed, something that does not occur in activation functions that are fixed, such as ReLU.

## 4 Methodology

To demonstrate how the fuzzy perceptron works, three experiments are conducted in which neural networks are trained using the proposed activation function and compared with ReLU. Each experiment is conducted with a different dataset. The datasets come from the UC Irvine Machine Learning Repository and are specifically: Heart Disease, Breast Cancer Wisconsin (Diagnostic), and CDC Diabetes Health Indicators.

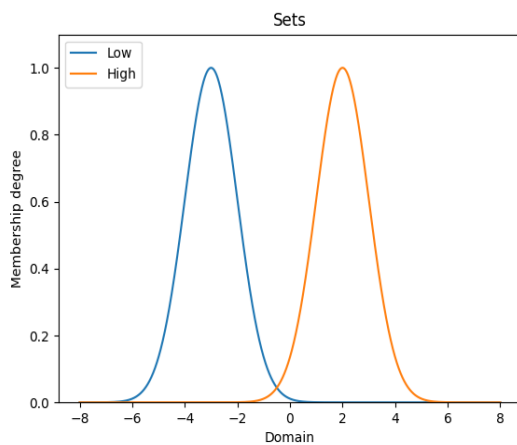
Heart Disease [32] is a dataset consisting of 303 instances and 13 features. It contains health information about patients with heart conditions. The objective of this dataset is to classify whether or not the patient has heart disease. It is one of the most popular UCIML datasets due to its small size

and simplicity as a binary classification. Breast Cancer Wisconsin (Diagnostic) [33] is the second dataset used, with 569 instances and 30 features. It is also quite popular as it is among the top results on the UCI website. Its information deals with cell nucleus characteristics from previous studies. The objective of this dataset is to use binary classification to determine whether the patient has a malignant or benign tumor.

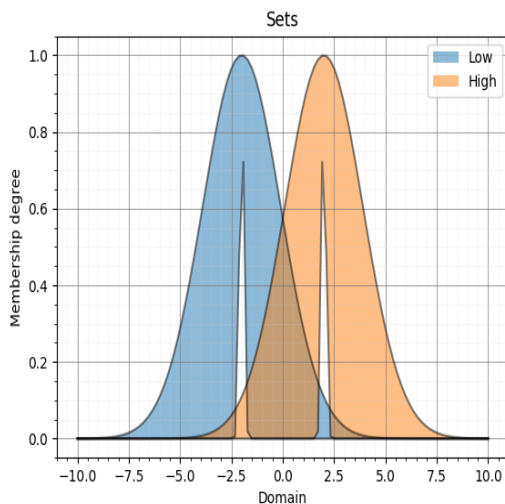
The last one is CDC Diabetes Health Indicators [34], which consists of information on people's health and lifestyle as well as their diabetes diagnosis. It has 21 features and 253,680 instances. It is less popular and has a much larger volume of data than the previous ones mentioned. Its objective is to predict, through binary classification, whether the patient is healthy or has diabetes (or prediabetes).

Datasets from the medical field are used due to the variety, volume, and even imbalance of their datasets, which allows for the study of different techniques with the aim of improving accuracy, reducing noise, etc. [35].

The first experiment conducted with Heart Disease uses a neural network architecture with 13 neurons in the input layer, each corresponding to a feature of the dataset, a hidden layer composed of 128 neurons, and an output layer consisting of a single neuron given its binary classification. The objective of this network is to evaluate whether or not the patient has heart disease. The activation functions used in the hidden layer are ReLU, which is compared with the proposed Type-1 and Type-2 fuzzy functions. The Sigmoid function is used for the output in all scenarios. Five hundred epochs are used for training and a batch size of 32. Figure



**Fig. 3.** Type-1 Gaussian membership functions



**Fig. 4.** Type-2 Gaussian membership function

2 shows a diagram of the neural network architecture used in the first experiment.

Type-1 and type-2 fuzzy logic is used by intervals for each of the experiments. In the case of the Heart Disease experiment, the membership functions for type 1 and type 2 are shown in Figures 3 and 4, respectively. Both membership functions are Gaussian, and each represents a rule of the fuzzy system. For type 1, the parameters are -3 and 2 for the centers and for both sigmas. Type 2 has parameters of -2 and 2 for the centers, 1 for the uncertainty height in both cases, 2 for the

uncertainty width (both cases), and 0.5 for the sigma for both functions.

The neural network in the second experiment has a similar architecture to the previous one. It consists of an input layer with 30 neurons, one for each feature in the dataset, a hidden layer with 64 neurons, and an output layer with a single neuron. This network has a binary output that evaluates the patient's diagnosis, i.e., whether they have a malignant or benign tumor. In the hidden layer, ReLU, Type-1 fuzzy, and Type-2 fuzzy activation functions are used in each of the three training sessions, where they are compared against each other. Five hundred epochs and a batch size of 32 are used for training. Figure 5 shows a diagram of the neural network architecture of the second experiment.

The membership functions of the fuzzy systems used in the Type-1 and Type-2 fuzzy activation functions vary for each experiment due to the variation in output values of the weighted sums of the perceptrons. For the second experiment, Figures 6 and 7 show the Type-1 and Type-2 membership functions by intervals, respectively. Type-1 has parameters of -2 and 2 for the centers of the Gaussian functions and a sigma of 2 for both functions. Type-2 has centers of -2 and 2, uncertainty height of 1, uncertainty width of 2, and sigma of 0.5. Figures 6 and 7 show the membership functions for experiment 2.

The third experiment consists of evaluating a patient's diagnosis using binary classification, assessing whether the patient is healthy or has diabetes or prediabetes. The neural network architecture for this experiment is slightly more complex than the previous ones due to the larger number of instances in the dataset used. There are 21 neurons in the input layer, one for each of the characteristics in the dataset, 64 neurons for the first hidden layer, 32 neurons for the second hidden layer, and 16 for the third and final hidden layer. The output consists of a single neuron. Three hundred epochs and a batch size of 32 are used for training. Figure 8 shows a diagram of the neural network architecture of the third experiment.

The Type-1 membership functions of the third experiment have parameters of -2 and 2 for the centers and 2 sigma for both functions. The Type-2 membership functions have parameters of -2 and 2 for the centers, 1 uncertainty height, 2

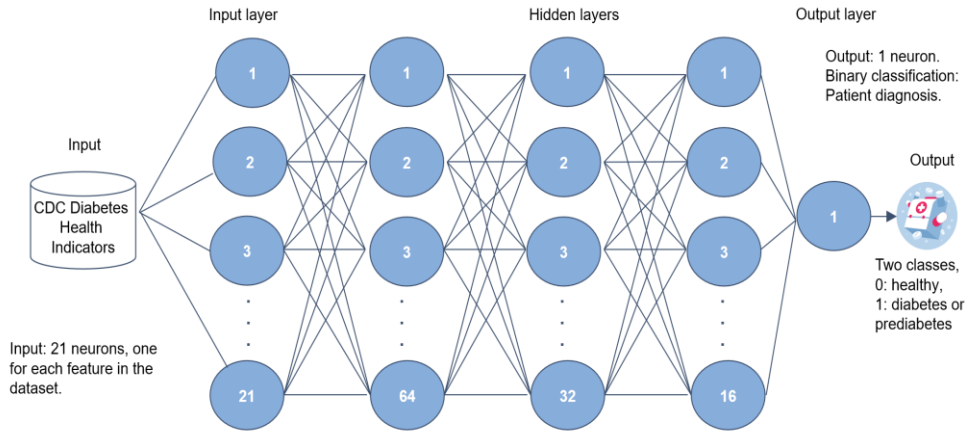


Fig. 5. Neural network architecture

uncertainty width, and 0.5 sigma. Figures 9 and 10 show the Type-1 and Type-2 membership functions, respectively.

## 5 Results

The ReLU and Type-1 fuzzy activation functions exhibit similar behavior in terms of loss and accuracy over time, as shown in Figures 11 and 12. In case of Type-2 fuzzy, it seems to take more epochs to achieve learning, as seen in Figure

13, since the loss is reduced more slowly than the other two and it takes more epochs to increase the accuracy compared to the previous ones.

Type-2 fuzzy logic could benefit from or require more epochs for better performance, which would be a disadvantage, as it means it requires more processing to learn.

However, as can be seen in the results in Table 1, Type-2 achieves better accuracy than Type-1, which in turn outperforms ReLU.

Thus, it can be said that with 500 epochs, Type-2 outperforms Type-1 and ReLU.

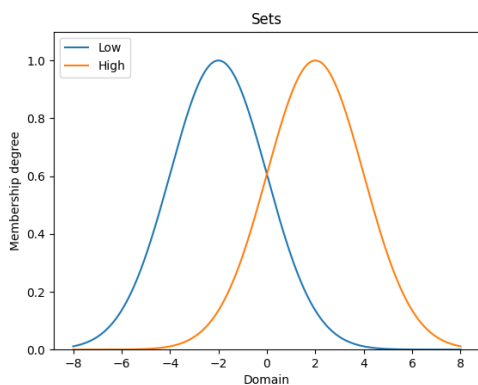


Fig. 9. Type-1 Gaussian membership functions

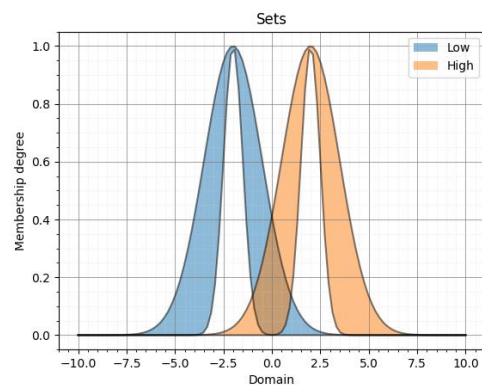
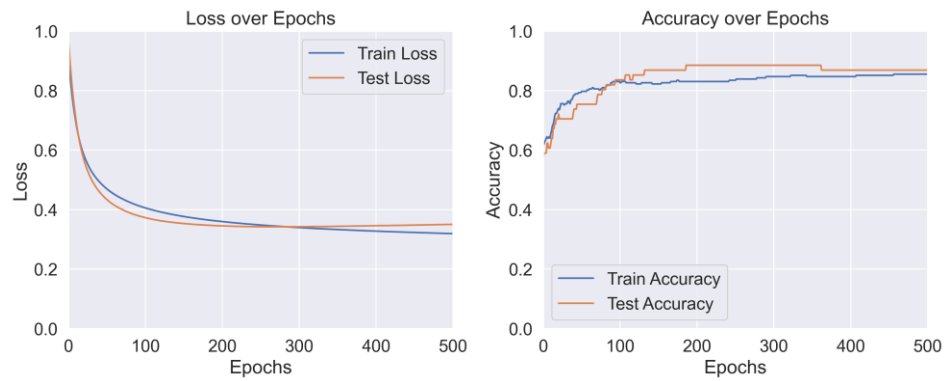
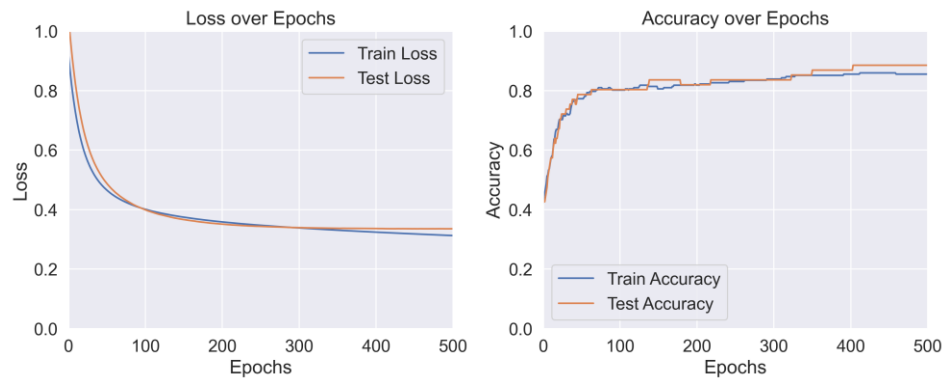


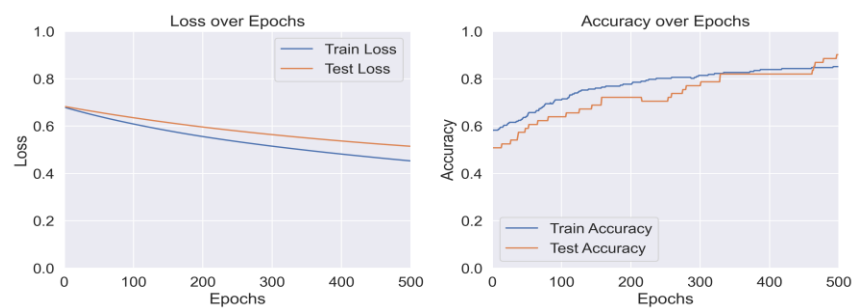
Fig. 10. Type-2 Gaussian membership functions



**Fig. 11.** Graphs showing loss (left) and accuracy (right) by epoch.



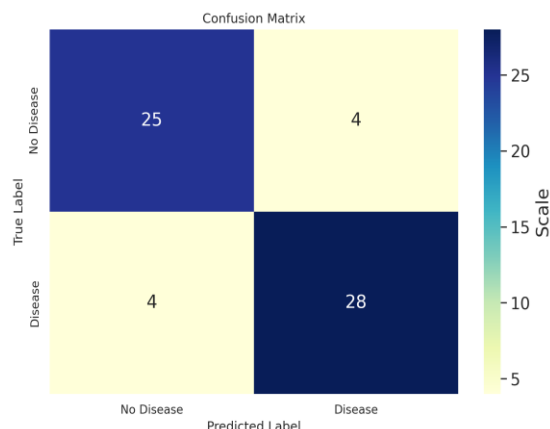
**Fig. 12.** Graphs showing loss (left) and accuracy (right) by epoch.



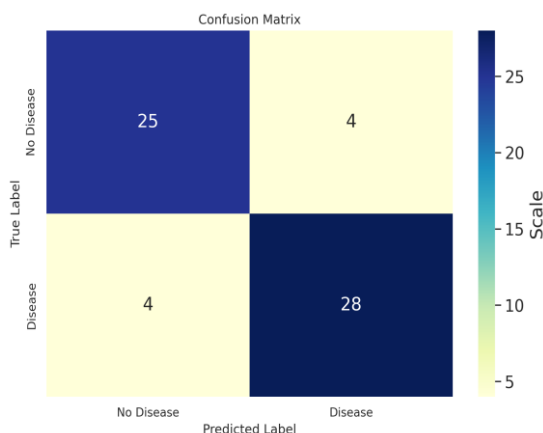
**Fig. 13.** Graphs showing loss (left) and accuracy (right) by epoch.

**Table 1.** Comparison of results from the first experiment

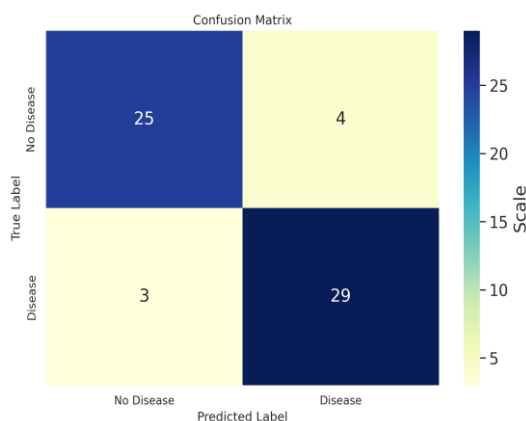
Metric	ReLU	Type-1	Type-2
Accuracy score	0.8688	0.8852	0.9016
Mean square error	0.1311	0.1147	0.0983
R <sup>2</sup>	0.4741	0.5378	0.6064



**Fig. 16.** Fuzzy Type-2 confusion matrix



**Fig. 14.** ReLU confusion matrix

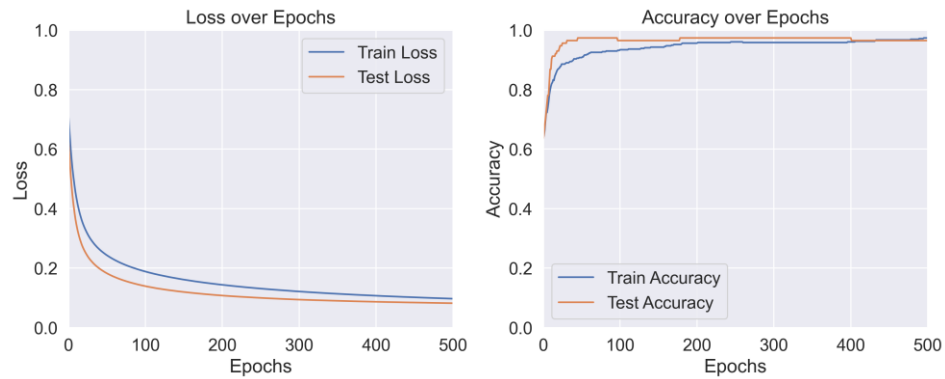


**Fig. 15.** Fuzzy Type-1 confusion matrix

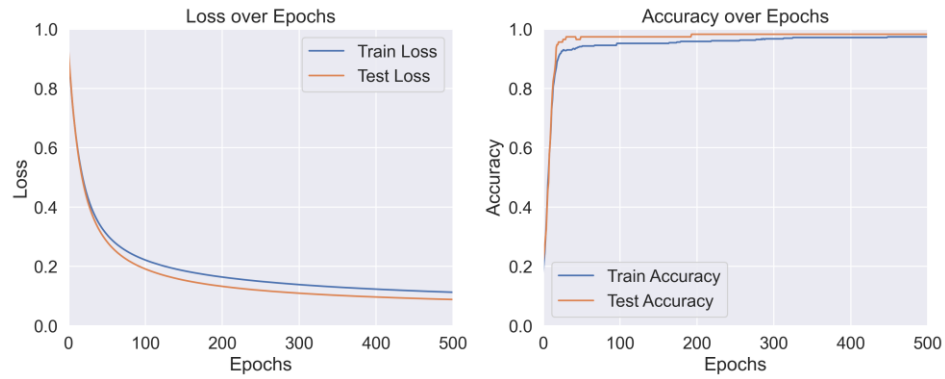
In the first experiment, it was observed that Type-2 requires greater effort to learn and is also the one that learns best, while Type-1 is only slightly better than ReLU. Figures 14, 15, and 16 show the confusion matrices for ReLU, Diffuse Type-1, and Diffuse Type-2, respectively. The three matrices are similar: ReLU is wrong in 8 cases, Type-1 in 7, and Type-2 in 6. The second experiment shows similar results for the three activation functions. ReLU seems to work very well, with loss decreasing rapidly and accuracy rising almost to the maximum after just a few epochs, as shown in Figure 17. Type-1 seems to take a little longer to reduce loss, but accuracy rises slightly more than with ReLU, as shown in Figure 18, and Type-2 shows better results overall than the other two, according to Figure 19 and Table 2.

The third experiment is performed with a much larger dataset than the previous ones.

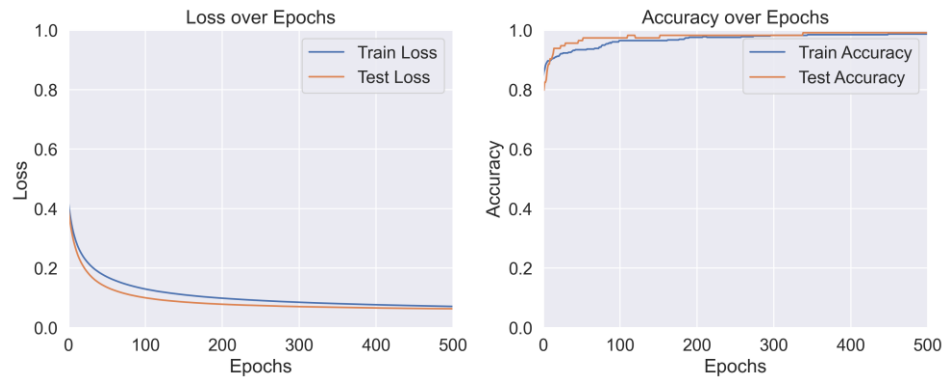
For all activation functions, the neural network seems to struggle to learn over the epochs, as the loss and accuracy hardly change, as seen in Figures 23, 24, and 25 for ReLU, Type-1 fuzzy, and Type-2 fuzzy, respectively. In this experiment, according to the metrics shown in Table 3, the winner in accuracy is ReLU, followed by Type-1 and Type-2. The second experiment, in contrast to the first, shows Type-2 learning faster than ReLU and Type-1, placing it in first place again in the metrics used, with Type-1 in second place. Overall, the three activation functions show good results, as can be seen in the confusion matrices in Figures



**Fig. 17.** Graphs showing loss (left) and accuracy (right) by epoch.



**Fig. 18.** Graphs showing loss (left) and accuracy (right) by epoch.



**Fig. 19.** Graphs showing loss (left) and accuracy (right) by epoch.

20, 21, and 22 for ReLU, Type-1, and Type-2, respectively.

In general, the accuracy for all three is not very high, as it does not reach an accuracy score of 0.8.

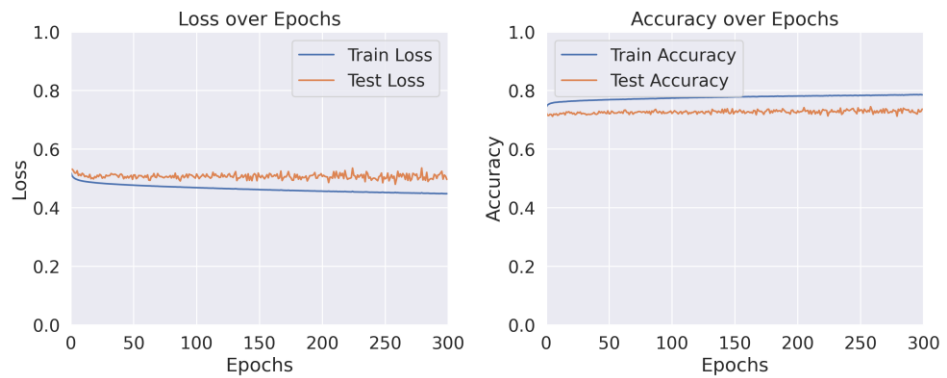


Fig. 23. Graphs showing loss (left) and accuracy (right) by epoch.

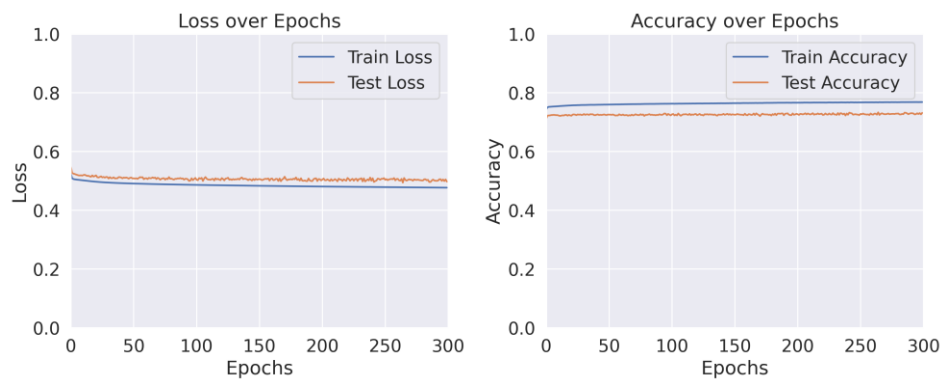


Fig. 24. Graphs showing loss (left) and accuracy (right) by epoch.

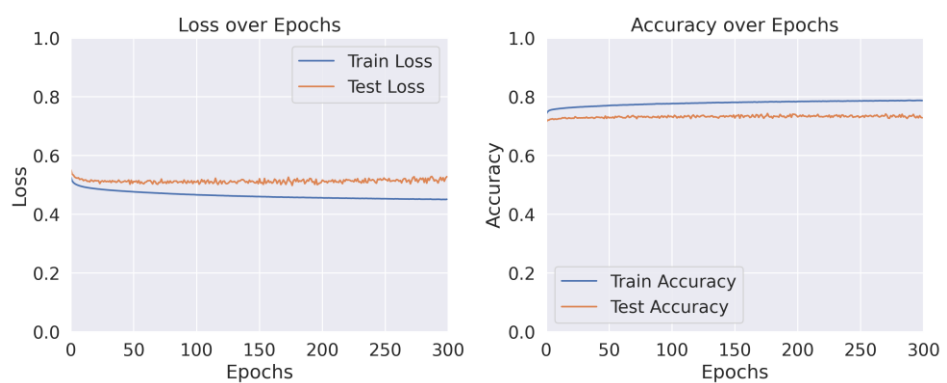


Fig. 25. Graphs showing loss (left) and accuracy (right) by epoch.

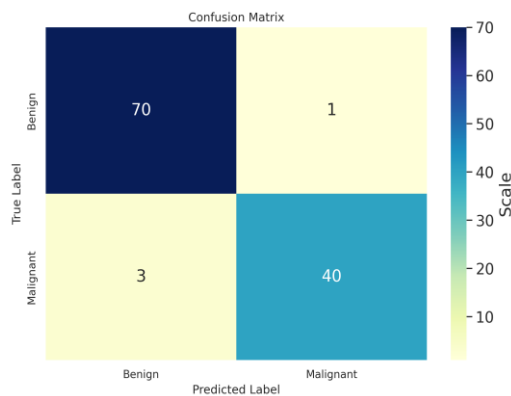


Fig. 20. ReLU confusion matrix

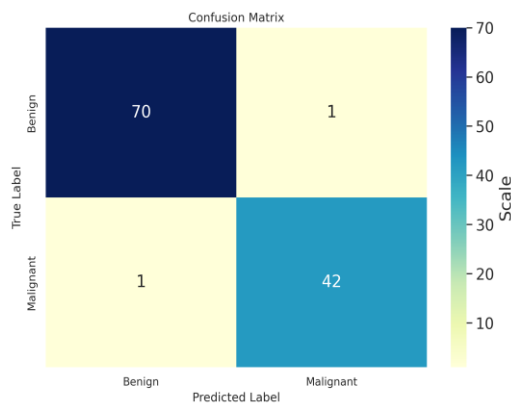


Fig. 21. Fuzzy Type 1 confusion matrix

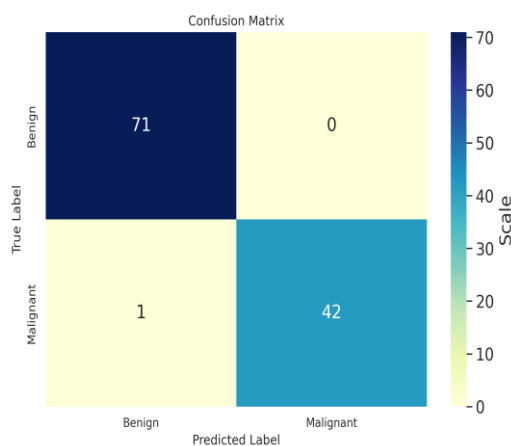


Fig. 22. Fuzzy Type-2 confusion matrix

**Table 2.** Comparison of results from the second experiment

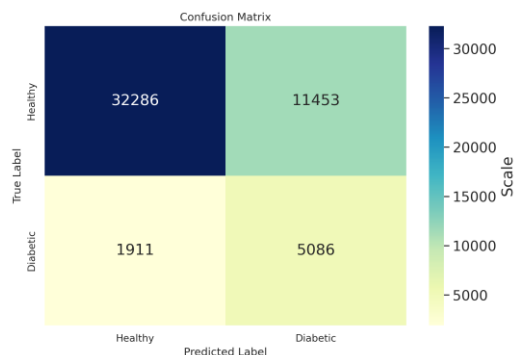
Metric	ReLU	Type-1	Type-2
Accuracy score	0.9649	0.9824	0.9912
Mean square error	0.0350	0.0175	0.0087
R <sup>2</sup>	0.8476	0.9253	0.9623

**Table 3.** Comparison of results from the third experiment

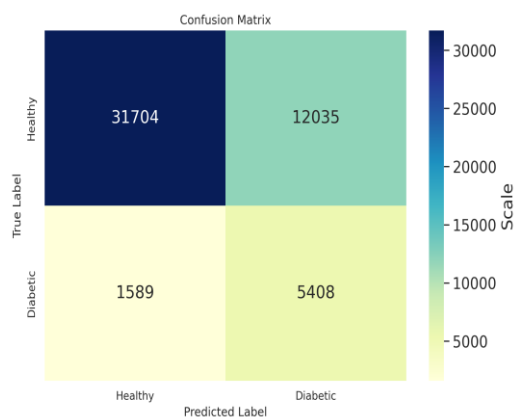
Metric	ReLU	Type-1	Type-2
Accuracy score	0.7365	0.7314	0.7285
Mean square error	0.2634	0.2685	0.2714
R <sup>2</sup>	-0.1988	-0.1902	-0.2142

It should be noted that CDC Diabetes Health Indicators is quite unbalanced in its classes, as nearly 90% of the dataset consists of instances of healthy patients, while the rest are for diabetics. With this imbalance, it takes effort for the neural network to learn. Even so, it is useful to show how activation functions behave in unbalanced scenarios, and judging by the results, it seems that fuzzy functions are not a good option.

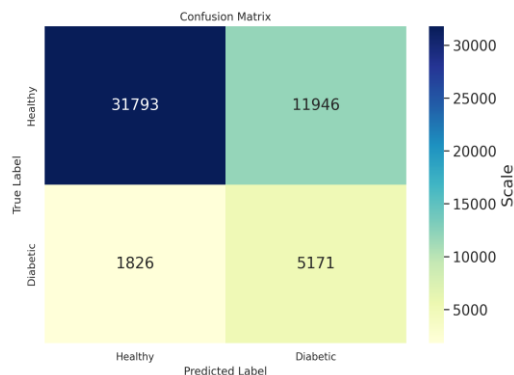
This last experiment took longer to train; the first two were completed in just a few seconds, while CDC Diabetes took more than an hour to complete the three training sessions, one for each activation function. It was decided to reduce the number of epochs from 500 to 300 to speed up training, also because no improvement was observed when increasing the number of epochs. Figures 26, 27, and 28 show the confusion matrices for ReLU, Type 1, and Type 2, respectively. As explained above, the performance of this neural network was not very good; the confusion matrices show a very high number of errors (more than 11,000 cases of healthy patients are classified as diabetic). In addition, the difference in the volume of healthy and diabetic patients can be observed, with healthy



**Fig. 26.** ReLU confusion matrix



**Fig. 27.** Type-1 confusion matrix



**Fig. 28.** Type-2 confusion matrix

patients numbering more than 40,000 and diabetic patients barely reaching 7,000.

## 6 Conclusions

Considering the results of the experiments, the use of fuzzy systems as an activation function may be a good implementation to slightly improve the accuracy of neural network training in exchange for a little more processing resource.

It was observed that in small datasets this represents an improvement, but in larger datasets the opposite was true.

To confirm whether the use of fuzzy logic in activation functions is recommended in large datasets, further testing is required, as the dataset used in the third experiment was unbalanced and it is therefore not possible to judge whether fuzzy logic performs worse in unbalanced or large datasets. Separate experiments are required to verify this.

Another detail is that no optimization algorithm such as a genetic algorithm or PSO was used to optimize both the neural network parameters and those of the fuzzy system.

This means that there is a limitation in that it is not known whether there is a better accuracy value for each of the experiments with fuzzy activation functions, i.e., Type-1 and Type-2 could show better results if such optimization techniques were used.

In future, we plan to optimize the parameters in order to determine whether it is possible to increase the difference in accuracy between ReLU and fuzzy functions, as well as to see if it is possible for fuzzy functions to outperform ReLU in the third experiment.

Another proposal for the future is to optimize the neural network parameters. In this case, the overall accuracy for the three activation functions would be improved, but the difference between one activation function and another should remain the same, unless, for example, experiment 2, which has almost perfect accuracy, the optimization of the neural network is such that the three activation functions show an accuracy score of 1.00, which would not be very useful when comparing.

In this way, datasets with noise or that present certain difficulties for training should also be sought.

Also, the parameters could be found using metaheuristics and the proposal could be applied in other problems [36-37].

## References

1. **McCulloch, W.S., Pitts, W. (1943).** A logical calculus of the ideas immanent in nervous activity, *The Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115–133.
2. **Rosenblatt, F. (1958).** The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review*, Vol. 65, No. 6, pp. 386–408. doi:10.1037/h0042519.
3. **Kanal, L.N. (2003).** Perceptron, *Encyclopedia of Computer Science*, John Wiley and Sons Ltd., GBR, pp. 1383–1385.
4. **Ramchoun, H., Ghanou, Y., Ettaouil, M., Janati Idrissi, M.A. (2016).** Multilayer perceptron: Architecture optimization and training, *International Journal of Interactive Multimedia and Artificial Intelligence*, Vol. 4, No. 1, pp. 26+.
5. **Popescu, M.C., Balas, V.E., Perescu-Popescu, L., Mastorakis, N. (2009).** Multilayer perceptron and neural networks, *WSEAS Transactions on Circuits and Systems*, Vol. 8, No. 7, pp. 579–588.
6. **Ahmadi, S., Beyhaghi, H., Blum, A., Naggita, K. (2021).** The strategic perceptron, *Proceedings of the 22nd ACM Conference on Economics and Computation*, pp. 6–25.
7. **Freund, Y., Schapire, R.E. (1998).** Large margin classification using the perceptron algorithm, *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pp. 209–217.
8. **Cesa-Bianchi, N., Conconi, A., Gentile, C. (2005).** A second-order perceptron algorithm, *SIAM Journal on Computing*, Vol. 34, No. 3, pp. 640–668.
9. **Hanashiro, R., Abernethy, J. (2020).** Linear separation via optimism, *arXiv Preprint*, arXiv:2011.08797.
10. **Mondal, R., Santra, S., Mukherjee, S.S., Chanda, B. (2019).** Morphological network: How far can we go with morphological neurons? *arXivLabs*. doi: 10.48550/arXiv.1901.00109.
11. **Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A., Arshad, H. (2018).** State-of-the-art in artificial neural network applications: A survey, *Heliyon*, Vol. 4, No. 11.
12. **Zadeh, L.A. (1988).** Fuzzy logic, *Computer*, Vol. 21, No. 4, pp. 83–93.
13. **Zadeh, L.A. (1965).** Fuzzy sets, *Information and Control*, Vol. 8, No. 3, pp. 338–353.
14. **Wu, D. (2010).** A brief tutorial on interval type-2 fuzzy sets and systems, *Fuzzy Sets and Systems*, Vol. 4, No. 1, pp. 220–225.
15. **Valdez, F., Castillo, O., Melin, P. (2025).** A bibliometric review of type-3 fuzzy logic applications, *Mathematics*, Vol. 13, No. 3, pp. 375.
16. **Gonzalez, C.I., Melin, P., Castillo, O. (2017).** Edge detection method based on general type-2 fuzzy logic applied to color images, *Information*, Vol. 8, No. 3, pp. 104.
17. **Valdez, F., Castillo, O., Melin, P. (2025).** A Review on Enhanced Evolutionary Algorithms Based on Extensions of Fuzzy Logic Systems, *Computación y Sistemas*, Vol. 29, No. 2.
18. **Guajardo, H.M., Valdez, F. (2024).** Dragonfly Algorithm for Benchmark Mathematical Functions Optimization, *Computación y Sistemas*, Vol. 28, No. 2, pp. 769–782.
19. **Lizarraga, E., Valdez, F., Melin, P., Castillo, O. (2025).** A Hybrid Enhanced Mayfly Optimization Algorithm with Improved Performance through Fuzzy-Based Automatic Parameter Adaptation, *Computación y Sistemas*, Vol. 29, No. 2.
20. **Trach, Y., Trach, R., Kuznietsov, P., Pryshchepa, A., Biedunkova, O., Kiersnowska, A., Statnyk, I. (2024).** Predicting the influence of ammonium toxicity levels in water using fuzzy logic and ANN models, *Sustainability*, Vol. 16, No. 14, pp. 5835.
21. **Belhadj, S.M., Meliani, B., Benbouhenni, H., Zaidi, S., Elbarbary, Z.M.S., Alammam, M.M. (2025).** Control of multi-level quadratic DC-DC boost converter for photovoltaic systems using type-2 fuzzy logic technique-based MPPT approaches, *Heliyon*, Vol. 11, No. 3.
22. **Riahi, J., Nasri, H., Mami, A., Vergura, S. (2024).** Effectiveness of the fuzzy logic

control to manage the microclimate inside a smart insulated greenhouse, *Smart Cities*, Vol. 7, No. 3.

23. **Sharma, S., Sharma, S., Athaiya, A. (2017).** Activation functions in neural networks, *Towards Data Science*, Vol. 6, No. 12, pp. 310–316.
24. **Ramachandran, P., Zoph, B., Le, Q.V. (2017).** Searching for activation functions, *arXiv Preprint*, arXiv:1710.05941.
25. **Misra, D. (2019).** Mish: A self regularized non-monotonic activation function, *arXiv Preprint*, arXiv:1908.08681.
26. **Kiliçarslan, S., Celik, M. (2024).** Parametric RSigELU: A new trainable activation function for deep learning, *Neural Computing and Applications*, Vol. 36, pp. 1–13. doi: 10.1007/s00521-024-09538-9.
27. **Bozanic, D., Tešić, D., Marinković, D., Milić, A. (2021).** Modeling of neuro-fuzzy system as a support in decision-making processes, *Reports in Mechanical Engineering*, Vol. 2, No. 1, pp. 222–234.
28. **Taylan, O., Alkabaa, A.S., Alqabbaa, H.S., Pamukçu, E., Leiva, V. (2023).** Early prediction in classification of cardiovascular diseases with machine learning, neuro-fuzzy and statistical methods, *Biology*, Vol. 12, No. 1, pp. 117.
29. **Obianyo, J.I., Udeala, R.C., Alaneme, G.U. (2023).** Application of neural networks and neuro-fuzzy models in construction scheduling, *Scientific Reports*, Vol. 13, No. 1, pp. 8199.
30. **Abou Naaj, M., Mehdi, R., Mohamed, E.A., Nachouki, M. (2023).** Analysis of the factors affecting student performance using a neuro-fuzzy approach, *Education Sciences*, Vol. 13, No. 3, pp. 313.
31. **Gollapalli, M., Musleh, D., Ibrahim, N., Khan, M.A., Abbas, S., Atta, A., et al. (2022).** A Neuro-Fuzzy Approach to Road Traffic Congestion Prediction, *Computers, Materials & Continua*, Vol. 73, No. 1.
32. **Janosi, A., Steinbrunn, W., Pfisterer, M., Detrano, R. (1989).** Heart Disease [Dataset], *UCI Machine Learning Repository*. doi: 10.24432/C52P4X.
33. **Wolberg, W., Mangasarian, O., Street, N., Street, W. (1993).** Breast Cancer Wisconsin (Diagnostic) [Dataset], *UCI Machine Learning Repository*. doi: 10.24432/C5DW2B.
34. **CDC. (2023).** CDC Diabetes Health Indicators [Dataset], *UC Irvine*. doi:10.24432/c53919.
35. **Rahman, M.M., Davis, D.N. (2013).** Addressing the class imbalance problem in medical datasets, *International Journal of Machine Learning and Computing*, Vol. 3, No. 2, pp. 224.
36. **Ramirez, E., Melin, P., Prado-Arechiga, G. (2019).** Hybrid model based on neural networks, type-1 and type-2 fuzzy systems for 2-lead cardiac arrhythmia classification, *Expert Systems with Applications*, Vol. 126, pp. 295–307. doi:10.1016/j.eswa.2019.02.035.
37. **Sanchez, D., Melin, P., Castillo, O. (2020).** Comparison of particle swarm optimization variants with fuzzy dynamic parameter adaptation for modular granular neural networks for human recognition, *Journal of Intelligent & Fuzzy Systems*, Vol. 38, No. 3, pp. 3229–3252. doi: 10.3233/JIFS-191198.

*Article received on 31/10/2025; accepted on 15/12/2025.*

*\*Corresponding author is Patricia Melin.*