FADR-MLP: A Lightweight Neural Network-Based Surrogate Model for Adaptive Data Rate Control in LoRaWAN

Fernando Jorge V. Santos[1,2], Thiago Allisson R. da Silva[1], Geraldo A. Sarmento Neto[1],
Luis H. O. Mendes[1], Pedro F. F. Abreu[1], Anderson Leonardo Sanches[3], and José Valdemir dos Reis Jr[1]
[1]*Federal University of Piauí (UFPI), Teresina - PI*, [2]*Federal Institute of Maranhão (IFMA), Porto Franco - MA,*
{fernandovieira, thiago.allisson, geraldosarmento, luishenriqueom, pedroffda, valdemirreis}@ufpi.edu.br
[3]*Federal University of ABC (UFABC), Santo André - SP* anderson.sanches@ufabc.edu.br

***Abstract*—The optimization of Adaptive Data Rate (ADR) mechanisms is fundamental to the energy efficiency and scalability of LoRaWAN networks. However, implementing intelligent control algorithms on resource-constrained Internet of Things (IoT) edge devices remains a significant challenge. This paper proposes and evaluates a novel mechanism based on a surrogate model for End Devices (EDs), which employs a Multi-Layer Perceptron (MLP) artificial neural network to adjust the following transmission parameters: Spreading Factor (SF), Transmission Power (TP), and the number of measurement packets, denoted as M. The MLP network was trained on a dataset generated by a pre-existing expert system, and its fidelity was rigorously validated via K-Fold cross-validation, achieving a Coefficient of Determination (R^2) exceeding 0.92. Furthermore, a complexity analysis quantified the model's low computational cost, revealing a memory footprint of 17.77 KB and an inference cost of 8,832 floating-point operations (FLOPs). The comparative analysis reveals that while the fuzzy inference system (FIS) it emulates may be more efficient in moderate-precision scenarios, the MLP's architectural advantage for hardware execution positions it as a promising solution for implementation at the edge of LoRaWAN-based IoT systems.**

***Index Terms*—Adaptive Data Rate, Computational Efficiency, Edge Computing, LoRaWAN, Multi-Layer Perceptron.**

I. Introduction

The prominence of LoRaWAN networks stems from their long-range connectivity and low-power capabilities, optimized by the Adaptive Data Rate (ADR) mechanism. However, the standard ADR implementation exhibits notorious deficiencies under volatile channel conditions, leading to packet loss and excessive energy consumption. To overcome such limitations, the literature has proposed Fuzzy Inference Systems (FIS) to robustly optimize transmission parameters such as SF, TP [1] [2], and the number of transmissions M [3]. Despite their effectiveness, the inherent computational complexity of FIS represents a substantial barrier to their implementation on edge devices [4], which are severely constrained in processing, memory, and power [5]. Consequently, embedding this intelligence for real-time adaptation remains a fundamental research gap for scalable and mobile LoRaWAN networks [5], [6].

To explore these gaps, this paper introduces FADR-MLP, a solution that conceptualizes and evaluates a lightweight surrogate model based on an MLP. The primary objective is not to redefine the optimization logic, but rather to make the expert fuzzy controller proposed in [3] more computationally efficient. The model is validated through experiments, and the estimated results demonstrate its ability to mimic the decisions of the fuzzy system with high precision, exploring the feasibility of implementing adaptive intelligence directly on the low-cost microcontrollers inherent to IoT devices.

The authors would like to acknowledge the support provided by IFMA.

The contributions of this work are therefore focused on three main axes:

- A surrogate model approach is introduced to compress the logic of a complex fuzzy controller into a lightweight MLP.
- The model's fidelity in replicating expert decisions is rigorously validated using robust statistical metrics, achieving an R^2 greater than 0.92.
- A computational efficiency analysis is presented, revealing a trade-off between the MLP and the FIS it emulates, and substantiating the MLP suitability for edge devices based on its architectural advantage for hardware execution.

The remainder of this paper is structured as follows. Section II presents the related work. Section III specifies the proposed FADR-MLP architecture. Section IV details the experimental methodology, including model training and evaluation. Section V discusses the results and their implications. Finally, Section VI concludes the paper and outlines future research directions.

II. Related Work

Fuzzy Logic and neuro-fuzzy approaches, such as ANFIS, are effective in optimizing ADR parameters in various scenarios [1], [3]. However, their practical implementation on edge devices is hindered by the empirical design of rules and, crucially, by the high computational complexity of steps like defuzzification [7]. This complexity, whether from Deep Learning models or FIS, imposes a critical bottleneck for low-cost IoT devices, rendering continuous operation and battery longevity unfeasible [6].

To circumvent this limitation, the field of model compression proposes training smaller, more efficient models from more complex ones, transferring their intelligence into a compact representation [8]. Despite these advances, a gap persists in demonstrating how the logic of a FIS controller can be efficiently transferred to a lightweight neural network in LoRaWAN, as previous works have focused on optimizing network metrics with complex models [3], [6] or on channel prediction [9], [10]. This work aims to investigate this gap by proposing and evaluating FADR-MLP as a surrogate model for an existing FADR-CVM (ADR Based on Fuzzy Logic with Dynamic Number of Uplink Transmissions Adjustment for LoRaWAN Networks in Mobility Scenarios) controller, with a rigorous focus on the fidelity of its decisions and on exploring its computational feasibility for deployment on resource-constrained devices.

III. Proposed Solution

To address the conflict between complex Adaptive Data Rate algorithms and the severe hardware constraints of IoT devices, this work presents FADR-MLP: a surrogate model based on a Multi-Layer Perceptron neural network. The fundamental objective is not to devise a new control heuristic, but rather to transfer the decision-making intelligence from a robust yet computationally expensive fuzzy expert system, such as FADR-CVM, into a more efficient architecture. This approach aims to replicate the expert's behavior with high fidelity while reducing the inference cost and memory footprint, making it intrinsically suitable for embedded execution.

A. FADR-MLP Architecture

The model's architecture, a sequential feedforward network composed of linear layers and non-linear activations, was deliberately designed to balance non-linear representation capability with computational efficiency.

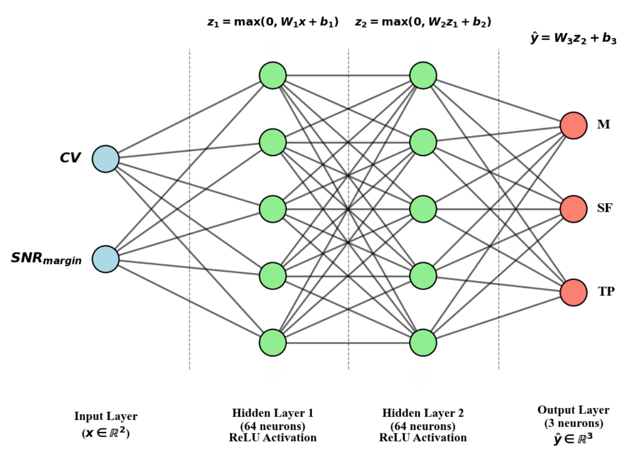

Fig. 1. FADR-MLP Architecture

The detailed architecture, illustrated in Fig. 1, consists of:

- An input layer that accepts a feature vector $x \in \mathbb{R}^2$.
- A first hidden layer composed of 64 neurons, which applies the affine transformation $W_1x + b_1$, where $W_1 \in \mathbb{R}^{64\times2}$ is the weight matrix and $b_1 \in \mathbb{R}^{64}$ is the bias vector.
- A Rectified Linear Unit (ReLU) activation function, defined as $f(z) = \max(0, z)$, applied to the output of the first layer to introduce non-linearity.
- A second hidden layer with 64 neurons, applying the transformation $W_2z_1 + b_2$, where z_1 is the output of the previous layer, $W_2 \in \mathbb{R}^{64\times64}$ and $b_2 \in \mathbb{R}^{64}$. This layer is also followed by a ReLU activation.
- A linear output layer with 3 neurons, which performs the final transformation $W_3z_2 + b_3$, with $W_3 \in \mathbb{R}^{3\times64}$ and $b_3 \in \mathbb{R}^3$, producing the final output vector.

B. Input and Output Mapping

The FADR-MLP was trained as a multivariate regression problem to learn the functional mapping $f : \mathbb{R}^2 \to \mathbb{R}^3$. The two input variables that comprise the vector x are the Signal-to-Noise Ratio margin (SNRmargin), which quantifies the quality of the communication link, and the Coefficient of Variation (CV) of the SNR, which captures its temporal instability.

The three output variables, which compose the prediction vector $\hat{y}$, are the optimal transmission parameters defined by the FADR-CVM system: the number of measurement packets, M, the Spreading Factor, SF, and the Transmission Power, TP. The model, therefore, learns to predict the optimal transmission configuration based on the current conditions and variability of the communication channel, mimicking the behavior of the expert system that trained it.

IV. Methodology

The validation of the proposed FADR-MLP was conducted via an experimental methodology, designed to assess both the fidelity of the surrogate model in replicating the expert system and its computational efficiency. This methodology encompassed the data preparation procedures, a robust training and validation strategy, and the performance metrics employed for the final evaluation.

The foundation for the training and evaluation of the FADR-MLP model was the dataset[1], which encapsulates the decision-making behavior of the FADR-CVM expert system under a wide range of simulated channel conditions. An initial exploratory analysis of the dataset revealed a pronounced imbalance in the distribution of the target variables (M, SF, and TP), a direct artifact of the expert system's optimized behavior, which could induce bias during the learning process.

To mitigate the risk of bias, a random oversampling technique was employed. Although the problem is fundamentally one of regression to predict optimal values, the discrete nature of the output variables (M, SF, and TP) allowed them to be treated temporarily as categorical classes for balancing purposes. This approach proved crucial to ensure the model learned the expert's complete behavior, mitigating bias without impairing its ability to learn the numerical relationship between the variables. Thus, samples belonging to the minority categories were duplicated until all had equitable representation. With the dataset duly balanced, the final preprocessing

[1] https://bit.ly/4lU88ju

step consisted of normalizing the continuous input variables, SNRmargin and CV.

The StandardScaler from the scikit-learn library[2] was used, which adjusts the data to have a zero mean $\mu = 0$ and a unit standard deviation $\sigma = 1$. This transformation, defined by the equation $z = \frac{x-\mu}{\sigma}$, is essential for ensuring stable and efficient convergence of the optimizer during the neural network's training phase.

To ensure the statistical robustness and generalization capability of the model, a K-Fold Cross-Validation methodology [11] was employed, implemented with the KFold function from scikit-learn version 1.5.0, using k=5 partitions. The dataset was divided into 5 folds, where, in each iteration, one fold was reserved for testing and the remaining 4 for training. The FADR-MLP model was implemented and trained using the PyTorch[3] framework version 2.3.1.

For each fold, the training was conducted using the Adam optimizer `(torch.optim.Adam)`, with an initial learning rate of $\alpha = 0.001$, and guided by the minimization of the Mean Squared Error (MSE) through the `nn.MSELoss` loss function, which calculates the average of the squared differences between the model's predictions $\hat{y}_i$ and the actual target values y_i. The process was configured for a maximum of 40 epochs, employing an Early Stopping strategy to mitigate overfitting, which halted the training if the validation loss did not improve after a patience of 5 epochs.

The primary hypothesis of this work is that the MLP can replicate the behavior of the Fuzzy model with high precision. To quantify this accuracy, four standard regression metrics were calculated for each fold on the test set: the Mean Squared Error (MSE), the Mean Absolute Error (MAE), the Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2). The final results were consolidated by calculating the mean and standard deviation of each metric across the 5 folds.

The core of this article's contribution lies in demonstrating the efficiency of FADR-MLP, whose complexity and performance analysis was structured based on four fundamental metrics. The Memory (Footprint) was calculated from the total number of model parameters (4,547), assuming a 32-bit floating-point representation. The Computational Cost FLOPs was systematically obtained using the ptflops library[4], which analyzes the architecture, calculates the Multiply-Accumulate (MACs) operations, and converts them to FLOPs (1 MAC = 2 FLOPs).

To evaluate practical performance, the Inference Latency was empirically measured in microseconds (μs) by averaging the execution time of 10,000 inferences after a warm-up period, using the `time.perf_counter` [5] function. Finally, the Estimated Energy Consumption was calculated as a proxy for energy efficiency, using the measured latency and the Thermal Design Power (TDP) of the host processor, allowing for a robust relative comparison between the models.

[2]https://scikit-learn.org

[3]https://pytorch.org

[4]https://pypi.org/project/ptflops/

[5]https://www.python.org/

V. Results and Discussion

The proposed model's evaluation was conducted on two complementary fronts: first, validating the surrogate model's fidelity in accurately replicating the expert system's outputs, and second, performing a computational complexity analysis to determine its implementation cost in terms of memory, FLOPs, latency, and energy.

A. Fidelity Assessment of the Surrogate Model

To quantify the precision with which the FADR-MLP mimics the behavior of the FIS, the model was subjected to a rigorous cross-validation process. The performance metrics were calculated on the test set of each fold, and the consolidated results, presenting the mean and standard deviation, are displayed in Table I.

TABLE I
Fidelity Metrics of the FADR-MLP Model (Mean and Standard Deviation over 5 Folds)

Metric	Mean Value	Standard Deviation
MSE	0.610008	0.00555
MAE	0.585500	0.00930
RMSE	0.692600	0.00330
R^2	**0.923100**	0.00120

The analysis of the results demonstrates the high fidelity of the proposed model. The consistently low values for the error metrics (MSE, MAE, and RMSE) indicate a small divergence between the outputs predicted by the MLP and the target values generated by the FIS. The average Coefficient of Determination (R^2) of 0.9231 is particularly noteworthy, indicating that the FADR-MLP model is capable of explaining more than 92% of the variance in the decisions of the original FADR-CVM system.

Additionally, the low standard deviation observed across all metrics over the 5 folds attests to the stability and robustness of the training process, suggesting that the model has an excellent generalization capability and is not subject to significant variations arising from data partitioning. Such results empirically validate the premise that the MLP can act as an effective and high-precision substitute for the FADR-CVM.

B. Computational Complexity Analysis of the MLP Model

Having established the model's fidelity, the second stage of the analysis focused on quantifying its computational efficiency, a fundamental requirement for embedded implementation. The complexity metrics for the FADR-MLP's architecture are summarized in Table II.

TABLE II
Computational Complexity Metrics for a Single FADR-MLP Inference

Metric	Calculated Value
Total Number of Parameters	4,547
Estimated Memory (Footprint)	17.76 KB
MAC Operations per Inference	4,416
Estimated FLOPs per Inference	8,832

The model has a total of 4,547 trainable parameters. Assuming a 32-bit floating-point representation (4 bytes) per parameter, this translates to a static memory footprint of only 17.76 KB. This value is compatible with the random-access memory (RAM) of the vast majority of low-cost microcontrollers commonly used in IoT nodes. From a processing cost perspective, the execution of a single inference requires 8,832 FLOPs.

C. Comparative Complexity Analysis

To unequivocally contextualize the efficiency of the FADR-MLP model, which was experimented in Python, its complexity metrics were contrasted with an estimate of the FADR-CVM, which was executed in the Network Simulator (NS-3)[6]. The complexity of a FIS, particularly in the Centroid defuzzification step, is directly proportional to its output resolution, a parameter denoted here as k (the number of discretization steps).

TABLE III
COMPARISON OF PERFORMANCE AND COMPUTATIONAL COST METRICS.

Mechanism	Memory	FLOPs	Latency	Energy
FADR-CVM (k=500)	2.17 KB	4,557	75.81 μs	3411.65 μJ
FADR-CVM (k=1000)	2.17 KB	9,057	86.74 μs	3902.04 μJ
FADR-MLP	17.77 KB	8,832	80.54 μs	3624.28 μJ

Table III presents this comparative analysis for two FADR-CVM resolution scenarios: a moderate precision scenario (k=500), another high precision scenario (k=1000), and the last one is the proposed MLP model where measurements for a single inference are taken.

The comparative analysis reveals a fundamental trade-off. While the FIS exhibits a lower computational cost in moderate-precision scenarios under the current simulation environment, the strategic advantage of the MLP lies in its architecture, which is based on massively parallelizable matrix multiplications. The MLP's operations benefit from hardware acceleration, resulting in a theoretical constant inference complexity, O(1). In contrast, the computational bottleneck of the FIS emerges in the defuzzification step, whose complexity, O(k), scales linearly with the output precision and is dominated by sequential logic that is difficult to optimize.

This architectural particularity substantiates the hypothesis that the MLP, despite showing comparable latency results in this simulation environment, will exhibit superior performance when deployed on actual embedded hardware. The choice, therefore, is not based solely on the current results, but on the model's dynamic potential and the suitability of its architecture for the target hardware, solidifying the surrogate approach as a promising path for practical implementation in IoT devices.

VI. Conclusions and Future Work

This work demonstrated the feasibility of an MLP as a high-fidelity surrogate model for a FIS in LoRaWAN networks, capable of replicating over 92% of the variance in its decisions. The analysis revealed a trade-off where the MLP's strategic advantage lies in its parallelizable architecture, even though the FIS may exhibit a lower computational cost in moderate-precision scenarios.

This architectural particularity substantiates the hypothesis that the MLP, despite comparable latency in simulations, will exhibit superior performance on actual embedded hardware. The choice of this approach is justified by this potential and by its focus on computational feasibility—a critical gap in the literature, which predominantly optimizes for network metrics—solidifying the proposal as a promising path.

Validating this hypothesis is the focus of future work. The next steps include: implementing and validating the FADR-MLP on a microcontroller to empirically quantify latency and energy gains; integrating it into a network simulator to assess its impact on Key Performance Indicators (KPI) such as Packet Delivery Ratio (PDR); and exploring quantization techniques for hardware optimization.

[6]https://www.nsnam.org/releases/ns-3-43/

References

[1] R. Kufakunesu, G. Hancke, and A. Abu-Mahfouz, "A fuzzy-logic based adaptive data rate scheme for energy-efficient lorawan communication," *Journal of Sensor and Actuator Networks*, vol. 11, p. 65, October 2022.

[2] G. A. Sarmento Neto, T. A. R. da Silva, P. F. F. Abreu, A. F. D. S. Veloso, L. H. D. O. Mendes, and J. V. dos Reis Junior, "A fuzzy logic approach to adaptive data rate for mobile lorawan environments," in *2024 XIV Brazilian Symposium on Computing Systems Engineering (SBESC)*, 2024, pp. 1–6.

[3] F. J. V. Santos, G. A. S. Neto, T. A. R. da Silva, P. F. F. de Abreu, L. H. de O. Mendes, and J. V. dos Reis Junior, "Adr based on fuzzy logic with dynamic number of uplink transmissions adjustment for lorawan networks in mobility scenarios," in *Proceedings of the Brazilian Conference on Intelligent Systems (BRACIS)*. Teresina, Piauí, Brazil: Federal University of Piauí (UFPI), 2025, aceito para publicação (Accepted for publication - BRACIS 2025).

[4] H. Wang, P. Pei, R. Pan, K. Wu, Y. Zhang, J. Xiao, and J. Yang, "A Collision Reduction Adaptive Data Rate Algorithm based on the FSVM for a Low-Cost LoRa Gateway," *Mathematics*, vol. 10, no. 21, p. 3920, 2022.

[5] M. A. Lodhi, L. Wang, A. Farhad, K. I. Qureshi, J. Chen, K. Mahmood, and A. K. Das, "A contextual aware enhanced LoRaWAN adaptive data rate for mobile IoT applications," *Computer Communications*, vol. 232, p. 108042, 2025.

[6] A. Farhad and J.-Y. Pyun, "AI-ERA: Artificial intelligence-empowered resource allocation for LoRa-Enabled IoT applications," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 12, pp. 11 640–11 653, 2023.

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

[8] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 160–174, 2020.

[9] M. González-Palacio, D. Tobón-Vallejo, L. M. Sepúlveda-Cano, M. Luna-Delrisco, C. Röhrig, and L. B. Le, "Machine-learning-assisted transmission power control for lorawan considering environments with high signal-to-noise variation," *IEEE Access*, vol. 12, pp. 54 466–54 487, 2024.

[10] S. A. Bhat, N.-F. Huang, I. Hussain, and U. Sajjad, "Correlating the ambient conditions and performance indicators of the lorawan via surrogate gaussian process-based bidirectional lstm stacked autoencoder," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 3413–3427, 2023.

[11] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013, chapter 5: Resampling Methods.