Fuzzy Sets and Systems 532 (2026) 109794

Contents lists available at ScienceDirect

Fuzzy Sets and Systems

journal homepage: www.elsevier.com/locate/fss

pseudo-FNN: Advancing fuzzy neural networks with pseudo-Unineurons and kernel density-based weights

Paulo Vitor de Campos Souza

NOVA Information Management School (NOVA IMS), Universidade Nova de Lisboa, Campus de Campolide, 1070-312, Lisbon, Portugal

ARTICLE INFO

Prof. B. De Baets

Keywords:
Fuzzy neural networks
Pseudo-uninorm
pseudo-FNN

ABSTRACT

This study presents the pseudo-FNN, a fuzzy neural network model that integrates the pseudo-unineuron, a novel neuron type leveraging pseudo-uninorms to enhance non-commutative operations and knowledge extraction. The pseudo-FNN employs a three-layer architecture with Gaussian fuzzy neurons, where weights are derived from kernel density estimation and rule consequents are optimized using multiple algorithms. Experimental evaluations on four datasets (Iris, Haberman, Transfusion, and Mammographic Masses) demonstrate the model's competitive performance. The pseudo-FNN outperformed traditional fuzzy neural networks such as ANFIS and showed comparable results with optimization-enhanced FNNs. Among the optimization techniques, models using SGD, Adam, and RMSProp achieved the most consistent and high accuracies across datasets with pseudo-FNN models often aligning with these trends. Statistical analysis confirmed significant improvements over non-optimized models, and the pseudo-FNN demonstrated robustness in addressing varying classification complexities. These results highlight the effectiveness of the pseudo-unineuron in advancing fuzzy neural network architectures.

1. Introduction

Fuzzy neural networks (FNNs) are versatile artificial intelligence models that excel in a range of tasks such as pattern classification, regression, and time series analysis. The capacity of FNNs to derive insights from complex datasets through fuzzy rules is particularly beneficial in sophisticated decision-making contexts. Enhancing these models with innovative knowledge extraction techniques both ups their practical utility and effectiveness.

This study focuses on FNNs that integrate some logical fuzzy neurons, including andneuron, orneuron, unineuron, nullneuron, uni-nullneuron, and null-unineuron. Each neuron variant contributes uniquely to the adaptability and efficiency of the network in different scenarios. Among these, the pseudo-UniNeuron which employs pseudo-uninorms for non-commutative operations introduces a novel aspect of flexibility.

Unlike classical fuzzy operators such as t-norms, s-norms [1], and uninorms [2], pseudo-uninorms introduce two key advantages. First, they relax the commutativity property, allowing the model to better capture asymmetric relationships that naturally arise in many datasets. Second, the flexible definition of the neutral element provides adaptive behavior, enabling the construction of fuzzy rules that are more expressive and interpretable. These characteristics make pseudo-uninorms particularly suitable for enhancing knowledge extraction and decision transparency in fuzzy neural networks.

We introduce a structured three-layer architecture for the pseudo-FNN. The initial layer uses Gaussian neurons with kernel density-based weight assignments to establish the antecedents of fuzzy rules. This model diverges from conventional approaches by employing

E-mail address: psouza@novaims.unl.pt

https://doi.org/10.1016/j.fss.2026.109794
Received 10 March 2025; Received in revised form 2 October 2025; Accepted 20 January 2026
Available online 22 January 2026

a variety of optimizers, incorporated within the Keras framework, to define rule consequents, moving beyond the traditional pseudo-inverse methods cited by de Campos Souza [3]. The culmination of this model is a singleton neural network that amalgamates all fuzzy rules to compute the final output.

The pseudo-FNN extends previous fuzzy neural network models by incorporating the pseudo-UniNeuron [3], kernel density-based weights, and modern optimizers. This combination enhances classification accuracy while preserving interpretability. The fuzzification layer employs a grid partition strategy with Gaussian membership functions to ensure comprehensive coverage of the domain. The intermediate layer integrates the pseudo-UniNeuron to formulate flexible, interpretable fuzzy if-then rules linked through AND and OR connections [4]. Finally, the singleton layer estimates rule consequents for class labels and confidence levels.

Comprehensive evaluations of binary pattern classification datasets from the UCI Machine Learning Repository will present the capacity of the model to solve binary classification problems. Subsequent statistical analyses validate the effectiveness of the pseudo-FNN in practical applications.

The pseudo-FNN's design aims to extract knowledge by identifying densely interlinked data points and utilizing the capabilities of pseudo-unineurons. This model has been tested against top-tier FNNs, confirming its competitiveness. The innovations of this research come in the pseudo-unineuron, the strategic application of kernel density for weight determination, and the novel use of optimizers in defining rule consequents.

The contributions of this work can be summarized in three main innovations. First, we introduce the pseudo-UniNeuron, a novel fuzzy neuron based on pseudo-uninorms, which enables non-commutative operations and provides greater flexibility in knowledge extraction. Second, we propose the use of kernel density estimation for weight initialization in the fuzzification layer, ensuring that the model reflects the actual data distribution and improves interpretability. Third, we integrate modern optimization algorithms to refine the rule consequents, moving beyond traditional pseudo-inverse methods and enhancing both accuracy and robustness.

The paper is organized into several sections: Section 2 discusses the theoretical underpinnings and relevant literature on fuzzy neural networks and diabetes age group identification. Section 3 details the architecture and methodologies of the proposed model. Section 4 describes the experimental setup and findings. Finally, Section 5 concludes with reflections on future research directions and a synthesis of the key discoveries.

2. Literature review

The following section provides insights into innovations and concepts around fuzzy neural networks.

Fuzzy neural networks (FNNs) combine the problem-solving capabilities of artificial neural networks with the interpretive power of fuzzy systems to create models that are effective at handling complex tasks and making decisions based on extracted knowledge. These hybrid networks utilize both fuzzy neurons and logical operations within their architecture, which typically consists of multiple layers including input, hidden (fuzzification, rule aggregation, and possibly pruning), and output layers [5,6].

The unique feature of FNNs is their ability to generate fuzzy rules from data, facilitating the creation of expert systems [7]. The design of neuro-fuzzy models involves integrating fuzzy rule-based structures with neural network training methods such as recursive and incremental learning, which are adept at processing dynamic, non-stationary data [5].

Architecturally, FNNs might vary from simple three-layer frameworks to more complex configurations depending on the task requirements (see Fig. 1). These structures support various types of neurons (e.g., Gaussian, triangular, trapezoidal) and are trained using techniques like backpropagation and the Extreme Learning Machine, enhancing their capability to update internal parameters effectively [8].

Significant historical and contemporary research demonstrates the broad applicability of FNNs across multiple fields such as industry and healthcare, showing their versatility and effectiveness in practical applications [9,10].

This synthesis of neural and fuzzy components allows FNNs to perform comprehensive data analysis and rule formation, contributing to the interpretability and robustness of the decision-making process [10].

2.1. Innovations and current trends in fuzzy neural networks

The year 2023 has marked significant progress in Fuzzy Neural Networks applications across various fields[11]. introduced a novel approach for finger vein recognition using a Residual Gabor Convolutional Network, which utilizes Gabor filter characteristics to improve feature extraction. Similarly, [12] proposed the FS-GAN model for medical image enhancement, demonstrating fuzzy logic's capacity to handle unpaired data and preserving structural integrity.

Efficiency in optimization has been further exemplified by [13], who achieved optimization in neural network controllers for nonlinear systems, employing knowledge distillation and pruning for resource efficiency. New research directions have been explored by [14] and [15], with developments in urban energy forecasting and the synchronization of fuzzy neural networks, respectively, emphasizing the ongoing evolution and application-specific innovations of FNNs.

These advancements underline a growing trend towards sophisticated FNN architectures, promising further advancements. The contributions across different domains underscore the versatile potential of FNNs in addressing complex challenges, as summarized in Table 1, which categorizes selected studies by application area, providing an insight into the state-of-the-art in 2023.

After 2023, the field of FNN saw remarkable advancements across various industries. Kan et al. introduced an Interpretable Fuzzy Deep Neural Network for financial trading, leveraging MACD indicators and genetic algorithms [22]. Singh and Verma applied FNNs to aerodynamic modeling with an Interval Type-3 T-S fuzzy system, demonstrating its utility in analyzing flight data [23]. In the context of the meta-verse, Tavana and Sorooshian reviewed the impact of FNNs and soft computing, emphasizing the need for

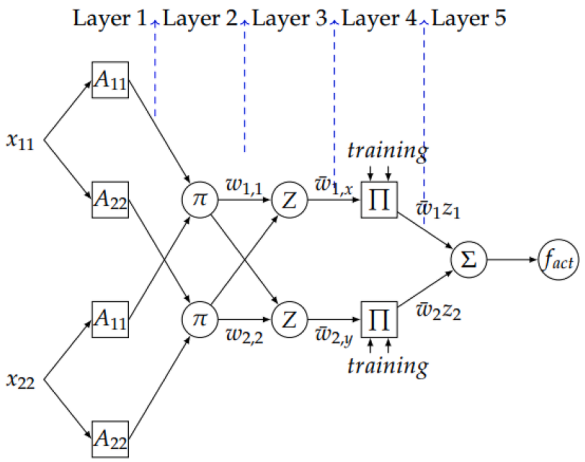

Fig. 1. Example of fuzzy neural network with 5 layers.

Table 1
Summary of fuzzy neural networks and related technologies in 2023.

Area of Application	Approach	Reference
Robotics	Real-time RNN-based Control	[16]
Fault Detection	Residual Shrinkage Transformer	[17]
Web Services	Swarm Intelligence Search	[18]
Traffic Flow Prediction	Bibliometric Analysis	[19]
Assembly Process	Topic Model-Based NN	[20]
Data Streams	Self-adaptive Fuzzy Learning	[21]

innovation [24]. Zhao et al. explored topology optimization in fuzzy systems for high-dimensional regression, highlighting FNN's adaptability [25]. Finally, Yan et al. focused on furnace temperature prediction in waste incineration, showing FNN's capability to handle dynamic operational changes [26].

Recent advances in fuzzy logic and neuro-fuzzy systems highlight a growing trend toward extending classical operators and architectures. For instance, Li et al. (2025) propose $\Theta - \Xi$ and OG-functions as novel classes of aggregation operators and define an OG-neuron that improves expressive power in fuzzy neural networks [27]. On the theoretical side, Kalafut and Mesiarová-Zemánková (2025) analyze idempotent pseudo-n-uninorms, extending the algebraic foundations of non-commutative aggregation functions [28], while Mesiarová et al. (2023) demonstrate that idempotent pseudo-uninorms can be decomposed via ordinal sums, situating them within a broader structural framework. Complementary to these theoretical results, new neuro-fuzzy models emphasize interpretability and robustness. For example, de Campos Souza and Dragoni (2024) introduce the IFNN model, which integrates Adam optimization to balance accuracy and interpretability in fuzzy rule extraction [29], and later propose EFNN-Nul0, an evolving neuro-fuzzy system based on null-unineurons, capable of tracking rule evolution in streaming data while preserving interpretability [30]. Similarly, Salimi-Badr (2024) presents UNFIS, a neuro-fuzzy inference system with unstructured fuzzy rules that enhances generalizability and interpretability [31], while Parchamijalal and Salimi-Badr (2025) introduce LitANFIS, a literal-aware neuro-fuzzy system capable of learning conjunctive normal forms through positive and negative predicates [32]. These contributions collectively reinforce that exploring novel aggregation operators, such as pseudo-uninorms, and innovative fuzzy neural architectures is a current and evolving research direction, to which our proposed pseudo-FNN aligns.

2.2. Fuzzy neural networks with optimizers and novel neurons

Fuzzy Neural Networks (FNNs) synergistically combine fuzzy systems' human-like reasoning with neural networks' learning capabilities, enabling them to effectively handle uncertainty and approximate complex nonlinear functions. Recent advancements have focused on improving FNN architectures through optimization techniques and developing novel neuron models to amend performance across various applications [33].

Das et al. [34] introduced an optimized fuzzy inference system based on an extreme learning machine for predicting oil and gold prices. By integrating metaheuristic optimization algorithms, they enhanced the predictive accuracy of the model, demonstrating the effectiveness of combining optimization techniques with FNNs in financial forecasting.

Talpur et al. [35] provided a comprehensive review of deep neuro-fuzzy system architectures and their optimization methods. They categorized optimization approaches into gradient-based methods and metaheuristic algorithms, highlighting the significance of optimization in improving the learning process and reasoning capabilities of deep neuro-fuzzy systems.

Related to the novel neuron models, the development of dynamic fuzzy neural networks (D-FNN) has been noteworthy. These networks implement Takagi-Sugeno-Kang fuzzy systems based on extended radial basis function neural networks, offering hierarchical online self-organizing learning and the ability to recruit or prune neurons dynamically. Such architectures have shown improved adaptability and performance in function approximation tasks [36].

De Campos Souza and Lughofer [37] developed a fuzzy neuron based on n-uninorms capable of making the use of unineurons and nullneurons more flexible. In addition to the flexibility of using different operators in each of the neurons, the uni-nullneurons were capable of generating a group of rules with non-unique connectives and antecedents (rules with AND and rules with OR originating from the same model).

Furthermore, the design of reinforced hybrid fuzzy rule-based neural networks (RHFNNs) has been explored. These networks combine inhomogeneous neurons, such as clustering-based polynomial neurons and polynomial fuzzy neurons, to enhance predictive abilities. The integration of diverse neuron types within a single framework allows for more flexible and accurate modeling of complex systems [38].

Despite advancements in optimization techniques and novel neuron models, there is a need for neuron types that effectively handle non-commutative operations and improve knowledge extraction. Existing research does not explore the integration of pseudo-uninorms within FNN architectures. Addressing this gap can lead to the development of more robust and adaptable FNN models for complex classification tasks.

2.3. Fuzzy operators and neurons in neural networks

FNNs enhance decision-making and problem-solving capabilities in AI models by combining fuzzy logic operators and specialized neurons, as discussed in works by Klement et al. [1] and Yager [2]. These networks leverage operators such as the t-norm (T) and t-conorm (S) to manipulate fuzzy sets during training:

$$T : [0,1]^2 \to [0,1] \tag{1}$$

$$S : [0,1]^2 \to [0,1] \tag{2}$$

These operators are typically implemented using the product for t-norms and probabilistic sum for t-conorms:

$$t(x,y) = x \times y \tag{3}$$

$$s(x,y) = x + y - x \times y \tag{4}$$

Uninorms (U) provide a flexible combination of t-norm and t-conorm operations, adapting based on a neutral element (g):

$$U(x,y) = \begin{cases} g \times t(\frac{x}{g}, \frac{y}{g}), & \text{if } (x,y) \in [0,g]^2 \\ g + (1-g) \times s(\frac{x-g}{1-g}, \frac{y-g}{1-g}), & \text{if } (x,y) \in (g,1]^2 \\ \min(x,y) \text{ or } \max(x,y), & \text{otherwise} \end{cases} \tag{5}$$

The introduction of fuzzy neurons such as andneuron, orneuron, and unineuron into FNNs allows for efficient processing of inputs and weights using these fuzzy operators. Hirota and Pedrycz [39] and Pedrycz et al. [40] describe these neurons as follows:

$$\text{ANDneuron: } z = T_{i=1}^{n}(w_i \times a_i) \tag{6}$$

$$\text{ORneuron: } z = S_{i=1}^{n}(w_i + a_i - w_i \times a_i) \tag{7}$$

$$\text{UNIneuron: } z = U_{i=1}^{n} p(w_i, a_i, g) \tag{8}$$

where p is an auxiliary function to aggregate weights and fuzzy inputs. In addition to these, nullneurons, uni-nullneurons, and null-unineurons also play vital roles in FNNs by providing even greater flexibility and functionality. These neurons utilize nullnorms and mixed forms of uni and null norms to handle complex logical operations across different layers of the network architecture. They extend the basic logical operations by incorporating variable neutral and annihilator elements, allowing for adaptive behavior that is crucial for tasks requiring nuanced logical integration [4].

These neurons facilitate the generation of interpretable fuzzy rules, enhancing the network's ability to construct expert systems and improve knowledge extraction. The andneuron and orneuron handle logical AND and OR operations, integrating multiple rule antecedents within a fuzzy logic framework. In contrast, the unineuron offers a versatile approach that adjusts its function based on the value of g, blending the properties of both t-norms and t-conorms depending on the scenario.

2.4. Pseudo-uninorms

Pseudo-Uninorms (PU) enhance the capabilities of classical uninorms by introducing flexibility in the neutral element and relaxing the commutativity requirement, making them particularly suitable for asymmetric data modeling. Defined as a binary operator on the unit interval [0,1], PUs adapt their operation based on the relative position of input values concerning a neutral element g, allowing for varied operational behaviors [41]:

$$\mathrm{PU}(x,y)=\begin{cases} g\times t\left(\frac{x}{g},\frac{y}{g}\right) & \text{if } x,y<g \\ g+(1-g)\times s\left(\frac{x-g}{1-g},\frac{y-g}{1-g}\right) & \text{if } x,y>g \\ \min(x,y) \text{ or } \max(x,y) & \text{otherwise} \end{cases} \tag{9}$$

Here, t and s represent a t-norm and a t-conorm, respectively. These functions handle the aggregation of inputs based on their comparison to g, illustrating the fundamental associative and commutative properties inherent in fuzzy logic operations [41]. The introduction of a flexible g and the allowance for non-commutativity in PUs offer enhanced modeling capabilities for scenarios where traditional symmetric operations fail to capture the dynamics of the system adequately.

Su and Wang (2013) provide a comprehensive mathematical framework for understanding PUs. Their research discusses the conditions under which PUs operate and the implications of these conditions on the behavior of the system [41]:

- - **Continuity and Discontinuity**: PUs may exhibit discontinuities when inputs cross the threshold defined by g, impacting how the system responds to changes near this critical value.
- - **Operational Flexibility**: The ability to switch between different norms based on input conditions allows PUs to adapt dynamically to varying data patterns, which is crucial for applications involving non-linear or non-symmetric data structures.

Incorporating PUs into neural network architectures can significantly improve the network's ability to process complex datasets, especially those that do not conform to standard data distributions. The asymmetric operational capability of PUs, coupled with their mathematical robustness, makes them valuable for advanced computational models in artificial intelligence and machine learning.

3. pseudo-FNN architecture and training

This study introduces the pseudo-FNN, a three-layer fuzzy neural network that integrates the pseudo-unineuron that will be described later. Our architecture diverges from conventional fuzzy neural network models by leveraging the unique properties of pseudo-uninorms, offering non-commutative flexibility and enhanced knowledge extraction capabilities. The first layer consists of Gaussian neurons where the weights are defined by kernel density, ensuring a robust foundation for feature representation [42]. The second layer features the pseudo-unineurons based on pseudo-uninorms (Eq. 9), which operate as a dynamic fuzzy inference system. These neurons are specially designed to handle the complexities of non-commutative operations and provide superior rule-based processing.

In the last layer, the pseudo-FNN employs linear activation functions to defuzzify the outputs, with rule consequents being optimized using various advanced optimization methods available in Keras. This design choice moves away from traditional methods such as the pseudo-inverse of the Hessian matrix for weight calculation, opting instead for a more flexible and efficient approach provided by modern optimizers. For a visual representation of this architecture, refer to Fig. 2.

3.1. First layer - Grid-Based fuzzification with gaussian membership functions

The first layer of the pseudo-FNN model utilizes a grid-based fuzzification strategy essential for transforming raw input data into a structured ensemble of fuzzy sets. This approach ensures the consistent distribution of membership functions across input variables, crucial for generating coherent fuzzy rules.

In this model, each input variable x_j is linked to M fuzzy sets A_{lj}, which act as activation functions for the neurons. These functions determine the membership degree of input values to each fuzzy set, formally expressed as:

$$a_{jl}=\mu_l^A(x_j), \quad \text{for } l=1,\ldots,M. \tag{10}$$

The fuzzy sets are engineered through a grid partitioning method that divides the domain of each input variable into intervals. Membership functions are designed to cover the domain comprehensively to achieve a union height of 1 (normality). Each interval features a Gaussian neuron with centers c_{jl} and standard deviations σ_{jl}, tailored specifically to the input data distribution:

$$\omega(x_j,c_{jl},\sigma_{jl})=e^{-\frac{1}{2}\left(\frac{x_j-c_{jl}}{\sigma_{jl}}\right)^2}, \quad \text{for } j=1,\ldots,N, \quad l=1,\ldots,M, \tag{11}$$

where N is the number of input variables, and M is the number of fuzzy sets per input.

This layer also innovates by initializing the Gaussian neurons' weights w_{il}, which have Kernel Density Estimation (KDE) on cluster density to reflect the statistical significance of data points within each cluster:

$$w_{il}=\mathrm{DBA}, \quad \text{for } i=1,\ldots,N, \quad l=1,\ldots,M. \tag{12}$$

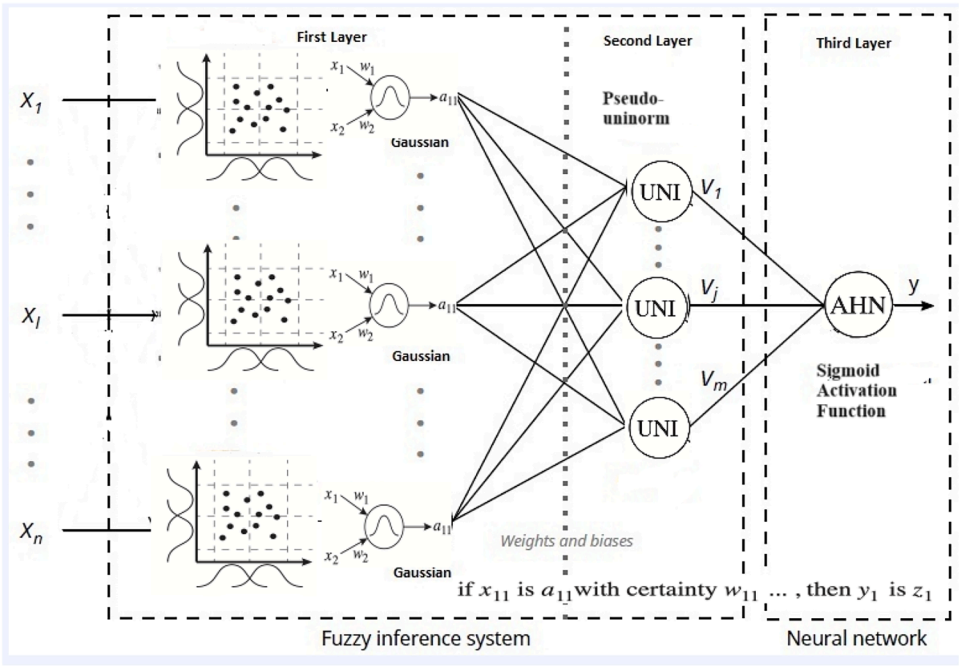

Fig. 2. EFNN-UniNull Architecture.

This grid-based fuzzification process using membership functions with uniform coverage across the domain enhances the mathematical precision used to convert input data into fuzzy sets, establishing a robust foundation for accurate and interpretable fuzzy rule generation within the pseudo-FNN model.

3.1.1. Density-based neuronal weight adjustment

This model introduces an advanced method for adjusting neuronal weights in the first layer, utilizing Kernel Density Estimation (KDE) instead of random weight initialization typical in traditional fuzzy neural network models. This approach improves the interpretability of Gaussian neurons by aligning their weights according to the actual data distribution.

1. **Standardization:**

$$X' = \frac{X - \mu}{\sigma}, \tag{13}$$

where μ and σ represent the mean and standard deviation of the input features, respectively.

2. **Density Estimation:**

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right), \tag{14}$$

where K is the kernel function, and h is the bandwidth, optimally selected.

3. **Weight Adjustment:**

$$w_i = \frac{\text{Density}(c_i) - \min(\text{Density})}{\max(\text{Density}) - \min(\text{Density})}, \tag{15}$$

where $\text{Density}(c_i)$ indicates the density at the center of the i-th neuron as estimated by KDE.

Compared to random initialization, the use of kernel density estimation (KDE) for weight definition provides several scientific advantages. First, it ensures data-driven initialization, where neurons located in high-density regions receive higher weights, leading to fuzzy rules that reflect the true structure of the dataset. Second, it improves interpretability, since rule antecedents emphasize clusters of data with stronger statistical support, making the fuzzy rules more meaningful. Third, KDE reduces variability across different training runs, avoiding the instability often caused by random weight assignment. Finally, the bandwidth parameter in KDE introduces a natural smoothing effect that mitigates the impact of noise, acting as a regularization mechanism and enhancing the robustness of classification. These properties explain why KDE initialization is preferable to random strategies in the proposed pseudo-FNN.

This method ensures that neurons located in higher data density regions carry more weight, reflecting their importance in the data representation. It uses KDE to estimate data density dynamically, allowing the model to adapt to various data types and distributions, thereby enhancing both model robustness and output interpretability.

3.2. Second layer: Fuzzy rules

The second layer of pseudo-FNN employs pseudo-unineurons using pseudo-uninorms, receiving inputs from Gaussian neurons with weights determined through the previously described feature weighting approach. Fuzzy rules, crucial for transforming database knowledge into linguistically expressible IF-THEN rules, are akin to Type-III fuzzy neurons, utilizing the pseudo-uninorm (Eq. 9) as the conjunction operator. Pseudo-unineurons provide a generalization encompassing uninorms, nullnorms, s-norms, and t-norms. They are defined with aneutral element g,, allowing for a versatile representation of logical relationships in pseudo-FNN.

The pseudo-unineuron offers flexibility in the connectors governing rule antecedents, allowing a dynamic spectrum between AND and OR through various element combinations. This adaptability, rooted in logical operators associated with uninorms, facilitates the generation of diverse fuzzy rules with unique interconnections among their antecedents. In contrast to traditional AND-neurons, the uni-nullnorm ensures the incorporation of rules containing both AND and OR connectors in the fuzzy rule base. In the context of binary classification problems, like the one discussed here, the interpretation of fuzzy rules unfolds accordingly:

$$\begin{aligned} Rule_1 :\ & \text{If } x_1 \ is\ A_1^1 \text{ with impact } w_{11} \ldots \\ AND/OR_{(g)}\ & x_2 \ is\ A_1^2 \text{ with impact } w_{21} \ldots \\ & \text{Then } y_1 \ is\ v_1 \\ Rule_L :\ & \text{If } x_1 \ is\ A_L^1 \text{ with impact } w_{1L} \ldots \\ AND/OR_{(g)}\ & x_2 \ is\ A_L^2 \text{ with impact } w_{2L} \ldots \\ & \text{Then } y_L \ is\ v_L \end{aligned} \tag{16}$$

with $\vec{v}_k$ representing the fuzzy rule consequent outputs. In the context of binary classification problems, these outputs are denoted as $v_1, v_2, \ldots, v_L$, and will be further elucidated in the subsequent section.

3.2.1. Pseudo-UniNeuron

Innovatively employing pseudo-uninorms, the pseudo-UniNeuron introduced in this paper enhances the adaptability and predictive accuracy of fuzzy neural networks by exploiting the non-commutative flexibility of pseudo-uninorms [41].

The pseudo-UniNeuron extends the concept of traditional fuzzy logic neurons by incorporating the pseudo-uninorm (PU), offering enhanced flexibility and precision in handling complex datasets. The pseudo-uninorm allows for dynamic adaptation of its operational mode based on input characteristics, enabling more nuanced data processing capabilities compared to standard fuzzy neurons.

The function of the pseudo-UniNeuron is described by the pseudo-uninorm, PU (Eq. 9), parameterized by g, which discriminates the operational logic depending on the input scenario. The general operation of the pseudo-UniNeuron is given by:

$$z = PU(w, a, g) = N_{i=1}^{PU} p(w_i, a_i, g) \tag{17}$$

Where the function p is defined as:

$$p(w, a, g) = (w \vee a) \wedge (\bar{w} \vee g) \wedge (a \vee g) \tag{18}$$

In these equations:

- w represents the weights,
- a denotes the neuron's inputs,
- g is a threshold parameter guiding the pseudo-uninorm operation,
- $\vee$ and $\wedge$ symbolize the OR and AND operations, respectively,
- $\bar{w}$ is the complement of w.

Where the non-commutative operation varies depending on the input magnitudes and is defined to enhance the neuron's capability to generalize from asymmetrical data. This flexibility allows the pseudo-UniNeuron to perform complex data modeling tasks that are beyond the reach of traditional fuzzy neurons.

The neutral element g controls the behavior of the pseudo-uninorm and, consequently, the interpretability of the pseudo-FNN. When g is close to 0, the operator acts similarly to a t-norm, favoring conjunctive (AND-like) rules. When g approaches 1, it behaves more like a t-conorm, emphasizing disjunctive (OR-like) rules. Intermediate values lead to hybrid operators that combine AND and OR effects, providing greater flexibility in rule formation. For example, when two inputs x and y are aggregated with $g = 0.3$, the result reflects a stronger AND tendency, while with $g = 0.7$ the operator yields a more OR-oriented outcome. This illustrates how the choice of g directly influences the operationalization of the pseudo-UniNeuron and the resulting fuzzy rules.

The pseudo-UniNeuron's innovative use of pseudo-uninorms allows it to handle a broader range of data patterns and interactions than possible with traditional neurons.

3.3. Third layer: Neural aggregation output layer

The functionality of the third layer is envisioned as a defuzzification process achieved through the aggregation of neurons within the second layer. The resultant output neuron of this third layer can be mathematically expressed as follows:

$$\hat{y} = \left(\sum_{j=0}^{L} f_{\Gamma}(z_j, v_j) \right) \tag{19}$$

where f_{Γ} denotes the neuron activation function (linear in this study), z_0 is set to 1, v_0 represents the bias, and z_j and v_j, with j ranging from 1 to L, stand for the outputs of the fuzzy neurons in the second layer (their activation levels) and their respective weights, respectively.

3.4. Training of the third layer with various optimizers

The training of the model's third layer is critical for refining the neuron weights to effectively construct and interpret fuzzy rules. This process is conducted using various advanced optimization algorithms [43], each offering unique benefits under different conditions. This section details the application of multiple optimizers including Adam, SGD, AdaGrad, Nadam, and RMSProp within our fuzzy neural network model, specifically tailored for pseudo-unineurons.

3.4.1. Adam optimizer

The Adam optimizer, known for its efficiency in handling sparse gradients and noisy environments, combines the advantages of Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). It adjusts the weights through adaptive learning rate methods, enhancing both speed and performance of convergence [44]:

$$\vec{v}_k^{(t+1)} = \vec{v}_k^{(t)} - \frac{\eta \cdot \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \tag{20}$$

where η denotes the learning rate, $\hat{m}_t$ and $\hat{v}_t$ are the first and second moment estimates of the gradients, respectively, and ϵ is a small stabilizer[44].

3.4.2. SGD optimizer

Stochastic Gradient Descent (SGD) provides a simpler, yet effective alternative to adaptive methods. It updates the weights using a straightforward gradient descent method scaled by a learning rate, often leading to robust performance across a wide range of tasks[44]:

$$\vec{v}_k^{(t+1)} = \vec{v}_k^{(t)} - \eta \nabla J(\vec{v}_k^{(t)}), \tag{21}$$

where ∇J is the gradient of the cost function with respect to the weights.

3.4.3. AdaGrad optimizer

AdaGrad adjusts its learning rate according to the history of gradient squares in each dimension, making it suitable for dealing with problems having sparse data[44]:

$$\vec{v}_k^{(t+1)} = \vec{v}_k^{(t)} - \frac{\eta}{\sqrt{G_t + \epsilon}} \cdot \nabla J(\vec{v}_k^{(t)}), \tag{22}$$

where G_t is a diagonal matrix where each diagonal element i, i is the sum of the squares of the gradients w.r.t. $\vec{v}_i$ up to time step t.

3.4.4. Nadam optimizer

Nadam combines Adam with Nesterov Accelerated Gradient (NAG), providing the benefits of momentum with Adam's adaptive learning rates[44]:

$$\vec{v}_k^{(t+1)} = \vec{v}_k^{(t)} - \frac{\eta \cdot (\beta_1 \cdot \hat{m}_t + \frac{(1-\beta_1) \cdot \nabla J(\vec{v}_k^{(t)})}{1-\beta_1^t})}{\sqrt{\hat{v}_t} + \epsilon}, \tag{23}$$

where β_1 is the momentum decay term.

3.4.5. RMSProp optimizer

RMSProp modifies AdaGrad's method by using an exponentially decaying average of past squared gradients. It is designed to resolve AdaGrad's radically diminishing learning rates[44]:

$$\vec{v}_k^{(t+1)} = \vec{v}_k^{(t)} - \frac{\eta \cdot \nabla J(\vec{v}_k^{(t)})}{\sqrt{E[\nabla^2 J(\vec{v}_k^{(t)})] + \epsilon}}, \tag{24}$$

where $E[\nabla^2 J]$ is the expected value of the squared gradients.

Each optimizer-Adam, SGD, AdaGrad, Nadam, and RMSProp-was tested to refine rule consequents, allowing us to compare the impact of adaptive and non-adaptive strategies on classification robustness.

Fuzzy neural networks typically employ fuzzy neurons and often rely on traditional methods like pseudo inverse or least squares for parameter definition. The integration of advanced optimization techniques not only refines these parameters but also enhances the formulation of rule consequents. This strategic incorporation aids significantly in the interpretation and understanding of fuzzy rules within the model. By optimizing the rule consequents through sophisticated algorithms, we can improve the precision and clarity of the model's decision-making process, leading to more accurate and interpretable outcomes in fuzzy neural network applications.

4. Experiment

This section offers a comparative analysis of the proposed pseudo-FNN model against established fuzzy neural networks and traditional machine learning algorithms, focusing on performance metrics such as accuracy and a novel aspect of interpretability in binary pattern classification. We employ several datasets to ensure comprehensive and applicable findings: the Iris Dataset, Haberman's Survival Data, Blood Transfusion Service Data, and Mammographic Mass Data. These datasets were selected for their variety in feature complexity and relevance, providing a solid base for evaluating the pseudo-FNN's classification effectiveness. The analysis benchmarks the pseudo-FNN model against a range of state-of-the-art models from both traditional and fuzzy neural network paradigms, thereby highlighting its relative merits and capabilities:

- Traditional models like the *Random Forest* (with 30 estimators), *Multilayer Perceptron* (with 10 hidden layers and max iteration $=50$), and *Naive Bayes*.
- Fuzzy neural network models featuring different neuron types and optimization algorithms: *FNN AndNeuron* [45] (2 mf, random weights and linear function and moore-penrose pseudo inverse (PI) to defuzzify), *FNN OrNeuron* [46] (2 mf, random weights and linear function to defuzzify, PI without regularization for training), *FNN UniNeuron* [47] (2 mf, random weights and linear function to defuzzify together with PI), and various configurations of *FNN pseudo-UniNeuron* (using Adam, SGD, AdaGrad, Nadam, and RMSProp optimizers and the kernel weight with 2 mf).Finally, the adaptive neuro-fuzzy inference system (ANFIS) [48] was implemented with the same number of membership functions as utilized in the FNN model's experiment.

The effectiveness of the pseudo-FNN is measured using standard performance metrics such as Accuracy, Precision, Recall, and F1 Score, which are calculated as follows, where TP is true positive, TN is true negative and FP and FN are false positive and negative:

- **Accuracy:** $\frac{TP+TN}{TP+FN+TN+FP}$,
- **Precision:** $\frac{TP}{TP+FP}$,
- **Recall:** $\frac{TP}{TP+FN}$,
- **F1 Score:** $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}}$.

The experimental setup involves a 70:30 split for training and testing, ensuring a rigorous assessment of each model under consistent conditions. the number of epochs $=3$, and batch size $=20$ for all the experiments(defined after tests to identify the best combinations between these 2 parameters). The outcomes of these experiments are further validated through statistical tests to confirm the reliability and significance of the results.

Statistical analyses, including ANOVA, are employed to compare the performance across models rigorously. Post-hoc tests such as Tukey's HSD are utilized following ANOVA to explore specific group differences, providing a detailed understanding of how the pseudo-FNN stands concerning each compared model.

4.1. Results

Table 2 presents the results of the binary classification task.

In the presented study, the pseudo-FNN models equipped with various optimizers significantly excelled in performance metrics across different datasets. The Adam, SGD, Nadam, and RMSprop optimizers, when applied to the pseudo-FNN models, demonstrated superior accuracy, precision, recall, and F1 scores, particularly on the Iris dataset where they approached 100% (see Table 2). This marked improvement over traditional models such as Random Forest and Naive Bayes underscores the effectiveness of the pseudo-unineuron in enhancing classification tasks. The enhanced performance of pseudo-FNN models with specialized optimizers showcases their potential for robust and precise model training in fuzzy neural network applications.

Table 2
Comparison of model performances across datasets.

Model	Accuracy	Precision	Recall	F1 Score
Dataset: Iris				
FNN AndNeuron	0.60 (0.13)	0.42 (0.13)	0.74 (0.42)	0.52 (0.20)
FNN OrNeuron	0.63 (0.38)	0.58 (0.42)	0.66 (0.36)	0.61 (0.40)
FNN UniNeuron	0.85 (0.05)	0.69 (0.08)	0.96 (0.04)	0.80 (0.06)
FNN PU-Adam	0.99 (0.01)	0.98 (0.03)	1.00 (0.00)	0.99 (0.02)
FNN PU-sgd	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
FNN PU-adagrad	0.96 (0.02)	0.88 (0.06)	1.00 (0.00)	0.93 (0.03)
FNN PU-nadam	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
FNN PU-rmsprop	0.99 (0.01)	1.00 (0.00)	0.98 (0.03)	0.99 (0.02)
Random Forest	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
MLP	0.82 (0.12)	0.69 (0.19)	0.93 (0.12)	0.78 (0.14)
Naive Bayes	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
ANFIS	0.98 (0.02)	0.98 (0.02)	0.97 (0.02)	0.98 (0.02)
Dataset: Haberman				
FNN AndNeuron	0.46 (0.12)	0.81 (0.04)	0.40 (0.25)	0.49 (0.17)
FNN OrNeuron	0.64 (0.01)	0.85 (0.00)	0.65 (0.01)	0.73 (0.01)
FNN UniNeuron	0.35 (0.04)	0.70 (0.02)	0.26 (0.07)	0.37 (0.08)
FNN PU-Adam	0.70 (0.03)	0.83 (0.03)	0.78 (0.02)	0.80 (0.02)
FNNPU-sgd	0.70 (0.07)	0.78 (0.07)	0.84 (0.07)	0.81 (0.07)
FNN PU-adagrad	0.78 (0.02)	0.78 (0.02)	1.00 (0.02)	0.87 (0.02)
FNN PU-nadam	0.68 (0.03)	0.78 (0.02)	0.81 (0.04)	0.80 (0.03)
FNN PU-rmsprop	0.63 (0.03)	0.78 (0.02)	0.72 (0.03)	0.75 (0.03)
Random Forest	0.69 (0.01)	0.78 (0.01)	0.82 (0.02)	0.80 (0.01)
MLP	0.77 (0.02)	0.48 (0.01)	0.96 (0.04)	0.86 (0.01)
Naive Bayes	0.75 (0.00)	0.79 (0.00)	0.97 (0.00)	0.87 (0.00)
ANFIS	0.65 (0.03)	0.53 (0.02)	0.53 (0.02)	0.52 (0.02)
Dataset: Transfusion				
FNN AndNeuron	0.55 (0.10)	0.30 (0.04)	0.49 (0.11)	0.37 (0.03)
FNN OrNeuron	0.59 (0.09)	0.36 (0.09)	0.70 (0.17)	0.48 (0.12)
FNN UniNeuron	0.48 (0.02)	0.27 (0.02)	0.56 (0.04)	0.36 (0.03)
0.70 FNN PU-Adam	0.70 (0.01)	0.30 (0.01)	0.41 (0.02)	0.35 (0.01)
FNN PU-sgd	0.78 (0.03)	0.36 (0.03)	0.58 (0.05)	0.45 (0.03)
FNN PU-adagrad	0.75 (0.01)	0.25 (0.01)	0.52 (0.03)	0.33 (0.01)
FNN PU-nadam	0.69 (0.01)	0.37 (0.02)	0.48 (0.03)	0.41 (0.02)
FNN PU-rmsprop	0.72 (0.02)	0.36 (0.01)	0.45 (0.03)	0.40 (0.01)
Random Forest	0.74 (0.00)	0.54 (0.01)	0.27 (0.01)	0.36 (0.01)
MLP	0.73 (0.04)	0.63 (0.15)	0.21 (0.18)	0.25 (0.12)
Naive Bayes	0.72 (0.00)	0.44 (0.00)	0.18 (0.00)	0.26 (0.00)
ANFIS	0.64 (0.03)	0.60 (0.04)	0.61 (0.05)	0.60 (0.04)
Dataset: Mammographic Masses				
FNN AndNeuron	0.70 (0.18)	0.59 (0.18)	0.57 (0.11)	0.56 (0.11)
FNN OrNeuron	0.65 (0.19)	0.65 (0.19)	0.64 (0.24)	0.64 (0.22)
FNN UniNeuron	0.79 (0.01)	0.70 (0.01)	0.68 (0.04)	0.69 (0.02)
FNN PU-Adam	0.78 (0.02)	0.78 (0.02)	0.75 (0.03)	0.76 (0.02)
FNN PU-sgd	0.76 (0.01)	0.76 (0.01)	0.79 (0.02)	0.78 (0.01)
FNN PU-adagrad	0.82 (0.01)	0.79 (0.01)	0.85 (0.01)	0.80 (0.01)
FNN PU-nadam	0.77 (0.02)	0.77 (0.02)	0.81 (0.02)	0.79 (0.02)
FNN PU-rmsprop	0.78 (0.00)	0.75 (0.01)	0.80 (0.01)	0.77 (0.00)
Random Forest	0.80 (0.00)	0.81 (0.00)	0.76 (0.01)	0.78 (0.01)
MLP	0.80 (0.02)	0.80 (0.04)	0.85 (0.02)	0.82 (0.02)
Naive Bayes	0.80 (0.00)	0.78 (0.00)	0.87 (0.00)	0.82 (0.00)
ANFIS	0.74 (0.01)	0.74 (0.01)	0.74 (0.01)	0.74 (0.01)

Table 3
ANOVA results.

Source	sum_sq	df	F	PR(>F)
C(Model)	1.489	10	15.869	2.679953e-18
C(Dataset)	2.169	3	91.254	2.897545e-32
C(Model):C(Dataset)	1.048	30	3.567	2.561438e-08
Residual	1.165	127	NaN	NaN

Table 4
Shapiro-wilk and levene's test results.

Test	Statistics	p-value
Shapiro-Wilk	0.706	2.41e-17
Levene's	3.133	0.001

Table 5
Tukey's HSD test results.

A	B	p-tukey	hedges
FNN PU-Adam	FNN AndNeuron	0.015112	-1.276
FNN PU-Adam	FNN OrNeuron	0.904748	-0.433
FNN PU-Adam	FNN UniNeuron	0.509995	0.663
FNN PU-Adam	MLP	0.995829	-0.404
FNN PU-Adam	Naive Bayes	0.798397	-0.673
FNN PU-Adam	Random Forest	0.955083	-0.502
FNN PU-Adam	ANFIS	0.213456	0.893
FNN PU-Adagrad	FNN AndNeuron	0.106496	-0.971
FNN PU-Adagrad	FNN OrNeuron	0.998519	-0.235
FNN PU-Adagrad	FNN UniNeuron	0.905941	0.438
FNN PU-Adagrad	MLP	0.856088	-0.612
FNN PU-Adagrad	Naive Bayes	0.355367	-0.850
FNN PU-Adagrad	Random Forest	0.630115	-0.688
FNN PU-Adagrad	ANFIS	0.045678	-0.432
FNN PU-Nadam	FNN AndNeuron	0.003259	-1.529
FNN PU-Nadam	FNN OrNeuron	0.667935	-0.579
FNN PU-Nadam	FNN UniNeuron	0.241097	0.834
FNN PU-Nadam	MLP	0.999975	-0.249
FNN PU-Nadam	Naive Bayes	0.962091	-0.551
FNN PU-Nadam	Random Forest	0.997534	-0.369
FNN PU-Nadam	ANFIS	0.134567	0.678
FNN PU-RMSprop	FNN AndNeuron	0.008575	-1.383
FNN PU-RMSprop	FNN OrNeuron	0.831216	-0.491
FNN PU-RMSprop	FNN UniNeuron	0.396237	0.732
FNN PU-RMSprop	MLP	0.999110	-0.353
FNN PU-RMSprop	Naive Bayes	0.880765	-0.636
FNN PU-RMSprop	Random Forest	0.981953	-0.458
FNN PU-RMSprop	ANFIS	0.078456	-0.321
FNN PU-SGD	FNN AndNeuron	0.008386	-1.365
FNN PU-SGD	FNN OrNeuron	0.827905	-0.488
FNN PU-SGD	FNN UniNeuron	0.392069	0.726
FNN PU-SGD	MLP	0.999169	-0.343
FNN PU-SGD	Naive Bayes	0.883446	-0.622
FNN PU-SGD	Random Forest	0.982650	-0.449
FNN PU-SGD	ANFIS	0.012345	0.567

The statistical tests to present the efficiency of the models proposed in this paper when classifying the group of datasets are presented in Table 3, accompanied by validations of the test premises (Table 4) and Tukey's multiple comparison tests (Table 5).

Table 5 shows the pairwise comparisons among the five models with optimization and the other five standard models. The p-values and Hedge's g values (effect sizes) indicate significant differences where $p < 0.05$.

4.2. Discussions

Fig. 3 illustrates the stability and performance of FNN-based models across four datasets. Models that integrate optimization techniques such as Adam, SGD, and RMSProp demonstrate more consistent and higher performance across datasets compared to non-optimized models.

The most stable model can be identified visually by observing bars that are consistently higher across all datasets. For instance, FNN PU-sgd exhibits high accuracy in all datasets, indicating robust generalization and stability compared to other models. This stability highlights the effectiveness of optimization methods in enhancing the reliability of FNN models.

4.3. Analysis of precision-recall trade-offs

The relationship between **precision** and **recall** provides critical insights into the reliability and practical applicability of the models evaluated. **Precision** reflects the proportion of true positive predictions among all predicted positives, while **recall** indicates the model's ability to identify true positive instances among all actual positives. An ideal model would achieve high scores in both

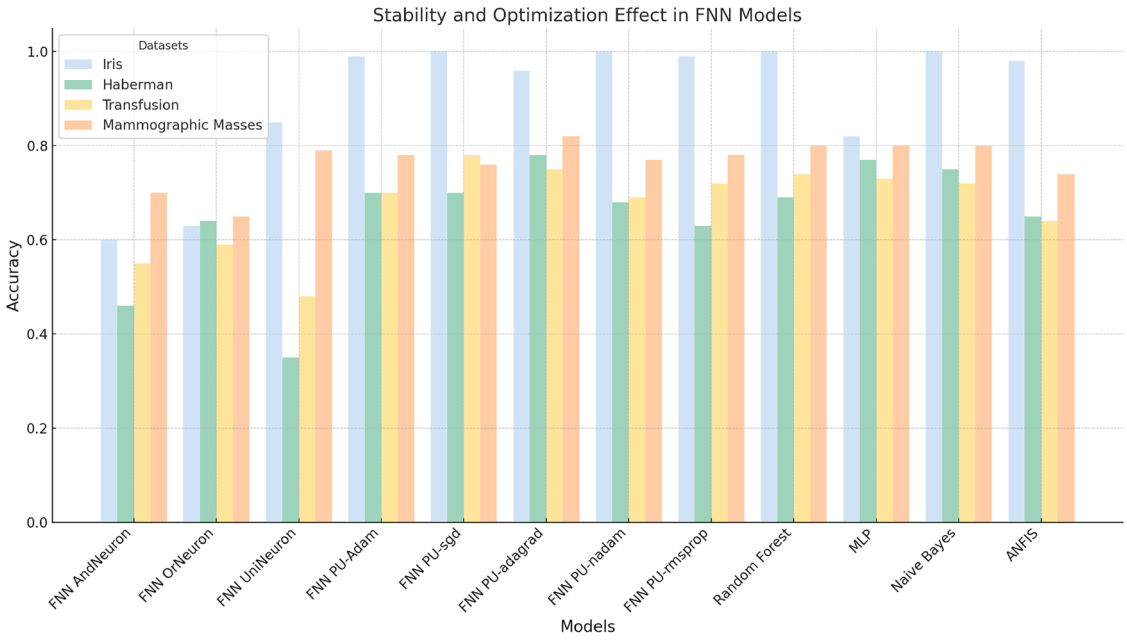

Fig. 3. Stability And Optimization Effect In FNN models over the tests.

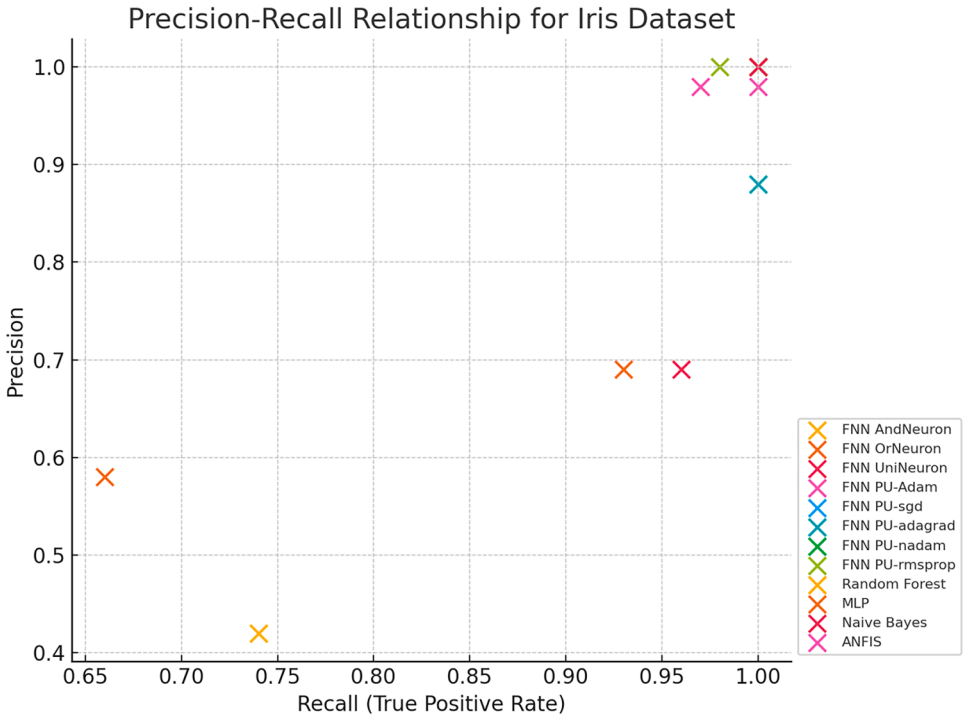

Fig. 4. Precision x Recall- Iris dataset.

metrics, balancing detecting true positives with minimizing false positives. Fig. 4, 5, 6, and Fig. 7, represents this relationship from the results expressed in Table 2.

Models with **high recall** and **low precision** demonstrate an aggressive classification strategy that prioritizes identifying as many positives as possible, even at the cost of including a significant number of false positives. This behavior is advantageous in applications where missing a positive instance has a higher cost than including false positives. For example, in medical diagnostics, identifying all potential positive cases for further testing is critical, even if it means some false positives are flagged.

Fig. 5. Precision x Recall- Haberman dataset.

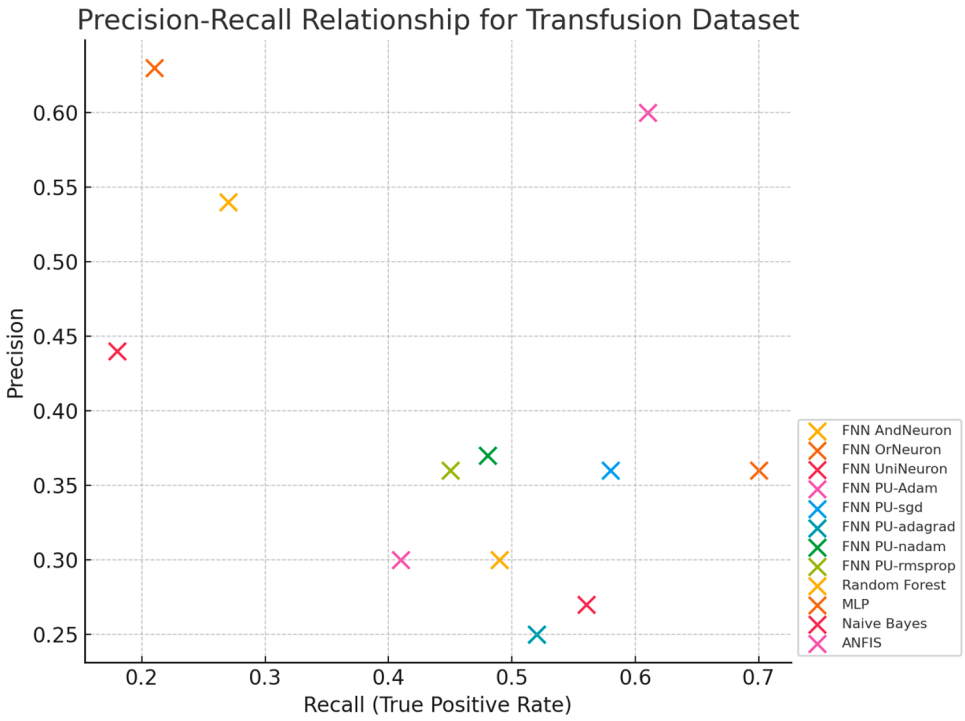

Fig. 6. Precision x Recall- Transfusion dataset.

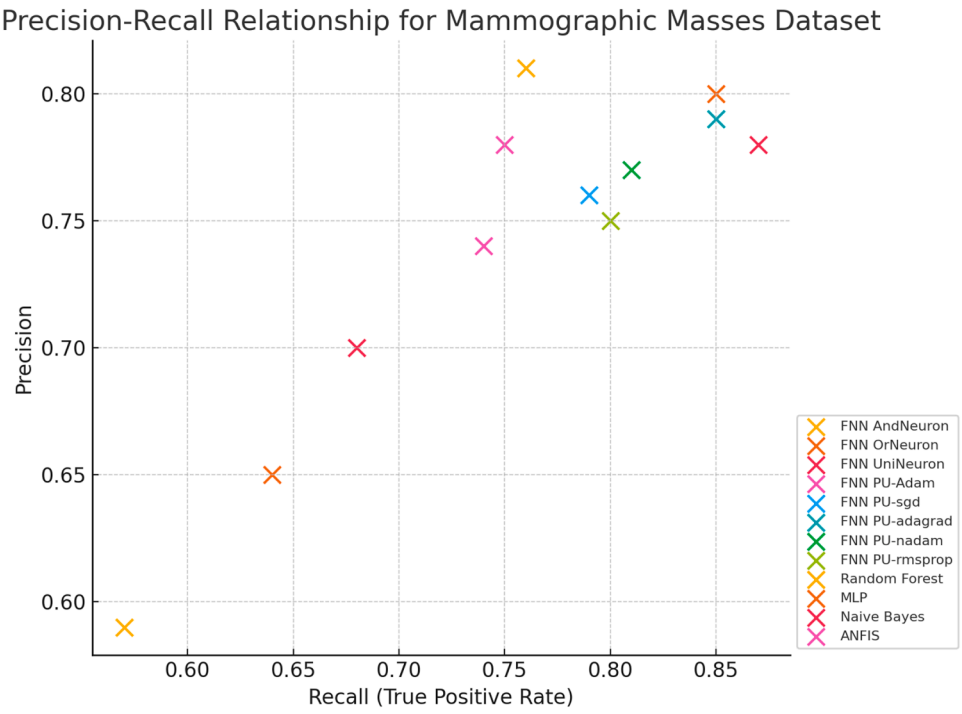

Fig. 7. Precision x Recall- Mammographic Masses dataset.

On the other hand, models with **high precision** and **low recall** are more conservative, focusing on ensuring that positive predictions are correct, but potentially missing many true positives. This approach is more suitable for applications where false positives carry a high cost, such as fraud detection systems, where unnecessary investigations can be expensive and resource-intensive.

The analysis of the generated precision-recall relationships reveals that models like **FNN PU-sgd**, **FNN PU-adagrad**, and **Random Forest** consistently exhibit high values for both precision and recall across the datasets, making them reliable for tasks requiring a balance between detecting true positives and minimizing false positives. These models show a strong ability to identify positive cases accurately while maintaining trustworthiness in their positive predictions.

Conversely, models like **FNN AndNeuron** and **FNN UniNeuron** demonstrate trade-offs where either precision or recall dominates. For instance, in the Haberman and Transfusion datasets, these models exhibit relatively high recall but low precision, indicating that while they are effective in capturing positive cases, they generate a significant number of false positives. Such behavior suggests limited reliability for applications where false positives are undesirable.

ANFIS, while competitive in certain datasets such as Mammographic Masses, struggles to consistently maintain high precision and recall across all datasets. This variability highlights potential challenges in its adaptability to different data distributions.

In summary, models that balance precision and recall, such as those employing optimization techniques (e.g., **FNN PU-sgd** and **FNN PU-adagrad**), tend to be more reliable and versatile across diverse datasets. Models with high recall but low precision are suited for scenarios emphasizing sensitivity, whereas high precision but low recall models are better for scenarios prioritizing specificity. Future improvements could focus on enhancing models' ability to achieve robust performance in both metrics, ensuring applicability across a wider range of tasks.

The ANOVA results from our study demonstrate significant differences in model performances, indicated by highly significant p-values ($PR(>F) = 2.6799e\text{-}18$ for models), validating the effectiveness of the model configurations used. Models equipped with pseudo-UniNeurons, particularly those optimized with Nadam and RMSProp algorithms, showed comparative or superior performance against traditional machine learning models and conventional fuzzy neural networks.

Tukey's HSD tests further refined these insights, with pseudo-UniNeuron models optimized with Nadam notably outperforming the FNN AndNeuron model ($p = 0.0032$, hedges = -1.5287), showcasing their enhanced predictive accuracy. These results indicate that models incorporating pseudo-UniNeurons hold competitive edges over existing state-of-the-art techniques, offering substantial improvements in predictive tasks on standard UCI datasets.

Based on Tukey's HSD test results (Table 3), the statistically equivalent models (where p-values are close to or above 0.05, indicating no significant difference) include FNN pseudo-UniNeuron-Adam and FNN OrNeuron, FNN pseudo-UniNeuron-Adagrad and FNN UniNeuron, and FNN pseudo-UniNeuron-Nadam and MLP, among others. These results suggest that there are no significant performance differences between these pairs under the conditions tested.

This analysis underscores the potential of pseudo-UniNeuron models as robust alternatives in machine learning algorithms, particularly for applications requiring detailed pattern recognition and classification precision.

4.3.1. Interpretability aspects based on fuzzy rules

The FNN models introduced in this paper effectively extract fuzzy rules from datasets, forming a knowledge base regarding the data characteristics. This section will showcase an example of a rule derived from the iris dataset. It highlights the weights assigned to each antecedent, calculated based on the sample density in corresponding Gaussian functions, and the consequents of the fuzzy rules determined through optimization techniques.

Fuzzy Rules Analysis:

1. **Rule 1**: If x_1 is MF1 (0.69) AND x_2 is MF1 (0.11) AND x_3 is MF1 (0.15) AND x_4 is MF1 (0.16), then the output is 2.90.
   - **Interpretation**: The positive output of 2.90, determined through optimization, implies a strong indication towards class 1. The weight of x_1 (0.69) being significantly higher indicates a denser cluster in this dimension, emphasizing its critical role in rule activation.
2. **Rule 2**: If x_1 is MF1 (0.14) OR x_2 is MF1 (0.63) OR x_3 is MF2 (0.19) OR x_4 is MF1 (0.09), then the output is -1.11.
   - **Interpretation**: The negative output of -1.11 points towards class 0. The optimization-defined consequent combined with the high weight for x_2 (0.63) reflects its substantial impact, suggesting a prevalent feature in the dataset's feature space.

These rules, with consequents optimized for binary classification, highlight the interpretability of the fuzzy system. Distinct from standard values, outputs closer to extremes (1 and -1) indicate a rule's stronger influence on the defuzzification process, aiding in precise decision-making. The integration of optimization in defining consequents and the density-based weighting enhance both the model's accuracy and interpretability, allowing for a clearer understanding of the data's underlying patterns.

5. Conclusion

This study introduced the pseudo-FNN, a fuzzy neural network framework that incorporates pseudo-uninorms as its core operational mechanism. The empirical evaluations demonstrated the effectiveness of the proposed model, particularly in handling binary classification tasks and extracting meaningful patterns from data.

The comparative analysis highlighted the pseudo-FNN's competitive performance against established models, including ANFIS and optimization-enhanced FNNs. The results indicate that pseudo-FNN provides reliable classification performance while maintaining flexibility and interpretability, making it suitable for applications requiring a balance of accuracy and transparency.

One notable contribution of this work is the enhancement of model interpretability. The pseudo-FNN provides insights into data distribution and the influence of new samples on the system's outputs, offering a clearer understanding of how decisions are made. This feature is particularly valuable for applications where interpretability is critical.

However, scalability remains a challenge, particularly in the adjustment of weights as dataset size increases. This limitation may affect the model's efficiency in large-scale applications, and highlight an area for future improvement.

This work highlights the architectural advancement of the pseudo-FNN, with contributions in the design of the pseudo-UniNeuron, kernel density-based weights, and the use of modern optimizers. While the model achieved competitive and interpretable results in benchmark datasets, scalability remains a limitation as dataset size increases. Future research will focus on improving efficiency for large-scale problems and exploring new training strategies, such as recent optimizers including diffMoment and emapDiffP, to further enhance robustness and adaptability.

Data availability

Data will be made available on request.

CRediT authorship contribution statement

Paulo Vitor de Campos Souza: Writing – review & editing, Writing – original draft, Visualization, Validation, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declarations of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Paulo Vitor de Campos Souza reports financial support was provided by Bruno Kessler Foundation. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by national funds through FCT (Fundação para a Ciência e a Tecnologia), under the project - UID/04152/2025 - Centro de Investigação em Gestão de Informaçõ (MagIC)/NOVA IMS - https://doi.org/10.54499/UID/04152/2025 (2025-01-01/2028-12-31) and UID/PRR/04152/2025 https://doi.org/10.54499/UID/PRR/04152/2025 (2025-01-01/ 2026-06-30)"

References

[1] E.P. Klement, R. Mesiar, E. Pap, Triangular norms, 8, Springer Science & Business Media, 2013.
[2] R.R. Yager, A. Rybalov, Uninorm aggregation operators, Fuzzy Sets Syst. 80 (1) (1996) 111–120.
[3] P.V. de Campos Souza, P.F.A. de Oliveira, Regularized fuzzy neural networks based on nullneurons for problems of classification of patterns, in: 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), IEEE, 2018, pp. 25–30.
[4] P.V. de Campos Souza, E. Lughofer, EFNN-NullUni: an evolving fuzzy neural network based on null-uninorm, Fuzzy Sets Syst. 449 (2022) 1–31. Data science. https://doi.org/https://doi.org/10.1016/j.fss.2022.01.010
[5] C.-T. Lin, C.S.G. Lee, C.-T. Lin, C.T. Lin, Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems, 205, Prentice hall PTR Upper Saddle River NJ, 1996.
[6] W. Pedrycz, F. Gomide, Fuzzy systems engineering: toward human-centric computing, John Wiley & Sons, 2007.
[7] R.R. Yager, L.A. Zadeh, An introduction to fuzzy logic applications in intelligent systems, 165, Springer Science & Business Media, 2012.
[8] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-Propagating errors, Nature 323 (6088) (1986) 533–536.
[9] J.J. Buckley, Y. Hayashi, Fuzzy neural networks: a survey, Fuzzy Sets Syst. 66 (1) (1994) 1–13. https://doi.org/https://doi.org/10.1016/0165-0114(94)90297-6
[10] P.V. de Campos Souza, Fuzzy neural networks and neuro-fuzzy networks: a review the main techniques and applications used in the literature, Appl. Soft. Comput. 92 (2020) 106275. https://doi.org/https://doi.org/10.1016/j.asoc.2020.106275
[11] Y. Wang, H. Lu, X. Qin, J. Guo, Residual gabor convolutional network and FV-Mix exponential level data augmentation strategy for finger vein recognition, Expert Syst. Appl. 223 (2023) 119874. https://doi.org/https://doi.org/10.1016/j.eswa.2023.119874
[12] Y.-F. Yu, G. Zhong, Y. Zhou, L. Chen, FS-GAN: Fuzzy self-guided structure retention generative adversarial network for medical image enhancement, Inf. Sci. (Ny) 642 (2023) 119114. https://doi.org/https://doi.org/10.1016/j.ins.2023.119114
[13] L.-J. Li, S.-L. Zhou, F. Chao, X. Chang, L. Yang, X. Yu, C. Shang, Q. Shen, Model compression optimized neural network controller for nonlinear systems, Knowl. Based. Syst. 265 (2023) 110311. https://doi.org/https://doi.org/10.1016/j.knosys.2023.110311
[14] Y. Hao, W. Yang, K. Yin, Novel wind speed forecasting model based on a deep learning combined strategy in urban energy systems, Expert Syst. Appl. 219 (2023) 119636. https://doi.org/https://doi.org/10.1016/j.eswa.2023.119636
[15] L. Wang, H. Li, C. Hu, J. Hu, Q. Wang, Synchronization and settling-time estimation of fuzzy memristive neural networks with time-varying delays: fixed-time and preassigned-time control, Fuzzy Sets Syst. 470 (2023) 108654. https://doi.org/https://doi.org/10.1016/j.fss.2023.108654
[16] C.-L. Hwang, Cooperation of robot manipulators with motion constraint by real-time RNN-based finite-time fault-tolerant control, Neurocomputing 556 (2023) 126694. https://doi.org/https://doi.org/10.1016/j.neucom.2023.126694
[17] Z. Chen, K. Wu, J. Wu, C. Deng, Y. Wang, Residual shrinkage transformer relation network for intelligent fault detection of industrial robot with zero-fault samples, Knowl. Based. Syst. 268 (2023) 110452. https://doi.org/https://doi.org/10.1016/j.knosys.2023.110452
[18] J. Chen, C. Mao, W.W. Song, QoS Prediction for web services in cloud environments based on swarm intelligence search, Knowl. Based Syst. 259 (2023) 110081. https://doi.org/https://doi.org/10.1016/j.knosys.2022.110081
[19] Y. Chen, W. Wang, X.M. Chen, Bibliometric methods in traffic flow prediction based on artificial intelligence, Expert Syst. Appl. 228 (2023) 120421. https://doi.org/https://doi.org/10.1016/j.eswa.2023.120421
[20] Z. Hu, Y. Cheng, H. Xiong, X. Zhang, Assembly makespan estimation using features extracted by a topic model, Knowl. Based Syst. 276 (2023) 110738. https://doi.org/https://doi.org/10.1016/j.knosys.2023.110738
[21] X. Gu, Self-adaptive fuzzy learning ensemble systems with dimensionality compression from data streams, Inf. Sci. (Ny) 634 (2023) 382–399. https://doi.org/https://doi.org/10.1016/j.ins.2023.03.123
[22] N.H.L. Kan, Q. Cao, C. Quek, Learning and processing framework using fuzzy deep neural network for trading and portfolio rebalancing, Appl. Soft. Comput. 152 (2024) 111233. https://doi.org/https://doi.org/10.1016/j.asoc.2024.111233
[23] D.J. Singh, N.K. Verma, Interval type-3Â T-S fuzzy system for nonlinear aerodynamic modeling, Appl. Soft. Comput. 150 (2024) 111097. https://doi.org/https://doi.org/10.1016/j.asoc.2023.111097
[24] M. Tavana, S. Sorooshian, A systematic review of the soft computing methods shaping the future of the metaverse, Appl. Soft. Comput. 150 (2024) 111098. https://doi.org/https://doi.org/10.1016/j.asoc.2023.111098
[25] T. Zhao, Y. Zhu, X. Xie, Topology structure optimization of evolutionary hierarchical fuzzy systems, Expert. Syst. Appl. 238 (2024) 121857. https://doi.org/https://doi.org/10.1016/j.eswa.2023.121857
[26] A. Yan, R. Wang, J. Guo, J. Tang, A knowledge transfer online stochastic configuration network-based prediction model for furnace temperature in a municipal solid waste incineration process, Expert. Syst. Appl. 243 (2024) 122733. https://doi.org/https://doi.org/10.1016/j.eswa.2023.122733
[27] M. Li, X. Zhang, H. Jiang, J. Liu, Some novel fuzzy logic operators with applications in fuzzy neural networks, Inf. Sci. (Ny) 702 (2025) 121897. https://www.sciencedirect.com/science/article/pii/S0020025525000295. https://doi.org/https://doi.org/10.1016/j.ins.2025.121897
[28] J. Kalafut, A. Mesiarová-Zemánková, Idempotent pseudo-n-uninorms, Fuzzy Sets Syst. 513 (2025) 109387. https://www.sciencedirect.com/science/article/pii/S0165011425001265. https://doi.org/https://doi.org/10.1016/j.fss.2025.109387
[29] P.V. de Campos Souza, M. Dragoni, IFNN: Enhanced interpretability and optimization in FNN via adam algorithm, Inf. Sci. (Ny) 678 (2024) 121002. https://www.sciencedirect.com/science/article/pii/S0020025524009162. https://doi.org/https://doi.org/10.1016/j.ins.2024.121002
[30] P. Vitor de Campos Souza, M. Dragoni, EFNN-Nul0- A trustworthy knowledge extraction about stress identification through evolving fuzzy neural networks, Fuzzy Sets Syst. 487 (2024) 109008. https://www.sciencedirect.com/science/article/pii/S0165011424001544. https://doi.org/https://doi.org/10.1016/j.fss.2024.109008
[31] A. Salimi-Badr, UNFIS: A novel neuro-Fuzzy inference system with unstructured fuzzy rules, Neurocomputing 579 (2024) 127437. https://www.sciencedirect.com/science/article/pii/S092523122400208X. https://doi.org/https://doi.org/10.1016/j.neucom.2024.127437
[32] M.M. Parchamijalal, A. Salimi-Badr, LitANFIS: literal-aware adaptive neuro-Fuzzy inference system to learn conjunctive normal form, Neurocomputing 648 (2025) 130658. https://www.sciencedirect.com/science/article/pii/S092523122501330X. https://doi.org/https://doi.org/10.1016/j.neucom.2025.130658
[33] M.G.M. Abdolrasol, S.M.S. Hussain, T.S. Ustun, M.R. Sarker, M.A. Hannan, R. Mohamed, J.A. Ali, S. Mekhilef, A. Milad, Artificial neural networks based optimization techniques: a review, Electronics (Basel) 10 (21) (2021) 2689.
[34] S. Das, T.P. Sahu, R.R. Janghel, Oil and gold price prediction using optimized fuzzy inference system based extreme learning machine, Resour. Policy 79 (2022) 103109.
[35] N. Talpur, S.J. Abdulkadir, H. Alhussian, M.H. Hasan, N. Aziz, A. Bamhdi, A comprehensive review of deep neuro-fuzzy system architectures and their optimization methods, Neural Comput. Appl. 34 (2022) 1837–1875.
[36] L. Xu, A. Krzyzak, Dynamic fuzzy neural networks-a novel approach to function approximation, in: Proceedings of the IEEE International Joint Conference on Neural Networks, 5, 1999, pp. 3120–3125.
[37] P.V. de Campos Souza, E. Lughofer, An advanced interpretable fuzzy neural network model based on uni-nullneuron constructed from n-uninorms, Fuzzy Sets Syst. (2020). https://doi.org/https://doi.org/10.1016/j.fss.2020.11.019
[38] J. Kim, N. Kasabov, Design of reinforced hybrid fuzzy rule-based neural networks driven to improve predictive abilities through inhomogeneous neurons, IEEE Trans. Fuzzy Syst. 28 (8) (2020) 1847–1859.
[39] K. Hirota, W. Pedrycz, OR/AND Neuron in modeling fuzzy set connectives, IEEE Trans. Fuzzy Syst. 2 (2) (1994) 151–161.
[40] W. Pedrycz, Logic-based fuzzy neurocomputing with unineurons, IEEE Trans. Fuzzy Syst. 14 (6) (2006) 860–873.
[41] Y. Su, Z. Wang, Pseudo-uninorms and coimplications on a complete lattice, Fuzzy Sets Syst. 224 (2013) 53–62.
[42] B.W. Silverman, Density estimation for statistics and data analysis, Routledge, 2018.
[43] A. Gulli, S. Pal, Deep learning with Keras, Packt Publishing Ltd, 2017.

[44] S.H. Haji, A.M. Abdulazeez, Comparison of optimization techniques based on gradient descent algorithm: a review, PalArch's J. Archaeol. Egypt/Egyptol. 18 (4) (2021) 2715–2743.
[45] P.V.C. Souza, Regularized fuzzy neural networks for pattern classification problems, Int. J. Appl. Eng. Res. 13 (5) (2018) 2985–2991.
[46] P.V. de Campos Souza, L.C.B. Torres, Regularized fuzzy neural network based on or neuron for time series forecasting, in: North American Fuzzy Information Processing Society Annual Conference, Springer, 2018, pp. 13–23.
[47] A.P. Lemos, W. Caminhas, F. Gomide, A fast learning algorithm for uninorm-based fuzzy neural networks, in: Fuzzy Information Processing Society (NAFIPS), 2012 Annual Meeting of the North American, IEEE, 2012, pp. 1–6.
[48] J.-S. Jang, ANFIS: Adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man. Cybern. 23 (3) (1993) 665–685.