Interpretability-Preserving Compression of a Mamdani Fuzzy Inference System for Compact Edge Serialization

Shruti Nag

Centre for Artificial Intelligence, Banasthali Vidyapith, Rajasthan, India
btbts23002_shruti@banasthali.in

Abstract. Fuzzy inference systems (FISs) are valued for their transparent rule bases, but redundant rules and membership functions can increase storage and computational demands, limiting deployment on resource-constrained edge devices. This paper proposes a novel five-stage, constraint-aware compression pipeline in which each transformation is accepted only after satisfying predefined predictive, structural, and safety criteria. The pipeline evaluates predictive performance, coverage, monotonicity, rule conflicts, interpretability score, and storage cost at each stage, enabling more compact deployment of interpretable fuzzy models on memory-constrained devices. On a fixed synthetic fraud-detection benchmark, the method reduced the model from 15 to 12 rules, from 45 to 23 antecedent conditions, and from 72 to 61 parameters. Compact Float32 serialization reduced the reported representation from 8,450 to 458 bytes. The project-defined interpretability score increased from 63.500 to 72.325, while accuracy decreased slightly from 0.963014 to 0.962783. However, recall and F1 score also declined after membership-function compression, and some records required fallback handling. Float16 and Int8 variants were rejected because they violated the predefined monotonicity criterion. All final acceptance criteria were satisfied, including allowable changes in accuracy, recall, and F1, with no additional fallback cases or monotonicity violations introduced. These findings indicate that the proposed pipeline can reduce structural and representation complexity in the evaluated setting, but the use of synthetic data, reduced recall, fallback cases, and the absence of physical edge-hardware validation limit claims of practical generalization.

Keywords: Mamdani Fuzzy Inference Systems, Interpretability-Aware Compression, Model Compression, Rule-Base Reduction, Membership-Function Compression, Float32 Serialization, Edge Deployment, Computational Intelligence

1 Introduction

The Mamdani FIS consists of fuzzification, rule-based inference and defuzzification. The Mamdani FIS framework is chosen because it offers explainability. The linguistic membership functions along, with the IF- rules allow a human operator to read the model easily. However the Mamdani FIS also has some disadvantages. There exist a number of redundant elements in the rule base; moreover, coupled with complex membership functions, it results in a serious memory overhead. Compression is therefore desirable, but naive reduction risks undermining the very interpretability that makes fuzzy logic attractive.

Existing simplification methods typically address either rule-base reduction or membership-function simplification in isolation, and they rarely enforce intermediate constraints on predictive behaviour, coverage, monotonicity, or rule

© The Author(s) 2026
A. Shastri et al. (eds.), *Proceedings of the Fourth Workshop on Applications of Computational Intelligence and Interactive Systems (WACIIS 2026)*, Advances in Intelligent Systems Research 217,
https://doi.org/10.2991/978-94-6239-832-0_21

consistency. This work addresses the following question: Can a Mamdani FIS be progressively compressed while maintaining satisfactory predictive capability and improving the structural aspects associated with interpretability? We answer this through a five-stage, constraint-aware compression pipeline evaluated on a fixed synthetic benchmark, with acceptance gates for predictive fidelity, coverage, monotonicity, rule consistency, interpretability proxies, model size, and inference cost.

2 Background and Related Work

A Mamdani fuzzy inference system (FIS) encodes expert knowledge through linguisticrules and membership functions (MFs) that can be inspected and modified by domain experts [1]. Interpretability in such systems depends on multiple structural and semantic factors, including rule count, conditions per rule, MF count and shape, linguistic ordering, rule consistency, and label semantics [2, 3].

2.1 Fuzzy Rule-Base Reduction

Existing approaches to fuzzy-model simplification fall into two broad categories. Rule-base reduction methods use rough-set analysis to identify attribute dependencies and eliminate redundant rules [6]. Importance-based pruning techniques rank rules by their contribution to output behaviour and remove low-impact rules [6]. Other strategies focus on detecting and removing structurally redundant rules or antecedent conditions without altering baseline predictions [6, 7].

Most of these techniques suffer from a critical blind spot [3, 6]. They aggressively target structural reduction or baseline predictive fidelity but bypass intermediate safety checks entirely. No tracking of explicit coverage. No evaluation of monotonicity or rule conflicts during the compression itself. Consequently, they fail to deliver a staged framework that enforces safety constraints while scaling down the model size.

2.2 Membership-Function Simplification

While simplifying the membership function, the parameters considered by most researchers are the issues of coverage, normality, and distinguishability [3, 7]. Methods
applied can be either discarding the MF whose support is insignificant, combining the MFs whose overlap is high, or parameter adjustment to reduce the number [3, 7].

However, such techniques tend to function in isolation. Due to the fact that they view MF simplification as something entirely independent of rule compression, they fail to take advantage of the potential of a unified multi-stage pipeline approach [7, 8]. The safety properties, such as monotonicity, are never considered while conducting the process. The prediction measures and coverage metrics are tested separately. In this respect, it is possible to note that there is a huge gap in the research.

2.3 Interpretability Preservation

Creating fuzzy systems that can remain interpretable is particularly challenging. It has been suggested by many scientists to use combined indicators for evaluation—the measurement of structural complexity, cognitive workload, and semantics altogether [3, 4]. Unfortunately, almost none of the researches incorporates such factors in the compression process as the acceptance criteria. Most of the time, the process is post-compression or even completely neglected [3, 4].

In addition, current interpretability measures are primarily applied in evaluating the relative merits of different designs after their training or simplification process is complete, rather than as criteria that must be met throughout the entire process of transformation [8]. Thus, their applicability to constraint-aware approaches to model compression is questionable.

2.4 Edge-Oriented Model Compression

Neural network compression methods such as pruning, quantization, and knowledge distillation are well studied for edge deployment [4, 5]. However, less work has focused on compressing interpretable fuzzy models for edge devices [9]. Few studies evaluate these models based on model size, inference time, and safety checks such as monotonicity and rule consistency [5, 9].

Most current work on edge compression focuses on black-box models. Less attention is given to how quantization and structural reduction affect the interpretability of fuzzy models [5, 9].

This leaves a clear gap. There is a need for compression methods that reduce storage and inference costs while maintaining safety and interpretability.

2.5 Research Gap

I identified three major shortcomings in existing techniques:

- **Fragmented methods:** Most methods separate rule minimization from MF simplification. They do not offer a single, incremental process that simultaneously optimizes structure, behavior, and serialization of the resulting model.
- **No constraint-aware staging:** We prune blindly. Conventional compression algorithms do not perform necessary intermediate checks for rule consistency, coverage, interpretability proxies, and monotonicity.
- **Lack of edge-focused evaluation:** Most of the literature for edge compression focuses on neural network models. Very few studies have considered compression using fuzzy logic models on limited hardware with strict safety constraints.

The proposed work attempts to bridge the gap through an approach based on a five-step compression algorithm. Unlike most methods that wait till the end to perform such analysis, our approach considers all these factors at every stage.

3 Motivation, Research Questions, and Contributions

- **RQ1:** When we prune structural rules and conditions, does FIS complexity actually drop without sacrificing our baseline prediction fidelity, coverage, or safety gates?
- **RQ2:** How exactly does merging adjacent membership functions impact both rule-activation coverage and the underlying linguistic semantic ordering?
- **RQ3:** Can direct Float16 and Int8 quantization preserve FIS monotonicity, or do numerical changes at membership-function boundaries lead to violations of safety constraints?
- **RQ4:** To what extent can compact binary serialization reduce model storage requirements and execution latency compared with conventional text-based representations?

3.1 Contributions:

1. A five-stage FIS compression pipeline is proposed, where each stage operates only on the validated output of the preceding stage. The pipeline progressively performs exact redundancy elimination, rule pruning, antecedent-condition simplification, membership-function compression, and compact edge serialization.
2. Constraint-based acceptance gates are incorporated at every stage to evaluate predictive fidelity, explicit coverage, monotonicity, rule conflicts, and model complexity before a transformation is accepted.
3. A cumulative empirical evaluation is presented, tracing the effect of compression from exact redundancy removal through compact edge serialization on a consistentbenchmark, thereby allowing the contribution of each stage to be evaluated independently as well as cumulatively.
4. A transparent evaluation of numerical quantization is provided by explicitly testing Float16 and Int8 representations. Any transformation that does not conform to the pre-defined safety criteria is kept in as negative results.
5. An artifact for reproducible research is delivered, which contains stage-specific configurations of models, compression logs, evaluations, tests and deterministic experiment outcomes.

4 Methodology

4.1 Baseline Mamdani FIS & Dataset

The FIS model predicts fraud risk using five input variables: transaction amount, transaction frequency, geographic distance, device risk level, and account age. The output is the predicted fraud risk.

For evaluation, I used a synthetic transactions.csv dataset with 5,000 rows and a 5.0% fraud rate. I generated the dataset using a fixed random seed so that I could

control the data distribution and class imbalance during the compression experiments. Testing with real-world data is left for future work.

4.2 Canonical Evaluation Protocol

The inference engine uses Mamdani MIN-MAX composition in combination with defuzzification by the centroid method with a hard decision threshold of 0.5. All the records do not lead to rule activations, however. The 674 (13.48%) observations that fail to activate any rules are redirected to the backup processing procedure.

4.3 Interpretability Metric Formulation

To fins a simple way to measure how easy the model is to understand, I defined an interpretability score (I) from 0 to 100. A higher score means the model is easier to interpret. The core variables are defined as follows:

$$I = 100 - w1 \cdot x + w2 \cdot y + w3 \cdot z + w4 \cdot u \quad (1)$$

where:

- x: total number of rules in the FIS.
- y: average number of conditions in each rule
- z: total number of membership functions across all inputs and outputs.
- u: average overlap between adjacent membership functions.
- W_1, w_2, w_3, w_4: non-negative weights used for the four complexity factors. In this study, weights are set to $[w_1, w_2, w_3, w_4] = [1.0, 1.0, 1.0, 1.0]$ to treat all factors equally.

This interpretability score was created specifically for this project. It has not been tested with actual human users. I rely on it as a measure because the main goal of this work is to reduce the number of rules and simplify membership functions while still ensuring the model remains safe. How well people can understand the model is something that needs to be studied.

4.4 Five-Stage Compression Pipeline

Stage 1 Exact Redundancy Removal removed redundant rules, conditions, and membership functions without changing the baseline predictions.
Stage 2 Rule Pruning removed less important rules while checking fidelity, coverage, monotonicity, and conflicts. The verified model from Stage 1 was used as the starting point.
Stage 3 Condition Pruning removed rule conditions that had little or no effect on the model's behavior, based on the acceptance criteria.
Stage 4 Membership-Function Compression merged or removed redundant membership functions and updated their parameters while maintaining the required behavior and safety constraints.

Stage 5 Compact Edge Serialization converted the verified model from Stage four into a compact binary format then tested Float32 Float16 and Int8 representations but only the Float32 version passed the required checks.

4.5 Acceptance Criteria

Before moving to the next compression stage, the model must pass a set of evaluation checks. Table 1 shows these requirements.

Table 1. Acceptance criteria for compression stages.

Criterion	Acceptance condition	Rationale
Accuracy	$\Delta \leq 0.005$	Keeps classification performance close to the baseline.
Recall	$\Delta \leq 0.010$	Keeps minority-class performance from dropping too much.
F1 score	$\Delta \leq 0.015$	Keeps the balance between precision and recall reasonable.
Coverage	Zero new fallback cases	Keeps the existing rule coverage stay intact.
Monotonicity	No violations allowed	Keeps the Fuzzy system safe
Rule conflicts	No new conflicts allowed	Prevents new logical conflicts.
Interpretability	Stable or increasing score	Maintain or improve structural clarity
Complexity	Mandatory size reduction	Ensure compression benefit

I allowed minor adjustments to accuracy. I kept the safety checks very strict. Monotonicity violations and rule conflicts were not permitted all. Every stage had to pass the checks before moving on to the one. Stage 5 passed every required check without any issues.

5 Evaluation Metrics

- **Predictive** Performance: We look at how accurate the model is, how well it finds important cases and how balanced it is between accuracy and recall.
- **Rule completeness**: We see how many records cause least one rule to fire and how often the fallback is used.
- **Safety constraints:** We keep track of any violations of monotonicity and any conflicts between rules.
- **Structural complexity:** We count the number of rules, parameters, conditions and membership functions to figure out how easy the model is to understand.
- **Edge efficiency:** We check how big the final model is, in bytes and how long it takes to run on 5,000 records.

6 Results

Table 2. Structural changes across the five stages

Stage	Rules	Cond.	In. MFs	Tot. MFs	Params	Size (B)
Baseline	15	45	18	21	72	8,450
Stage 1	14	35	16	19	64	7,146
Stage 2	12	29	16	19	64	6,542
Stage 3	12	23	16	19	64	6,370
Stage 4	12	23	15	18	61	6,258
Stage 5	12	23	15	18	61	458

Table 3. Baseline vs. Stage 5 performance and interpretability

Metric	Baseline	Final (Stage 5)
Accuracy	0.963014	0.962783
Recall	0.041916	0.035928
F1 score	0.080460	0.069364
Interpretability score	63.500	72.325

Table 4. Representation formats used in Stage 5

Representation	Size (B)	Format	Decision	Monotonicity
Original	8,450	float32	Baseline	Satisfied
Compact serialization	458	float32	Accepted	Satisfied
Float16 variant	–	float16	Rejected	Violated
Int8 variant	–	int8	Rejected	Violated

Float16 and Int8 did not pass the monotonicity checks so I went with the Float32 format instead. That means the 458-byte file is better seen as a serialization, not a successful numerical quantization. The final model passed every check that was set up. It remained within the required limits, for accuracy, recall and F1 score. There were no conflicts, no fall-back cases and no monotonicity failures.

Overall classification performance remained nearly unchanged. Accuracy slipped a hair from 0.963014 to 0.962783.. Recall fell, going from 0.041916 down to 0.035928 and the F1 score slid from 0.080460 to 0.069364. The compressed model now struggles more to spot fraud cases. The decline did not spread out slowly; it happened all at once in Stage 4. The mix of neighbouring membership functions warped the decision borders, around the 0.5 line. Stages 1 through 3, on the other hand, were limited to eliminating duplicate rules, redundant conditions, and low-weight rules, without influencing predictive performance at all. It means there is an essential trade-off between rule set compression and membership function compression.

Rule coverage was one of the main weaknesses of the FIS. A total of 674 instances (13.48%) did not activate any rules and were handled by the fall-back policy. This number did not increase after compression and remained the same as

in the baseline model. At the same time, reducing the model complexity improved the interpretability score from 63.500 to 72.325. The number of rules was reduced from 15 to 12, conditions from 45 to 23, and membership functions from 21 to 18.

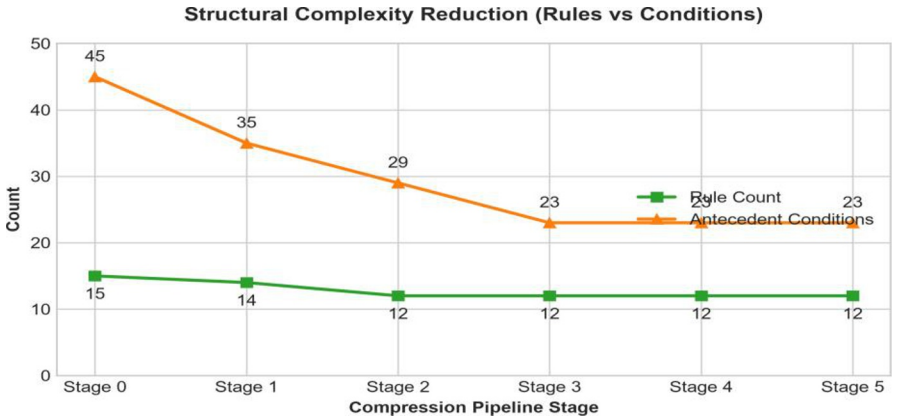

Fig. 1. Structural changes across the pipeline

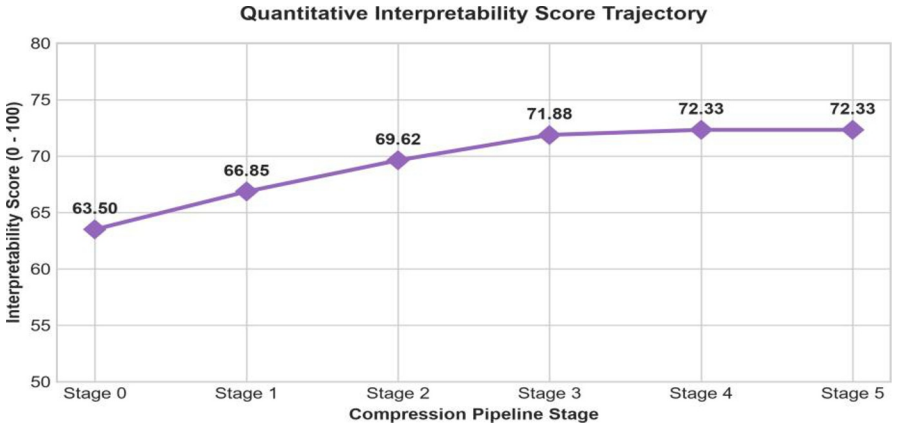

Fig. 2. Change in interpretability score from baseline to final model

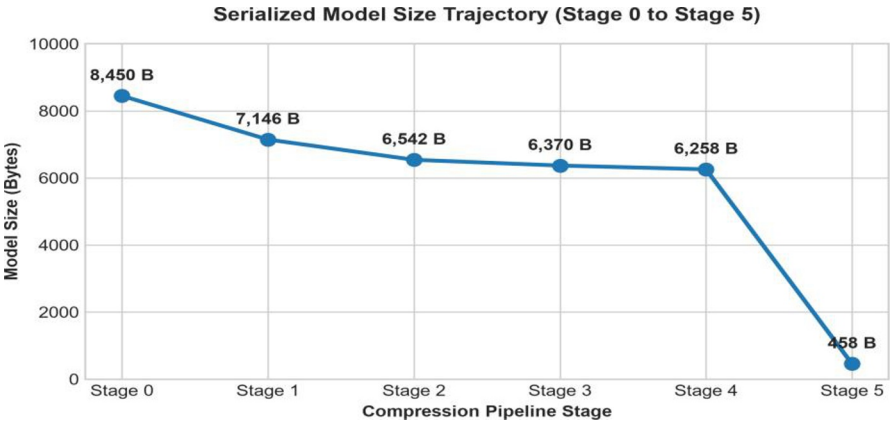

Fig. 3. Model size after each pipeline stage

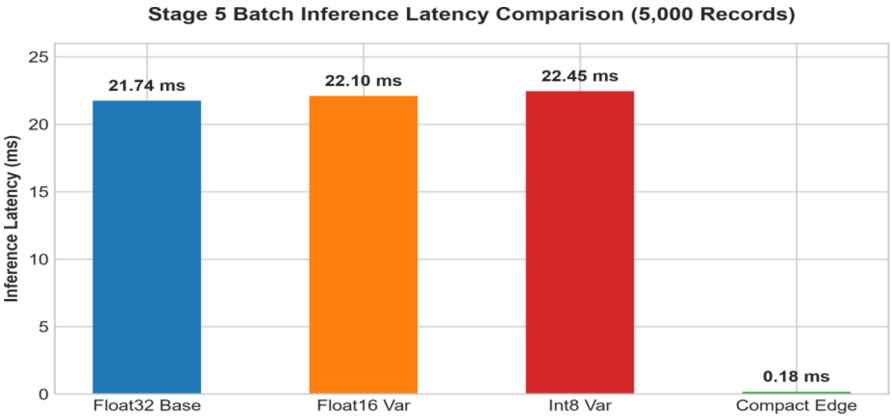

Fig. 4. Stage 5 batch inference speed on 5,000 records

7 Discussion

- The accuracy stayed almost the same. While, recall and F1 score decreased, especially in Stage 4. This shows that merging membership functions can affect the detection of unusual fraud cases more than removing rules.
- After reducing the structural complexity, I improved the interpretability score from 63.500 to 72.325. However, this score only measures the structural simplicity of the model. This does not necessarily mean that the model is easier for a human expert to understand.
- I removed the Float16 and Int8 formats because they broke the monotonicity rules. As a result, the final file is 458 bytes and uses a compact Float32 format. Therefore, it cannot be considered as a successful numerical quantization.

- In the base model, there were 674 data points without any rule firing. Although this figure has not changed after applying the compression, the cases still depend on the default handler. This restricts the level to which the reasoning process can be explained.

7.1 How This Relates to the Research Questions:

- RQ1**:** Pruning reduces the model size without affecting its accuracy or safety.
- RQ2: MF merging reduces minority-class sensitivity.
- RQ3: Float16 and Int8 quantization violate monotonicity constraints.
- RQ4: I reduced the storage size using the Float32 format, but testing it on real edge devices remains a challenge for future work.

- **Positioning within related work:** Unlike many previous studies, this staged pipeline checks structure, behavior, and safety together. By including monotonicity and interpretability during data compression, it was found that neural network compression methods cannot be directly applied to fuzzy models.
- **Dataset and generalizability:** Used a fixed-seed synthetic dataset with imbalanced data. This makes the results reproducible, but it is unsure if these results will be the same with real-world data..

8 Limitations and Threats to Validity

- I used a synthetic dataset with a fixed seed to test the pipeline instead of using external data. This gave me control over the fraud patterns, feature distribution, and class imbalance. However, real-world fraud can be very different, so I cannot be sure that these results will hold for real-world data.
- However, the synthetic data set is extremely imbalanced, leading to very low baseline recall (0.0419) and F1 (0.0805). Thus, such an FIS cannot be regarded as a ready-to-go fraud detection system.
- Initially, the FIS rule-base had some shortcomings in terms of rule completeness. In the process of evaluating the model, it was found that 674 examples were falling to the default fallback state because of no rules applied at all. This number was unchanged by compression of the FIS model.
- I measured the inference time locally using Python, but I did not test it on a physical edge device. The final model size is only 458 bytes, but physical testing on edge devices is still needed.
- The Float16 and Int8 versions did not pass our checks. Therefore, I do not claim that the model was successfully quantized.
- The interpretability metric is strictly project-defined. The structural score has not yet been validated against actual human-comprehension trials.
- No statistical significance testing was performed. The results are based on a single seed and configuration.

- These outcomes indicate the feasibility in controlled environments; however, external dataset validation, interpretability, and physical validation on edge devices are required.

9 Reproducibility Statement

All stage-specific models, configuration files, automated tests, and deterministic report/plot-generation scripts are versioned in the project repository under a fixed random seed. All automated tests in the validated test suite passed at the reported commit. The repository is available at: https://github.com/shrutinag18/ FIS-Compression-Research.

10 Conclusion & Future Work

This work demonstrates that a Mamdani FIS can be progressively compressed using a staged, constraint-aware pipeline, reducing rules from 15 to 12 and serialized size from 8,450 to 458 bytes while improving the project-specific interpretability score from 63.500 to 72.325. Accuracy remains nearly unchanged ($\Delta = 0.000231$), but recall and F1 decline after membership-function compression, and 674 records require fallback handling. Float16 and Int8 representations violate monotonicity, so only compact Float32 serialization is accepted.These results establish the feasibility of constraint-guided FIS compression under controlled conditions. The next logical step is taking this framework out of a controlled environment. We plan to validate the model on external datasets and verify our interpretability scores with actual human subjects. Future research must include, therefore, statistical analysis through several random seeds as well as physical evaluation on edge devices in order to derive empirical data about memory, latency, and power costs.

References

[1] Zadeh, L.A.: Fuzzy sets. Information and Control, vol. 8, no. 3, pp. 338–353 (1965).
[2] Ojha, V., Abraham, A., Snášel, V.: Heuristic design of fuzzy inference systems: A review of three decades of research. Engineering Applications of Artificial Intelligence, vol. 85, pp. 845–864 (2019).
[3] Gacto, M.J., Alcalá, R., Herrera, F.: Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. Information Sciences, vol. 181, no. 20, pp. 4340–4360 (2011).
[4] Ouifak, H., Idri, A.: A comprehensive review of fuzzy logic based interpretability and explainability of machine learning techniques across domains. Neurocomputing, 647, p. 130602 (2025).
[5] Cajas Ordóñez, S.A., Samanta, J., Suárez-Cetrulo, A.L., Carbajo, R.S.: Intelligent edge computing and machine learning: A survey of optimization and applications. Future Internet, vol. 17(9), p. 417 (2025).

[6] Camastra, F., Ciaramella, A., Salvi, G., Sposato, S., Staiano, A.: On the interpretability of fuzzy knowledge base systems. PeerJ Computer Science, vol. 10, p. e2558 (2024).
[7] Azam, M.H., Hasan, M.H., Hassan, S., Talpur, N., Azam, M.H.: Explainable fuzzy modeling through antecedent reduction and antecedent association in Type-1 ANFIS system. Journal of Computational and Cognitive Engineering (2026).
[8] Alangari, N., El Bachir Menai, M., Mathkour, H.: Assessing interpretability of data-driven fuzzy models: Application in industrial regression problems. Expert Systems, vol. 41, no. 12, p. e13710 (2024).
[9] Jalilvand, A.H., Panahi, P.H.S., Najafi, M.H.: Hardware-enabled fuzzy inference: Architectures, platforms, and emerging trends. arXiv:2608.04031 (2026).

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

