Generalizing the Antecedent Layer of ANFIS: A Performance Analysis Under Classes of Fuzzy Aggregators

Bruna Novack, Juliano Buss, Gabriel Silva, Renata Reiser
PPGC - Programa de Pós-Graduação em Computação
Universidade Federal de Pelotas (UFPel)
Pelotas-RS, Brasil
{bcdnovack, jsbuss, grosilva, reiser}@inf.ufpel.edu.br

Helida Santos, Giancarlo Lucca
Centro de Ciências Computacionais
Universidade Federal do Rio Grande (FURG)
Rio Grande-RS, Brasil
{helida, giancarlo.lucca}@furg.br

***Abstract*—Machine Learning, a branch of Artificial Intelligence, allows systems to infer patterns and make data-driven decisions without explicit programming. Among its approaches, supervised learning and Neural Networks are fundamental to solving complex problems, from medical diagnoses to banking security. Fuzzy Logic complements this area in decision-making involving uncertain and imprecise information formalized by linguistic variables, from modeling to computational simulation of human reasoning. Neuro-fuzzy systems foster the integration of these technologies by exploring hybrid models that combine fuzzy inference with neural network learning. A prominent example is ANFIS (Adaptive Neuro-Fuzzy Inference System), which uses the Takagi-Sugeno method to model complex, non-linear systems using processing layers and automatic rule adjustment. In this context, this work proposes a generalization of the ANFIS architecture. The generalized architecture component analyzes the network's performance by applying different classes of fuzzy aggregators during the input data fuzzification step. For validation, we use specific metrics to evaluate the results of the proposed case studies.**

***Index Terms*—Fuzzy Logic, ANFIS generalization, Fuzzy Aggregators, Machine Learning, Neural Networks.**

I. Introduction

Machine Learning (ML) [1] is increasingly present in contemporary society, driven by technological advancements in various fields of knowledge. From recommendations on streaming platforms [2] to water desalination [3], fake images detection [4], air quality prediction [5] and others.

Among the various ML approaches, Neural Networks (NNs) [6] stand out, playing a significant role in the development of intelligent algorithms. The diversity of current application areas includes detecting potential rain-induced landslides [7], supporting breast cancer diagnosis [8], and even predicting stock behavior in the financial market [9].

Fuzzy Logic (FL) [10] provides a logical framework for the systematic modeling of uncertainty, offering benefits for the development of science and computational technologies. By addressing the uncertainty inherent in human reasoning, FL uses linguistic variables associated with the membership relations of Fuzzy Sets (FSs), enabling a flexible and realistic representation of complex phenomena.

Fuzzy inference systems can provide interpretations of complex, imprecise processes through qualitative representations and inference rules established by experts, derived from numerical data or statistical methods. The process begins by fuzzifying non-fuzzy inputs (numerical data), typically derived from measurements or observations, and mapping them to fuzzy input sets. Next, the fuzzy inference mechanism is executed to activate fuzzy rules, expressed as linguistic statements. Thus, operations are performed with fuzzy sets, including rule antecedent matching, implication, and fuzzy inference. The fuzzy input sets, related to the rule antecedents, and the output set, associated with the consequent, are evaluated to identify the rules to be applied. After generating the fuzzy output set, the defuzzification step follows, in which the qualitative information produced by the inference system is transformed into a numerical value that defines the result [11].

In this context, the Adaptive Neuro-Fuzzy Inference System (ANFIS) [12] is designed as a hybrid solution that combines the adaptive learning capabilities of neural networks with the approximate reasoning provided by FL, using the Takagi-Sugeno inference method [13]. ANFIS provides a clear interpretation of how various inputs affect the output in a system involving complex, and non-necessarily linear input-output data relationships. The ANFIS architecture considers five layers, starting with the fuzzification of the input data. Following this, there is the calculation and normalization of rule weights, the selection of rule consequences, and finally, the aggregation that generates the result after the learning process. This hybrid architecture supports automatic adjustment of fuzzy rules and Membership Functions (MFs), thereby capturing patterns in the input data more precisely through supervised learning.

As an example of ANFIS application, the work of Serri [14] stands out, using ANFIS as a tool to control nonlinear systems, specifically for the efficient control of water levels in a didactic plant. In this study, ANFIS replaces the Proportional-

This research was partially supported by Brazilian funding agencies: CAPES, CNPq (309160/2019-7, 3305805/2021-5, 150160/2023-2), PqG/ FAPERGS (21/ 2551-0002057-1), FAPERGS (24/2551-0001396-2), FAPERGS/CNPq (23/2551-0000126-8), and PRONEX (16/ 2551-0000488-9).

Integral-Derivative (PID) controller, replicating the traditional controller response with a shorter actuation time. The ANFIS implementation used is publicly available [15] and will be considered in this study.

Continuing previous studies addressing aggregators and fuzzy operators, such as t-norms and t-conorms, this work aims to explore the application of these aggregators directly in the ANFIS architecture to observe changes in the results predicted by the network when incorporating this new approach.

The paper is organized as follows. Section II recalls some important background concepts. In Sect. III details of the methodology and the computational environment are discussed. Then, Section IV provides the results and a brief analysis of the experiments. And at last, Section V presents the conclusions and future works.

II. Preliminary Concepts

The proposed generalized architecture rests upon the synergy between FSs and adaptive neural architectures. To provide the foundation for the modifications introduced in this work, this section recalls these fundamental definitions.

A. Fuzzy Logic

Fuzzy Logic [10] generalizes classical logic by allowing elements to belong to sets with different degrees of membership, modeling uncertainties and linguistic terms [16]. A Fuzzy Set (FS) A in a universe $X \neq \emptyset$ is characterized by its Membership Function (MF) defined as follows.

Definition 2.1: The MF denoted by $\mu_A(x)\colon X \to [0,1]$ defines the membership level of x to subset A, where the FS A is given by:

$$A = \{(x, \mu_A(x))\colon x \in X\}. \tag{1}$$

Definition 2.2: A function $M\colon [0,1]^n \to [0,1]$ is an n-ary aggregation function (aggregator) if it verifies the following axioms [17]:

(A1) If $x_i \leq y_i$ for each $i = 1, \ldots, n$, then $M(x_1, \ldots, x_n) \leq M(y_1, \ldots, y_n)$ (isotonicity);

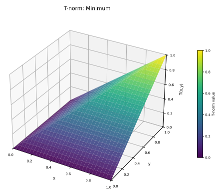
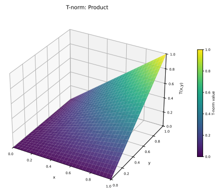
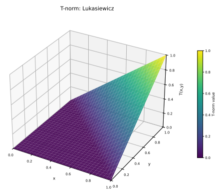
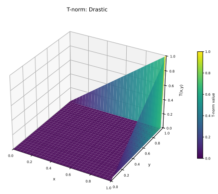
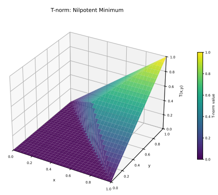
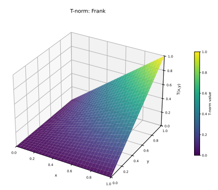
(A2) $M(0, \ldots, 0) = 0$ and $M(1, \ldots, 1) = 1$ (boundary conditions).

Among the various fuzzy connectives, the binary aggregator called fuzzy t-norm is defined as follows:

Definition 2.3: A triangular norm, t-norm for short, is a binary operation $T\colon [0,1]^2 \to [0,1]$, such that $\forall x, y, z \in [0,1]$, satisfies [18], [19]: commutativity, associativity, monotonicity, and the boundary condition: $T(x,1) = x$.

To assess the impact of different logical behaviors on the performance of the generalized architecture, we consider a diverse set of t-norms. These include both fundamental non-parametric t-norms and parametric families, as detailed in Tables I and II.

TABLE I
Definitions of the parametric and non-parametric t-norms used in the comparative study (Part 1).

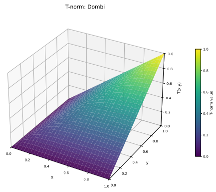
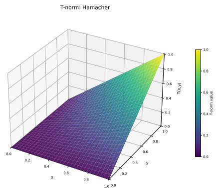
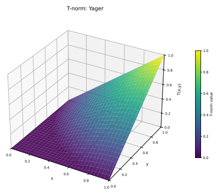
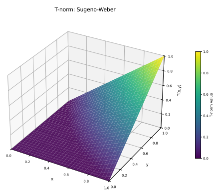
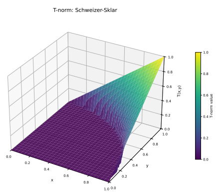
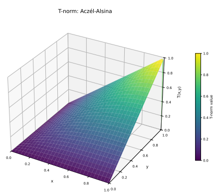
Definition	**3D graphic**
$T_M(x,y) = \min(x,y)$	
$T_P(x,y) = x \cdot y$	
$T_L(x,y) = \max(0, x+y-1)$	
$T_D(x,y) = \begin{cases} x, & \text{if } y = 1, \\ y, & \text{if } x = 1, \\ 0, & \text{otherwise.} \end{cases}$	
$T_{NM}(x,y) = \begin{cases} \min(x,y), \\ \quad \text{if } x+y > 1, \\ 0, \\ \quad \text{otherwise.} \end{cases}$	
$T^F_\lambda(x,y) = \begin{cases} T_M(x,y), \text{if } \lambda = 0, \\ T_P(x,y), \text{if } \lambda = 1, \\ \log_\lambda(1 + \frac{(\lambda^x - 1)(\lambda^y - 1)}{\lambda - 1}), \\ \quad \text{otherwise.} \end{cases}$	

B. Inference Systems & Fuzzy Control

A Fuzzy Inference System (FIS) models decision processes through "If-Then" linguistic rules, allowing the construction of interpretable controllers that replace complex mathematical models [20]. The main components of a FIS, include the fuzzification step (mapping inputs to membership degrees), the database and rule base, the inference unit, and the defuzzification unit (converting FSs to crisp values) [21], [22].

The rules have the deductive format "If *antecedent* then *consequent*", where the antecedent defines the activation con-

TABLE II
DEFINITIONS OF THE PARAMETRIC AND NON-PARAMETRIC T-NORMS USED IN THE COMPARATIVE STUDY (PART 2).

Definition	3D graphic
$T_\lambda^D(x,y)=\begin{cases} T_D(x,y), \text{ if } \lambda=0, \\ T_M(x,y), \text{ if } \lambda=\infty, \\ \frac{1}{1+((\frac{1-x}{x})^\lambda+(\frac{1-y}{y})^\lambda)^{1/\lambda}}, \\ \text{otherwise.} \end{cases}$	
$T_\lambda^H(x,y)=\begin{cases} T_D(x,y), \text{ if } \lambda=\infty, \\ 0, \text{ if } \lambda=x=y=0, \\ \frac{xy}{\lambda+(1-\lambda)(x+y-xy)}, \\ \text{otherwise.} \end{cases}$	
$T_\lambda^Y(x,y)=\begin{cases} T_D(x,y), \text{ if } \lambda=0, \\ T_M(x,y), \text{ if } \lambda=\infty, \\ \max(1-((1-x)^\lambda+(1-y)^\lambda)^{1/\lambda},0) \\ \text{otherwise.} \end{cases}$	
$T_\lambda^{SW}(x,y)=\begin{cases} T_D(x,y), \text{ if } \lambda=-1, \\ T_P(x,y), \text{ if } \lambda=\infty, \\ \max((\frac{x+y-1+\lambda xy}{1+\lambda}),0), \\ \text{otherwise.} \end{cases}$	
$T_\lambda^{SS}(x,y)=\begin{cases} T_M(x,y), \text{ if } \lambda=-\infty, \\ T_P(x,y), \text{ if } \lambda=0, \\ T_D(x,y), \text{ if } \lambda=\infty, \\ \max(x^\lambda+y^\lambda-1,0)^{\frac{1}{\lambda}}, \\ \text{otherwise.} \end{cases}$	
$T_\lambda^{AA}(x,y)=\begin{cases} T_D(x,y), \text{ if } \lambda=0, \\ T_M(x,y), \text{ if } \lambda=\infty, \\ e^{-\left((-\log x)^\lambda+(-\log y)^\lambda\right)^{\frac{1}{\lambda}}}, \\ \text{otherwise.} \end{cases}$	

ditions and the consequent represents the inferred output. Mathematically, the controller's behavior is defined by a fuzzy relationship $\mathcal{R}$ between the input spaces X and the output Y, as defined in Eq. (2) [23]:

$$\mu_{\mathcal{R}(A,B)}(x,y)=\mu_A(x)\rightarrow\mu_B(y). \tag{2}$$

1) Takagi-Sugeno Controller: The Takagi-Sugeno method is a high-precision controller that models the consequent of a rule as a linear function of the input terms, avoiding computationally exhaustive defuzzification. In a first-order model, the output of each rule R_i is given by $f_i(x)=\sum a_{ij}x_j+b_i$, and the overall output $F_{TS}(x)$ is obtained by the weighted average [13]: $F_{TS}(x)=\frac{\sum_{i=1}^{n}\mu_{A_i}(x)\cdot f_i(x)}{\sum_{i=1}^{n}\mu_{A_i}(x)}$, where the weights $\mu_{A_i}(x)$ correspond to the rule activation degree. This structure resembles a local linear regression for each activated rule [24]. Although robust, the manual definition of its parameters is complex in large volumes of data, a gap that the ANFIS architecture addresses by integrating neural network mechanisms for automatic adjustment.

C. Architecture & ANFIS Layers

Neural Networks are adaptive structures composed of layers of neurons interconnected by weights, whose learning occurs by adjusting these weights to minimize the discrepancy between the network output and the target value [25], [26]. In the context of ANFIS, this directed graph architecture is used to functionally represent the fuzzy inference process, allowing premise parameters (membership functions) and consequent parameters to be automatically optimized via supervised learning algorithms [27]. In this way, the NN structure provides the necessary computational basis for the model's adaptive capacity, combining the convergence efficiency of neural networks with the interpretability of fuzzy logic.

ANFIS is a neuro-fuzzy system that implements a Takagi-Sugeno inference model within an adaptive neural network framework, making it effective for modeling nonlinear systems and time series [12], [28]. ANFIS learning uses a hybrid algorithm that combines the Gradient Descent Method (GDM) to adjust the parameters of the membership functions (premise) and the Least Squares Estimator (LSE) for the parameters of the linear functions (consequent) [29].

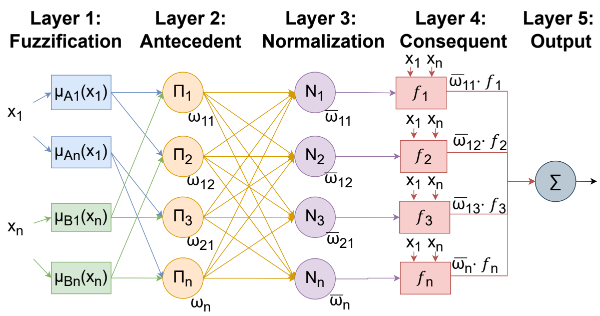


Fig. 1. A Sugeno structure of the ANFIS model. Adapted from [12].

The architecture, seen in Fig. 1, is organized into five layers, where the output of the i-th node in layer k is denoted by O_i^k:

- **C1 (Fuzzification):** Extracts the membership degrees of the entries x in the fuzzy sets A_i: $O_i^1=\mu_{A_i}(x)$.
- **C2 (Antecedent):** Calculates the firing strength (ω_i) of each rule via t-norms (e.g., the product t-norm): $\omega_i=\mu_{A_i}(x)\cdot\mu_{B_i}(y)$.
- **C3 (Normalization):** Normalizes the firing strength: $\bar{\omega}_i=\omega_i/\sum\omega_j$.

- **C4 (Consequent):** Applies the linear consequent functions: $O_i^4 = \bar{\omega}_i f_i = \bar{\omega}_i(p_i x + q_i y + r_i)$.
- **C5 (Output):** Computes the global output like the sum of the contributions: $f_{out} = \sum \bar{\omega}_i f_i$.

The training aims to minimize the total squared error $E = \sum(T_i - f_{out_i})^2$, where T_i is the target value, through forward propagation and backpropagation cycles [12].

III. The new architecture: Generalization and Protocol

This section details the proposed method and the experimental protocol used to evaluate its performance. We begin by describing the generalization of the second layer, where the static product operator is replaced by a flexible selection module. Subsequently, we outline the experimental configuration, including the optimization of parametric t-norms and the cross-validation protocol. By systematically varying the t-norm classes across diverse datasets, we aim to identify how different logical aggregators influence the network's ability to model non-linear relationships and handle data uncertainty.

A. The proposed Generalization

The new method extends the original architecture of [12] by making Layer 2 (antecedent generation) more flexible. While standard ANFIS invariably uses the Product t-norm (T_P), the proposed architecture inserts an Aggregator Selection Module in the second layer of ANFIS. Thus, the calculation of the activation strength ω_i, originally defined by $\prod_{j=1}^{n} \mu_{A_{ij}}(x_j)$, is replaced by a generic operator $\mathfrak{B}$: $\omega_i = \mathfrak{B}(\mu_{A_{i1}}(x_1), \ldots, \mu_{A_{in}}(x_n))$.

Note that $\mathfrak{B}$ encompasses non-parametric t-norms (T_M, $T_Ł$, T_D, T_{NM}) and parameterized t-norms (T_λ^F, T_λ^{SS}, T_λ^D, T_λ^H, T_λ^{SW}, T_λ^Y). To mitigate combinatorial explosion, $n(MF) = 2$ was fixed by attribute, resulting in $n(R) = 2^{n(A)}$ rules.

B. Configuration and Protocol

The experimental process was conducted in three integrated stages. Initially, 15 datasets were selected from the KEEL repository [30] and preprocessed using categorical label coding and normalization. Training utilized 5-fold cross-validation [1] to ensure statistical robustness. The final performance (accuracy) is the arithmetic mean of the iterations, where in each round, four subsets trained the model and one validated the results. For the parametric t-norms, the λ parameter was optimized via exhaustive search over the domain of each family, selecting the parameter with the highest average accuracy. As a stopping criterion, two epochs was determined, as preliminary analyses indicated premature convergence without significant gains beyond the second epoch. The systematic execution compared the baseline (classic ANFIS with T_P) against the generalized architecture variants using 11 selected t-norms.

C. Considered Dataset

The experiments were conducted on 15 datasets from the Knowledge Extraction based on Evolutionary Learning (KEEL) dataset repository [30]. Their main features are presented in Table III, with their **Id**, **name**, and the number of Instances (**Inst.**), Attributes (**Aatt.**), and Classes (**Cla.**). Due to ANFIS's requirement for exclusively numerical entries, categorical variables were mapped beforehand.

TABLE III
FEATURES OF THE DATASETS USED IN THE EXPERIMENTS.

Id	Dataset name	Inst.	Att.	Cla.	Id	Dataset name	Inst.	Att.	Cla.
1	Appendicitis	106	7	2	9	Iris	150	4	3
2	Balance	625	4	3	10	Led7digit	500	7	10
3	Banana	5300	2	2	11	Newthyroid	215	5	3
4	Bupa	345	6	2	12	Phoneme	5404	5	5
5	Ecoli	336	7	8	13	Pima	768	8	2
6	Glass	214	9	6	14	Titanic	2201	3	2
7	Haberman	306	3	2	15	Yeast	1484	8	10
8	Hayes-roth	160	4	3					

D. Computational Environment

The experiments performed on the datasets, seen in Table III, considered a machine equipped with an AMD Ryzen Threadripper PRO 5955WX processor (16 physical cores and 32 threads), 64 GB of RAM, x86-64 architecture supporting 32-bit and 64-bit instructions, and Ubuntu 20.04 LTS OS.

The implementation was made in Python and followed the ANFIS structure from the repository used in [14], publicly available at [15].

Once the execution was complete, the data was processed. The results were collected, organized, and analyzed to compare the aggregator classes of the new approach with the baseline performance (original ANFIS), culminating in the generation of tables for the consolidation and discussion of the experimental findings, which will be properly discussed in the following section.

IV. Results and Assessments

This section presents the empirical results and a comparative performance analysis of the proposed generalized architecture against the standard ANFIS baseline. The evaluation focuses on classification accuracy across 15 diverse datasets, assessing how integrating 12 t-norm classes affects the model's predictive power.

Table IV summarizes these results, organized by dataset (rows in alphabetical order) and fuzzy aggregators (columns), including the baseline product t-norm. The table includes the overall average accuracy per aggregator (M_{Agg}) and the average performance per dataset for the new model (M_DS). Values in bold indicate the best performance for each dataset.

The comparative analysis (Table IV) shows that the generalized architecture surpassed the baseline in 11 of 15 datasets (73.3%). In 86.6% of cases, the results were equal to or better than the original ANFIS. Notably, in the *Led7digit* dataset (Id 10), the new approaches achieved an average accuracy of 0.1670, a significant improvement over the 0.1360 achieved by the standard model.

The results indicate that the proposed variants exhibit superior consistency, with T_λ^{SW} and T_λ^F emerging as the most effective aggregators. Performance gains are particularly evident in datasets such as *Led7digit*, while in higher-dimensional datasets (e.g., *Ecoli* and *Glass*), flexible operators capture more

TABLE IV
RESULTS OF THE CLASSES OF T-NORMS APPLIED TO ALL 15 DATASETS.

Id	**ANFIS** (T_P)	T_λ^D	T_λ^F	T_λ^H	T_λ^{SS}	T_λ^{SW}	T_λ^Y	T_λ^{AA}	T_M	$T_Ł$	T_D	T_{NM}	M_DS
1	0.8775	0.8684	0.8775	0.8775	**0.9152**	0.8957	0.8957	0.8775	0.8775	0.8957	0.8684	0.8775	0.8842
2	0.6864	0.6864	0.6864	0.6864	0.6864	**0.6928**	0.6864	0.6864	0.6864	0.6864	0.6864	0.6864	0.6870
3	**0.5674**	0.5672	**0.5674**	**0.5674**	**0.5674**	0.5617	0.5666	0.5666	0.5666	0.5613	0.5632	0.5613	0.5652
4	0.7101	0.7101	0.7101	0.7101	**0.7420**	0.7101	0.7101	0.7101	0.7101	0.7101	0.7101	0.7101	0.7130
5	0.2620	0.2709	0.2709	0.2709	0.2710	**0.3215**	0.2710	0.2709	0.2709	0.2710	0.2650	0.2650	0.2745
6	0.4577	0.4670	0.4577	0.4577	0.4577	**0.4720**	0.4670	0.4577	0.4577	0.4672	0.4670	0.4577	0.4624
7	0.7451	0.7451	0.7451	0.7451	**0.7614**	0.7451	0.7451	0.7451	0.7451	0.7451	0.7451	0.7451	0.7466
8	0.7496	0.7496	0.7496	0.7496	0.7496	0.7496	0.7496	**0.7696**	0.7496	0.7496	0.7496	0.7496	0.7514
9	0.9733	0.9733	0.9733	0.9733	0.9733	0.9733	0.9733	0.9733	0.9733	0.9733	0.9733	0.9733	0.9733
10	0.1360	0.2260	**0.2300**	0.2260	0.1360	0.1360	0.1360	0.1960	**0.2300**	0.1360	0.1100	0.1040	0.1696
11	0.8140	0.8140	0.8140	0.8140	0.8140	0.8140	0.8140	0.8140	0.8140	0.8140	0.8140	0.8140	0.8140
12	0.7515	**0.7519**	0.7517	0.7518	0.7515	**0.7519**	**0.7519**	0.7517	0.7517	**0.7519**	**0.7519**	**0.7519**	0.7518
13	0.7799	0.7786	0.7812	0.7799	0.7799	**0.7877**	0.7786	0.7812	0.7812	**0.7877**	0.7786	0.7812	0.7814
14	**0.7760**	**0.7760**	**0.7760**	**0.7760**	**0.7760**	**0.7760**	**0.7760**	**0.7760**	**0.7760**	**0.7760**	0.6820	**0.7760**	0.7675
15	0.3235	0.3228	0.3221	0.3241	0.3235	**0.3302**	0.3228	0.3221	0.3221	**0.3302**	0.3228	0.3235	0.3242
M_{Agg}	0.6407	0.6472	0.6475	0.6473	0.6470	**0.6478**	0.6429	0.6465	0.6475	0.6437	0.6325	0.6384	

complex relationships. This suggests that parametric t-norms improve adaptability to nonlinear dependencies.

A. Execution time

TABLE V
RUNTIME ANALYSIS BY DATASET.

Id	Rules	Inst.	Avg(h)	Max(h)	Id	Rules	Inst.	Avg(h)	Max(h)
1	2^7	106	3.16	4.47	9	2^4	150	1.46	1.46
2	2^4	625	0.59	0.59	10	2^7	500	4.89	4.89
3	2^2	5300	1.12	1.12	11	2^5	215	1.80	1.80
4	2^6	345	1.92	1.92	12	2^5	5404	14.58	16.89
5	2^7	336	3.45	3.45	13	2^8	768	32.83	46.51
6	2^9	214	41.35	45.81	14	2^3	2201	6.35	6.35
7	2^3	306	1.19	1.19	15	2^8	1484	73.37	74.45
8	2^4	160	1.29	1.29					

Table V presents the runtime performance across the 15 evaluated datasets. The results indicate that the computational cost of the proposed ANFIS generalization is primarily driven by the dimensionality of the input space rather than the number of instances. Since the number of fuzzy rules in the ANFIS architecture grows exponentially with the number of attributes (2^n for two membership functions per input), datasets with high dimensionality, such as *Yeast* ($n = 8$) and *Glass* ($n = 9$), exhibited the highest execution times, reaching up to 73.37h and 41.35h, respectively. The impact of the rule base size becomes evident when comparing the *Banana* (Id 3) and *Pima* (Id 13) datasets. Although *Banana* contains significantly more instances (5300 vs. 768), its low dimensionality ($n = 2$) results in only 4 rules, allowing execution to complete in 1.12h. In contrast, *Pima* ($n = 8$, 256 rules) requires over 32h. This confirms that the antecedent layer complexity is the main bottleneck for the A2-ANFIS model. Furthermore, the variability in execution time across different t-norms was found to be marginal, ranging between 1% and 16%. This suggests that while the choice of fuzzy aggregator significantly impacts predictive accuracy, the computational overhead of switching between different t-norms (e.g., Minimum T_M vs. Dombi T_λ^D) is negligible within the overall training process.

B. Statistical Analysis of Results

To validate the proposed extensions, a statistical analysis was performed following Demšar's [31] recommendations. The **Friedman test** yielded a p-value of 0.041, rejecting the null hypothesis and confirming that varying the t-norms statistically alters the system's performance.

The **Nemenyi post-hoc test** identified the best average ranks for T_λ^{SW} (4.7), $T_Ł$ (5.4), and T_λ^{SS} (5.5), all outperforming the **standard ANFIS** (7.4). Despite the conservative nature of the Nemenyi test for $k = 12$ algorithms, a clear trend of performance gain was observed with the proposed operators.

For a sensitive pairwise comparison (control vs. treatment), the **Wilcoxon signed-rank test** confirmed that $T_\lambda^{SW}(p = 0.0251)$ and $T_Ł$ ($p = 0.0425$) significantly outperform the original ANFIS ($p < 0.05$). These results indicate that replacing the algebraic product with flexible t-norms yields a genuine optimization of inference capacity, with T_λ^{SW} standing out as the most robust choice.

V. CONCLUSIONS

This work introduces a novel architecture, i.e., an approach that generalizes the traditional neuro-fuzzy ANFIS model by allowing the integration of different t-norms in the antecedent layer. Replacing the fixed product t-norm with parametric and nonparametric operators, the proposal delivered the network greater adaptive flexibility, essential for handling uncertainty in complex datasets.

The analysis demonstrates that t-norm performance correlates with their 3D geometric properties. The parameter λ provides adaptive flexibility, allowing parametric operators to emulate fundamental behaviors (e.g., T_M, T_P, or T_D) and tune antecedent curvature to data-specific needs. While simpler datasets (e.g., *Iris* and *Newthyroid*) show convergence across different aggregators, the Schweizer-Sklar (T_λ^{SS}) and Sugeno-Weber (T_λ^{SW}) t-norms excel in complex scenarios by modeling non-linear dependencies. Ultimately, this flexibility allows A2-ANFIS to overcome the rigidity of the standard product operator, especially in high-dimensional problems.

Experimental results demonstrated that the baseline ANFIS model in 73.3% of the tested cases and remained the same in 13.3%. The Schweizer-Sklar (T_λ^{SS}) and Sugeno-Weber (T_λ^{SW}) operators stood out, achieving the best results across different datasets, demonstrating that adjusting the λ parameter allows for the capture of fundamental nonlinear patterns. Furthermore, in critical cases such as the *Led7digit* dataset, the appropriate choice of aggregator resulted in significant performance gains.

Observing Table IV, the M_DS column validates the work's proposal, as it demonstrates that, in most cases, using alternative aggregators results in higher accuracy than the fixed use of the product t-norm. Although in two cases (datasets with IDs 3 and 14) performance was slightly lower, the overall balance is largely positive, raising the overall average accuracy of the system.

Future research will move in two distinct directions. First, the framework will be expanded to include new datasets. Additionally, we intend to develop a hybrid learning algorithm that enables the automatic tuning of the t-norm parameters (λ) during the training cycle itself, thereby reducing the computational overhead associated with exhaustive searches.

REFERENCES

[1] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

[2] P. Mitra and H. H. Sidhwa, "Collaborative filtering for personalized movie recommendations on streaming platforms," in *2025 Intl. Conf on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI)*, 2025, pp. 1–6.

[3] Q. Khaled, U. Kaymak, and L. Genga, "Interpretable fuzzy systems for forward osmosis desalination," in *2025 IEEE Intl. Conf on Fuzzy Systems (FUZZ)*, 2025, pp. 1–7.

[4] T. M. H Gedara, V. Loia, and S. Tomasiello, "Detecting fake images using neuro-fuzzy inference systems: A brief comparative analysis," in *2024 IEEE Int.l Conf. on Fuzzy Systems (FUZZ-IEEE)*, 2024, pp. 1–7.

[5] A. Karki, C. Hardikar, M. A. Kumar, and S. Das, "Improved ANFIS Model Based on Confinement Optimized PSO to Predict the Air Quality of Delhi, India," in *2023 IEEE Intl. Conf on Fuzzy Systems (FUZZ)*, 2023, pp. 1–6.

[6] S. Russell, P. Norvig, and A. Intelligence, "A modern approach," *Artificial Intelligence*, vol. 25, no. 27, pp. 79–80, 1995.

[7] A. A. M. Lêu, S. F. Amorim, and A. F. L. Amorim, "Aplicação de redes neurais artificiais para predição de deslizamentos de terra induzidos por chuvas," *Revista Políticas Públicas & Cidades*, vol. 13, no. 2, pp. e1344–e1344, 2024.

[8] L. C. A. Da Silva and J. R. Monteiro, "Sistema para apoio ao diagnóstico de câncer de mama baseado em redes neurais artificiais," in *Engenharias-automação, robótica, metrologia e energia: estudos e tendências - volume 1*, vol. 1. Editora Científica Digital, 2023, pp. 49–65.

[9] G. V. Moura *et al.*, "Previsão do preço de ações da petrobras utilizando redes neurais artificiais," Master's thesis, Universidade Tecnológica Federal do Paraná, 2025.

[10] L.A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[11] H. H. Tang and N. S. Ahmad, "Fuzzy logic approach for controlling uncertain and nonlinear systems: a comprehensive review of applications and advances," *Systems Science & Control Engineering*, vol. 12, no. 1, p. 2394429, 2024.

[12] J.-S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[13] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[14] S. G. L. B. Serri, "Implementação de Controlador Neuro-Fuzzy e Sistema Supervisório para Planta Didática," *Final Term Paper (B.S. in Electrical Engineering)*, p. 55, 2022.

[15] T. Meggs, "Anfis," https://github.com/twmeggs/anfis, 2020, last accessed on May 16, 2025.

[16] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning - I," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[17] G. Beliakov, A. Pradera, T. Calvo *et al.*, *Aggregation functions: A guide for practitioners*. Springer, 2007, vol. 221.

[18] G. J. Klir and B. Yuan, *Fuzzy sets and fuzzy logic - theory and applications*. Prentice Hall, 1995.

[19] E. P. Klement, R. Mesiar, and E. Pap, *Triangular Norms*. Dordrecht, Netherland: Springer, 2000.

[20] N. Sabri, S. Aljunid, M. Salim, R. Badlishah, R. Kamaruddin, and M. Malek, "Fuzzy inference system: Short review and design," *Int. Rev. Autom. Control*, vol. 6, no. 4, pp. 441–449, 2013.

[21] R. R. Yager and L. A. Zadeh, *An introduction to fuzzy logic applications in intelligent systems*. Springer Science & Business Media, 2012, vol. 165.

[22] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on systems, Man, and Cybernetics*, no. 1, pp. 28–44, 1973.

[23] T. J. Ross, *Classical Relations and Fuzzy Relations*. USA: John Wiley and Sons, Ltd, 2010.

[24] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, "Statistical learning," in *An introduction to statistical learning: With applications in Python*. Springer, 2023, pp. 15–67.

[25] S.-C. Wang, *Artificial Neural Network*. Boston, MA: Springer US, 2003, pp. 81–100.

[26] A. Dongare, R. Kharde, A. D. Kachare *et al.*, "Introduction to artificial neural network," *Intl. Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 1, pp. 189–194, 2012.

[27] K. Mehrotra, C. K. Mohan, and S. Ranka, *Elements of artificial neural networks*. MIT press, 1997.

[28] P. Melin, J. Soto, O. Castillo, and J. Soria, "A new approach for time series prediction using ensembles of ANFIS models," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3494–3506, 2012.

[29] D. Karaboga and E. Kaya, "Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey," *Artificial Intelligence Review*, vol. 52, pp. 2263–2293, 2019.

[30] J. Derrac, S. Garcia, L. Sanchez, and F. Herrera, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Mult. Valued Logic Soft Comput*, vol. 17, pp. 255–287, 2015.

[31] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.