FADR-MLP Lite: Compact Neural Adaptive Data Rate for Mobile LoRaWAN

Fernando J. V. Santos[1,2], **Thiago A. R. da Silva**[1], **Geraldo A. S. Neto**[1], **Isaac S. S. Ramos**[1], **Alex F. dos Santos**[3,4], **Anderson L. Sanches**[4], **Jose Valdemir dos Reis Jr.**[1]

[1]Federal University of Piauí (UFPI), Teresina - PI – Brazil

[2]Federal Institute of Maranhão (IFMA), Porto Franco – MA – Brazil

[3]Federal University of Recôncavo da Bahia (UFRB), Feira de Santana - BA – Brazil

[4]State University of Feira de Santana (UEFS), Feira de Santana - BA – Brazil

[5]Federal University of ABC (UFABC), Santo André - SP – Brazil

{fernandovieira, thiago.allisson, geraldosarmento}@ufpi.edu.br

alex.ferreira@ufrb.edu.br, anderson.sanches@ufabc.edu.br

{isac_ramos1, valdemirreis}@ufpi.edu.br

Abstract. *The computational complexity of Adaptive Data Rate (ADR) algorithms in mobile LoRaWAN challenges their execution on resource-constrained devices. This work validates FADR-MLP Lite, a surrogate modeling strategy that compresses a fuzzy controller into a compact neural network. Through co-simulation with a network simulator, technical feasibility was evidenced: a 92% reduction in parameters, a 1.45 KB memory footprint, and only 672 floating-point operations per inference. Under mobility, neural generalization acted as a regularizer, outperforming the expert system that trained the network with a 19% energy gain while sustaining a packet delivery ratio above 60% in high-density scenarios, validating its robustness for low-cost Internet of Things (IoT) devices.*

1. Introduction

The consolidation of the Long Range Wide Area Network (LoRaWAN) standard in the Internet of Things (IoT) is driven by its unique ability to deliver long-range communication while preserving device battery life. Characterized as a Low Power Wide Area Network (LPWAN) technology, LoRaWAN typically employs a star-of-stars topology where End Devices (EDs) communicate via single-hop wireless links to one or more gateways, which then forward data to a centralized Network Server. The optimization of this trade-off between communication range and energy consumption is managed by the Adaptive Data Rate (ADR) mechanism. To overcome such limitations, recent literature has expanded the boundaries of intelligent control. New approaches range from topology learning via Graph Neural Networks (GNN) for energy optimization in multi-hop networks [Yang et al. 2025], to effective feature selection techniques for Spreading Factor (SF) prediction in mobile scenarios [Prakash et al. 2025]. In parallel, machine learning-assisted transmission power control has been proposed to address environments with high signal-to-noise variation [González-Palacio et al. 2024]. In this context, Fuzzy Inference Systems have established themselves as a robust approach to model channel uncertainty, optimizing parameters such as SF and Transmission Power (TP) [Kufakunesu et al. 2022, Sarmento Neto et al. 2024], as well as the number of transmissions (M) to ensure reliability [Santos et al. 2026].

Despite the algorithmic effectiveness of these advanced solutions, the computational complexity inherent to fuzzy inference systems and Deep Learning models represents a substantial barrier to their practical implementation on edge devices [Wang et al. 2022]. Such devices are, by nature, severely limited in terms of processing, memory, and energy, creating a gap between the sophistication of Artificial Intelligence (AI) algorithms and the hardware reality of low-power wide-area networks [Lodhi et al. 2025a]. As recently highlighted [Ali Lodhi et al. 2025], the integration of embedded AI and AI-based resource allocation is crucial, but the direct execution of complex models remains a critical challenge for battery longevity and the scalability of mobile networks [Farhad and Pyun 2023].

Consequently, the feasibility of embedding such intelligence for proactive real-time adaptation remains a fundamental research gap. To mitigate this problem, this article advances the research initiated in [Santos et al. 2025], presenting the systemic validation of FADR-MLP Lite. The primary objective is not only to propose a new optimization logic, but rather to validate the Surrogate Modeling strategy, compressing the intelligence of an expert fuzzy controller into a compact Multi-Layer Perceptron (MLP) Neural Network. This approach makes adaptive intelligence, previously restricted to resource-abundant environments, technically feasible on low-cost microcontrollers.

Extending the preliminary results of [Santos et al. 2025], the contributions of this work are:

- The integration of FADR-MLP Lite as a neural ADR mechanism in collaboration with a network simulator. This methodological approach allows evaluating the simulated impact of inference on the network control loop and validating the model's robustness in dynamic scenarios.
- The proof of feasibility for edge execution through an architecture that compresses expert logic, reducing trainable parameters by 92% and resulting in a memory footprint of only 1.45KB.
- The demonstration that neural generalization acts as a stabilizing filter under mobility, outperforming the reference model with a 19% gain in energy efficiency and sustaining a packet delivery ratio greater than 60% even under high density.

The remainder of this article is organized as follows. Section 2 contextualizes the research, discussing the state of the art and related work. Section 3 details the proposed solution, describing the neural network architecture, the training methodology, and the adopted co-simulation strategy. Section 4 presents and discusses the experimental results, evaluating the fidelity of the surrogate model, quantifying the computational cost, and the impact on network performance in terms of packet delivery ratio, energy efficiency, and average energy consumed per device. Finally, Section 5 presents the conclusions and points out directions for future work.

2. Related Work

The literature on resource management in LoRaWAN networks has sought to overcome the limitations of standard ADR, especially in dynamic scenarios where channel variability degrades network performance. Initial predictive filtering approaches, such as Kalman filters, smoothed SNR estimates to improve SF allocation, reducing packet loss and energy consumption [Lodhi et al. 2025b]. With the advancement of Artificial Intelligence, supervised Machine Learning algorithms, such as Support Vector Machines (SVM) [Wang et al. 2022] and k-Nearest Neighbors (KNN) [Prakash et al. 2025], began to be employed to classify link quality and select communication parameters. Such ML-based approaches stand out for using these techniques to deal with network configuration complexity. At a more advanced frontier, Deep Learning models have been applied for proactive resource allocation. AI-ERA, employs Deep Neural Networks (DNNs) to model SF selection as a classification problem, demonstrating notable improvements in the Packet Success Rate (PSR) by performing allocation before each uplink transmission [Farhad and Pyun 2023]. Similarly, CA-ADR uses hybrid CNN-LSTM models for mobile applications, obtaining significant gains in energy efficiency [Lodhi et al. 2025a].

Another strand, explored by [González-Palacio et al. 2024], employs ML to predict SNR based on environmental variables, allowing for power control independent of historical data. Despite the efficacy of Deep Learning, Fuzzy Logic has consolidated itself as a robust approach to model the uncertainty inherent in wireless channels [Sarmento Neto et al. 2024]. Schemes such as FL-ADR demonstrate the ability to optimize SF and TP aiming at energy efficiency [Kufakunesu et al. 2022]. More recently, [Santos et al. 2026] proposed a fuzzy logic-based ADR specifically for mobile devices, outperforming traditional approaches. However, the integration of these techniques faces challenges. Neuro-fuzzy systems, such as ANFIS, combine fuzzy interpretability with the learning capability of neural networks, but the defuzzification stage remains computationally intensive, hindering real-time implementation [Jang 1993]. Computational complexity is, therefore, the critical bottleneck. The direct execution of complex Deep Learning models or sophisticated fuzzy systems on low-cost IoT devices compromises battery longevity [Farhad and Pyun 2023]. Aiming to enable execution at the edge, the literature has explored transfer learning via model compression. This approach uses dense models to guide the learning of lightweight models, optimizing the trade-off between accuracy and efficiency [Cheng et al. 2020].

Although previous works have focused on optimizing network metrics with complex models [Santos et al. 2026, Farhad and Pyun 2023] or predicting channel conditions [González-Palacio et al. 2024, Bhat et al. 2023], a gap persists in the systemic validation of compressed models acting directly on the network control loop. This work consolidates and not only proposes the compression of an expert fuzzy controller [Santos et al. 2026] into a compact MLP Neural Network, but also validates its execution in co-simulation with the network simulator (NS-3)[1], a well-established tool in LoRaWAN academic research that offers high-fidelity Physical and MAC layer modeling and support for large-scale simulations with realistic mobility models – critical factors for validating CV as an instability metric.

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

3. Proposed Solution

To overcome the fundamental conflict between the complexity of advanced ADR algorithms and the severe hardware constraints in IoT devices, this work presents the FADR-MLP, a more compact version of the original [Santos et al. 2025]. Although robust solutions such as the Fuzzy Adaptive Data Rate based on Coefficient of Variation and Sampling Window Size (FADR-CVM) demonstrate high effectiveness, their inherent computational cost limits their feasibility in low-power microcontrollers [Santos et al. 2026]. Therefore, our approach proposes a more compact surrogate model, based on a Multilayer Perceptron Neural Network, designed to transfer the decision-making intelligence of the fuzzy expert to a computationally more efficient architecture, replicating its behavior with high fidelity, but with drastically lower inference cost and memory footprint, indicating its technical feasibility for edge processing.

3.1. FADR-MLP Lite Architecture

The proposed neural network architecture is a fully connected feedforward network, structured to map the two-dimensional input vector: the Signal-to-Noise Ratio margin SNR_{margin}, which quantifies the quality of the communication link, and the SNR Coefficient of Variation CV, which captures its temporal instability; to a three-dimensional output vector: SF, TP, and M, being the number of measurement packets.

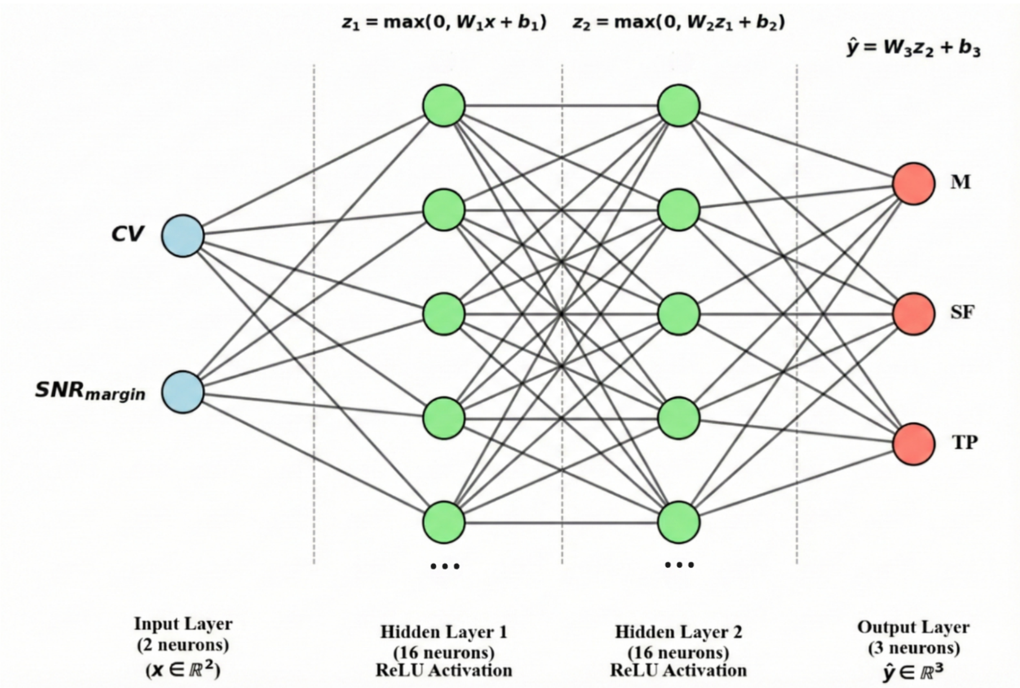


Figure 1. FADR-MLP Architecture

The proposed architecture is a fully connected feedforward network designed to map the input vector $\mathbf{x} \in R^2$ (SNR_{margin}, CV) to the output vector $\hat{\mathbf{y}} \in R^3$ $(SF, TP$ and $M)$. The structure comprises an input layer, two hidden layers with 16 neurons each, and an output layer. Mathematically, the forward propagation is generalized by the recursion:

$$\mathbf{h}^{(l)} = \phi(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}) \quad (1)$$

Where $\mathbf{h}^{(0)} = \mathbf{x}$ represents the input, and $\mathbf{h}^{(l)} \in R^{16}$ is the output of the l-th hidden layer. Finally, the network prediction is obtained at the output layer:

$$\hat{\mathbf{y}} = \phi^{(out)}(\mathbf{W}^{(3)}\mathbf{h}^{(2)} + \mathbf{b}^{(3)}) \tag{2}$$

Here, $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ denote the synaptic weight matrices and bias vectors, respectively. The hidden layers employ the ReLU non-linear activation function, while $\phi^{(out)}$ fits the regression nature of the target ADR parameters.This architecture resulted from an empirical exploration of several neural network configurations, activation functions, and hyperparameter tuning, targeting a strategic balance between high predictive fidelity and reduced structural complexity to enable execution on low-cost devices.

3.2. FADR-CVM Simulation

The FADR-CVM constitutes a resource allocation mechanism for LoRaWAN networks designed specifically to mitigate channel uncertainties inherent to mobility scenarios. Unlike the standard ADR algorithm, which assumes linearity and stability in propagation conditions, it employs a two-input fuzzy inference system that integrates the normalized SNR_{margin} and, innovatively, the link CV. The use of CV allows the mechanism to quantify the instability and fluctuations of the communication channel independently of the absolute SNR scale, serving as a robust statistical indicator of the end device's mobility dynamics.

Table 1. Parameterization employed.

Simulation Parameter	*Value*
Area dimensions	5 km x 5 km
Number of EDs	200, 400, 600, 800, 1000
Packet rate	144 packets/day
Simulation time	24 hours
Packet size	30 bytes
Mobility model	2D Random Walk
ED movement speed	[0.5 , 1.5] m/s
Propagation model	log-distance
Shadowing model	110 m
Path loss exponent	3.76
Carrier frequency	868 MHz (EU-868)
Bandwidth	125 KHz
Coding rate	4/8
Energy Parameters (SX1272)	*Value*
Supply Voltage	3.3 V
Tx Current	28 mA
Rx Current	11.2 mA
Standby Current	1.4 mA
Sleep Current	1.5 μA

Operationally, the algorithm performs statistical pre-processing on a sliding window of SNR measurements before the inference stage. The SNR_{margin} normalization adopts an approach based on the median and the interquartile range, a technique that confers immunity to outliers and prevents impulsive noise from distorting the mapping to the universe of discourse of the membership functions. The inference engine, based on the Mamdani method with a base of nine linguistic rules and centroid defuzzification, simultaneously controls three transmission parameters: the SF, the TP, and the sampling window size M for future uplink transmissions.

The defining characteristic of FADR-CVM is the modulation of system responsiveness through the dynamic adaptation of the historical window M, guided by the CV. In this mechanism, the controller adjusts the size of M inversely to the channel instability: while high-volatility scenarios (high CV) require reduced windows to ensure agile adaptation, stable channels (low CV) benefit from larger windows for noise smoothing. To validate this approach, a dataset was generated by simulating a modified version of FADR-CVM with an expanded spectrum of windows ($M \in \{2, 3, 4, 5, 6, 7, 8\}$) using a network simulator compliant with the LoRaWAN protocol specifications [Sarmento Neto et al. 2024, Santos et al. 2026]. The implementation of this fuzzy logic-based system was facilitated by the Fuzzylite library[2].

3.3. Model Training

The foundation for the training and evaluation of the FADR-MLP model was the database[3] containing 144,000 records, which encapsulates the decision-making behavior of the FADR-CVM expert system under a wide range of simulated channel conditions. The model was trained as a multivariate regression problem to learn the functional mapping $f : R^2 \rightarrow R^3$. The two input variables that constitute the vector x are SNR_{margin} and CV. The three output variables, which compose the prediction vector $\hat{y}$, are: M, SF, and TP.

The core of the learning process lies in the iterative minimization of a cost function $J(\theta)$, defined in this work by the Mean Squared Error (MSE), which quantifies the global divergence between the network prediction $\hat{\mathbf{y}}$ and the target vector $\mathbf{y}$ generated by FADR-CVM. To optimize the free parameters of the network (θ), the error backpropagation algorithm is employed, which efficiently computes the gradient of the cost function $\nabla_\theta J(\theta)$ by applying the chain rule to propagate the error from the output layer backwards through the hidden layers. The synaptic weight update is performed by the optimizer, which adjusts the parameters in the opposite direction of the gradient to find the global minimum of the error surface, according to the general update rule:

$$\theta \leftarrow \theta - \eta \cdot \nabla_\theta J(\theta)$$

Where η represents the learning rate, a critical hyperparameter that modulates the magnitude of the adjustment at each step. This mechanism ensures that the model internalizes the non-linear dynamics of the fuzzy expert, adjusting its internal connections to replicate the ADR decision policy. This iterative process allows the neural network model to internalize the fuzzy logic [Santos et al. 2026]. To ensure statistical robustness and the model's generalization capability, as shown in Fig. 2, a *K-Fold* Cross-Validation

[2] `https://fuzzylite.com/`
[3] `https://bit.ly/41U88ju`

methodology was employed, implemented with the *KFold* function from the scikit-learn library, using $k = 5$ partitions. The dataset was divided into 5 subsets (*folds*), where, in each iteration, one subset was reserved for testing and the remaining 4 for training. The proposed model was implemented and trained using the PyTorch library[4].

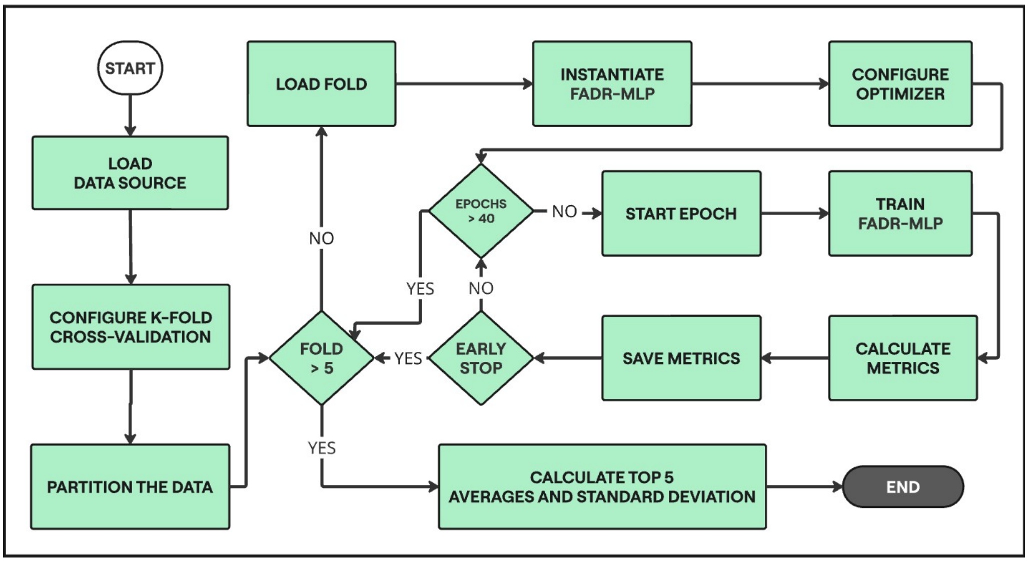


Figure 2. Training and validation diagram of the FADR-MLP model

For each partition, training was conducted using the Adam optimizer (`torch.optim.Adam`) for offering faster and more efficient convergence on complex error surfaces with noisy gradients in regression problems, with an initial learning rate of $\alpha = 0.001$, and guided by the minimization of the MSE via the loss function `nn.MSELoss`, which calculates the average of the squared differences between the model predictions $\hat{y}_i$ and the actual target values y_i, as during training and cross-validation, it guides the optimizer to penalize larger errors more severely during weight adjustment. The process was configured for a maximum of 40 epochs, employing an Early Stopping strategy to mitigate overfitting, which interrupted training if the validation loss did not show improvement after a patience of 5 epochs.

This methodology was adopted with the hypothesis that the model is capable of replicating the behavior of the fuzzy model with high precision. To quantify this accuracy, four standard regression metrics were calculated for each partition (fold) in the test set: the MSE, which consists of the average of the squared errors; the Mean Absolute Error (MAE), being the average of the absolute value of the errors; the Root Mean Squared Error (RMSE), which returns the error to the original unit of the outputs; and the Coefficient of Determination (R^2), which indicates the proportion of variance explained by the model, where a value close to 1 denotes a near-perfect fit. The final results were consolidated by calculating the mean and standard deviation of each metric across the 5 partitions.

3.4. Online Simulation

The validation of the proposed model in a dynamic and stochastic environment requires the integration between the discrete-event simulator, responsible for modeling the physical and MAC layers of the LoRaWAN protocol, and the execution environment of the

[4] `https://pytorch.org`

trained neural model. To enable this evaluation, as illustrated in Fig. 3, a co-simulation architecture was developed based on inter-process communication via a shared file system.

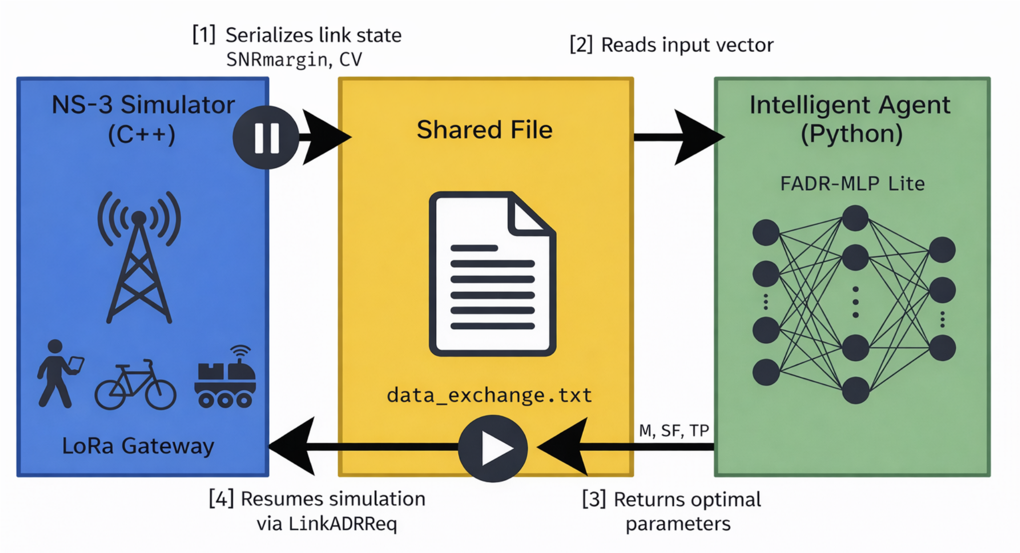


Figure 3. Illustrative scheme of the simulation in NS-3 with the trained FADR-MLP

This approach establishes a loose coupling interface that allows the network simulator to delegate the ADR decision logic to the external intelligent agent without the need to reimplement the complex matrix operations of the MLP in C++. The operational flow occurs in a synchronous and cyclic manner, ensuring temporal consistency between the state of the simulated network and the model inference. The process is governed by the following interaction protocol:

1. State Acquisition and Writing (NS-3 $\rightarrow$ File): Upon receiving an uplink packet(data transmitted from the End Device(ED) to the gateway) at the Network Server, which manages the network administration, the network management module of the simulator calculates the link quality metrics. The simulator then serializes the input feature vector, composed of the SNR_{margin} and the Coefficient of Variation CV, writing them to the interface file. At this moment, the event execution in NS-3 enters a blocking wait state, awaiting the response from the Python server.
2. Surrogate Model Inference (Python): A Python script loads the trained model, operating in a monitoring loop, detecting the update of the interface file. The input vector $\mathbf{x} \in R^2$ is read and submitted to the previously trained proposed model network. The inference process generates the prediction vector $\hat{\mathbf{y}} \in R^3$, determining the optimal transmission parameters: M, SF, and TP.
3. Actuation and Resumption (File $\rightarrow$ NS-3): The inferred parameters are written back to the interface file, overwriting the previous request. Upon detecting the availability of the response, the simulator reads the new configuration parameters and resumes the execution flow. Then, the Network Server schedules the MAC LinkADRReq command to be sent in the next downlink (transmission from the gateway to the ED), instructing the End Device to reconfigure its transmission parameters according to the model's decision.

The experimental validation in online mode was conducted in the NS-3 environment, preserving the parameterization detailed in Table 1. In this Online simulation stage, the network simulator was configured for 24 hours of transmissions, generating the necessary adjustment to ensure the collection of sufficient samples of network performance metrics, thus allowing a robust comparative analysis of the effectiveness of the proposed model acting as an ADR mechanism.

4. Results and Discussion

The experimental validation of the proposed model adopts a comprehensive perspective, aiming to ensure decision integrity and efficiency in IoT devices. The evaluation strategy is structured into three complementary axes: first, the statistical fidelity of the model in replicating the expert is analyzed; next, computational complexity, memory and Floating Point Operations (FLOPs) are examined, grounding the technical feasibility of embedding in low-cost microcontrollers; and finally, the systemic network performance is evaluated in stochastic scenarios, contrasting the neural model's effectiveness against traditional approaches under mobility conditions.

4.1. Surrogate Model Fidelity Evaluation

To quantify the precision with which the proposed model mimics the behavior of the fuzzy system[Santos et al. 2026], the model was submitted to a rigorous cross-validation process. The performance metrics were calculated on the test set of each partition, and the results were consolidated, presenting the best results and the standard deviation of each architecture variation, being displayed in Table 2.

Table 2. Comparison of Best Metric Results by Neuron Configuration in MLP.

Model	Neurons	Metrics: Best Value (Standard Deviation)			
		MSE	MAE	RMSE	R²
FADR-MLP Lite	**2-16-16-3**	**0.7220 ± 0.0621**	**0.6114 ± 0.0137**	**0.8527 ± 0.0346**	**0.9199 ± 0.0061**
FADR-MLP	2-64-64-3	0.6100 ±0.0056	0.5855 ±0.0093	0.6926 ±0.0033	0.9231 ±0.0012

Although the analysis of Table 2 indicates that the architecture with 64 neurons presents marginally superior performance 0.9231 versus 0.9199 in R^2, the more compact configuration was selected as the final model. This decision reflects a strategic compromise between predictive fidelity and structural complexity, since the increase to 64 neurons yields a precision gain of less than 0.5% but imposes a twelve-fold increase in trainable parameters from ≈ 371 to $\approx 4,547$. Given the central objective of feasibility in resource-constrained IoT devices, the chosen architecture offers an advantageous trade-off, ensuring high replication fidelity $> 91\%$ with a fraction of the computational cost, thereby optimizing inference latency and energy efficiency.

4.2. Model Computational Cost Analysis

Once the precision of the proposed model was established, the subsequent step consists of quantifying its computational cost to ensure the feasibility of inference at the edge. To this end, dimensions that directly impact device autonomy are evaluated: total parameters and memory occupation, metrics that define the spatial footprint of the model in microcontrollers with limited memory; and the volume of FLOPs, a standardized measure of arithmetic complexity. Minimizing FLOPs is crucial, as it reduces processor activity time during inference, mitigating energy consumption and enabling a response in a shorter time.

Table 3. Computational Cost Comparison between Architectures.

Model (Architecture)	Total Params	Memory (32-bit)	MAC Ops	Estimated FLOPs
FADR-MLP Lite *Proposed*	371	$\approx$ 1.45 KB	336	672
FADR-MLP *Comparative*	4.547	$\approx$ 17.76 KB	4.416	8.832

Note: The 16-neuron architecture reduces computational effort by over 90% compared to the 64-neuron version, maintaining fidelity $>91\%$.

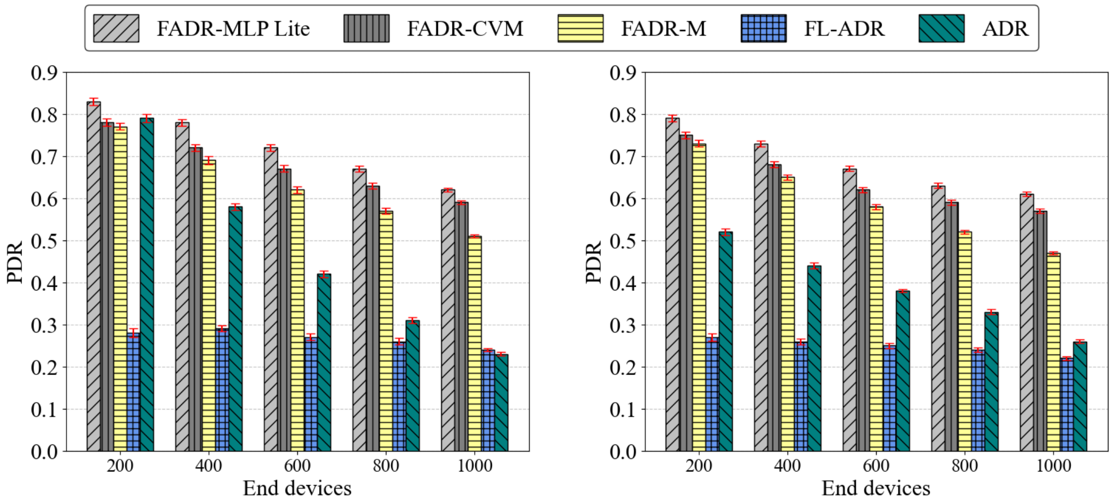
The quantitative analysis in Table 3 evidences the efficacy of the FADR-MLP Lite surrogate strategy. Contrasting with the 64-neuron variant, a reduction of approximately 92% in parameters from 4.547 to 371 decreased static memory occupation to merely 1.45 KB, fitting entry-level microcontrollers without dedicated hardware. Additionally, the demand of 672 FLOPs per inference ensures a low, deterministic computational cost, vital for avoiding current peaks and preserving battery life during wake-up operations.

In the operational context of LoRaWAN networks, these findings transcend mere numerical optimization and corroborate the computational feasibility of intelligence at the edge. The disparity in computational cost demonstrates that the search for marginally superior precision – a the gain of $<0.5\%$ in R^2 obtained by the larger network – would incur a higher processing and storage cost. Therefore, the proposed configuration reaches a balance point, allowing the complex ADR decision logic, typically centralized in resource-abundant servers, to become technically suitable for on-device execution. This suggests greater autonomy for end devices to adapt their transmission parameters with minimal latency, corroborating the hypothesis that compact neural models are fundamental enablers for the scalability of massive and energy-efficient IoT networks.

4.3. Comparative Network Performance Analysis

The effectiveness of FADR-MLP was evaluated by the Packet Delivery Ratio (PDR) varying from 200 to 1000 end devices in Fig. 4. Although PDR decay is inherent to the increase in collisions in the ALOHA protocol, the proposed model presented significantly attenuated degradation compared to the references. In the critical static scenario (1000 EDs) in Fig. 4a, FADR-MLP maintained a PDR of 0.62, outperforming standard ADR (0.23) by approximately 170%. The error bars, with a 95% confidence interval, statistically confirm the superiority of the model against FADR-CVM (0.59) and FADR-M (0.51), evidenced by the absence of overlap in the intervals.

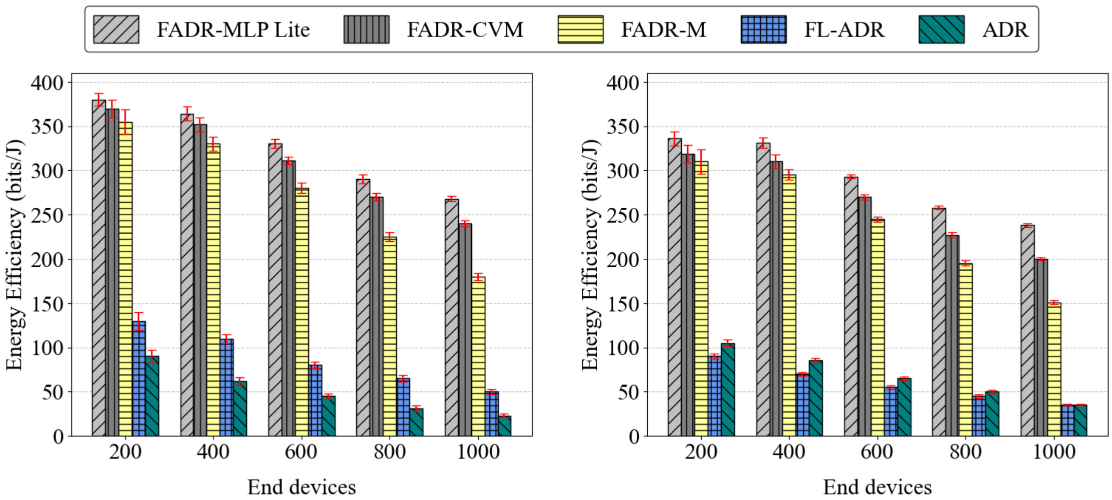
Under mobility conditions Fig. 4b, the robustness of the model is accentuated. While standard ADR suffers a performance collapse, falling from 0.52 to 0.26, due to the slow convergence of its moving average regarding channel variation, FADR-MLP sustains PDR greater than 0.60 at all densities. Inheriting the responsiveness to the CV from the expert algorithm, the neural model not only mitigated losses but also outperformed FADR-CVM at all experimental points, recording a 6% gain in this dynamic scenario.



(a) (b)

Figure 4. Average PDR - (a) Scenario Static and (b) Scenario Mobile

The systematic superiority of FADR-MLP over the original fuzzy system is attributed to the intrinsic generalization capability of neural networks. While the fuzzy controller operates with discrete rules subject to abrupt transitions and oscillations due to channel noise, MLP training via MSE minimization acts as a regularizer, smoothing the decision surface. By filtering instabilities and focusing on the optimal statistical trend, the MLP stabilizes resource allocation SF, resulting in more efficient and robust connectivity than that of the heuristic generating the data.



(a) (b)

Figure 5. Energy Efficiency - (a) Scenario Static and (b) Scenario Mobile

The energy efficiency analysis in Fig. 5, defined by the bits/Joule ratio, corroborates the superiority of the proposed model in both reliability and operational cost. In the static scenario Fig. 5a, the model sustains leadership with an efficiency of 268 under maximum load (1000 EDs), outperforming standard ADR (23) by more than an order of magnitude and FADR-CVM (240) by approximately 11%. This statistically significant advantage stems from the effective mitigation of retransmissions and the optimization of channel occupation time through more assertive SF decisions. Under mobility in Fig. 5b, the robustness of the model is amplified, sustaining an efficiency of 238 against 35 for ADR and 200 for FADR-CVM, representing a 19% gain over the expert system itself. This superior performance is due to the generalization of the neural network, which acts as a regularization filter over the rigid rules of the fuzzy controller. By smoothing abrupt parameter oscillations caused by transient noise in the CV, the MLP reduces signaling overhead and avoids the premature allocation of energetically costly SFs, maximizing the network's energy efficiency.

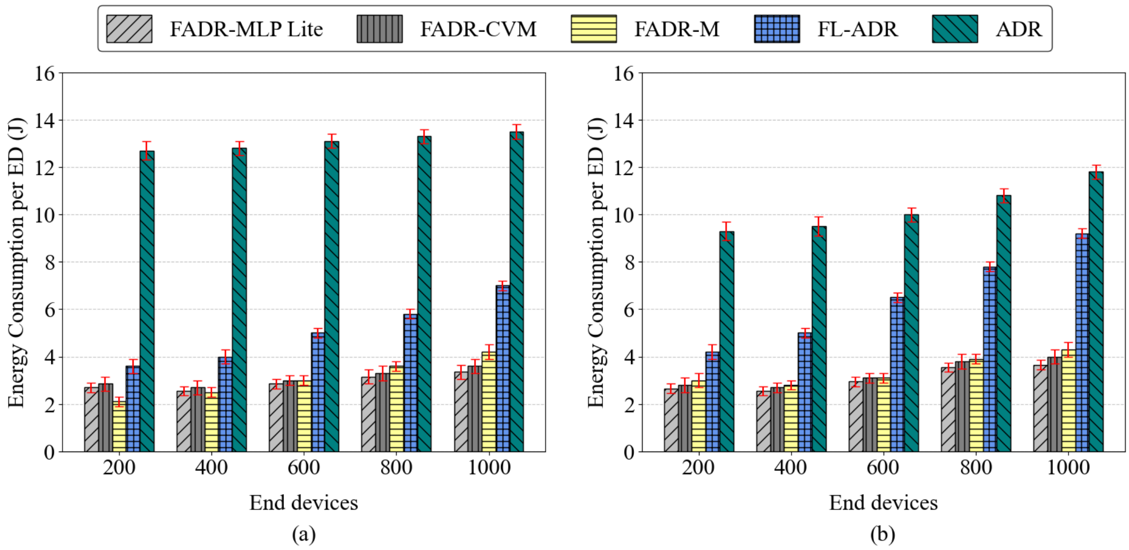


Figure 6. Energy Consumption - (a) Scenario Static and (b) Scenario Mobile

The analysis of the average energy consumption per device Fig. 6 evidences the operational efficiency of the proposed model. In the static scenario Fig. 6a, a significant disparity is observed: while standard ADR consumes approximately 13.5 J at maximum load (1000 EDs) due to the conservative use of SFs and excessive retransmissions, FADR-MLP Lite maintains consumption at 3.35 J, statistically equivalent to the fuzzy expert, confirming that compression preserved the allocation intelligence, ensuring savings of over 75%. Under mobility Fig. 6b, robustness was determinant; unlike FL-ADR and ADR, which reach peaks of 9.20 J and 11.8 J respectively, the neural model demonstrates a nearly flat curve, ending with only 3.65 J. This stability is justified by the generalization capability—stemming from training based on Mean Squared Error (MSE) acting as a regularizer that filters noisy oscillations of the CV, avoiding unnecessary parameter switches and marginally outperforming even the efficiency of the expert algorithm.

5. Conclusions and Future Work

Advancing the research in [Santos et al. 2025], this work consolidates FADR-MLP as a computationally efficient solution for resource management in IoT. While the preliminary study focused on architecture, this continuation proves that the surrogate modeling strategy maintains the high determination degree of the expert fuzzy controller while significantly reducing the processing overhead. The crucial advance lies in the systemic validation via co-simulation, confirming that replacing the standard ADR with this model-which features a 92% reduction in parameters and requires only 672 FLOPs - did not compromise efficacy. This reduction in computational complexity demonstrates the model's suitability for resource-constrained environments, potentially easing the processing demand on network servers.

Experimental results revealed that the neural model not only replicated the expert system with high fidelity but surpassed its performance. FADR-MLP Lite acted as a regularization filter, smoothing abrupt fuzzy rule transitions and mitigating transient channel noise. This characteristic resulted in superior robustness, maintaining a PDR above 60% under high density, whereas standard ADR collapsed to 26% and ensuring an average consumption reduction greater than 75% in static scenarios, 3.35 J versus 13.5 J. Under mobility, the model demonstrated critical stability in 3.65 J, avoiding the exponential consumption growth of traditional approaches. Consequently, energy efficiency was significantly expanded, registering a 19% gain relative to the reference algorithm itself in dynamic scenarios, validating the hypothesis that neural generalization contributes to a more stable and economical resource allocation.

As future work, we intend to advance on three main fronts. First, investigate post-training quantization techniques to further reduce the memory footprint, allowing portability to resource-constrained microcontrollers. Second, expand model training to contemplate different traffic patterns and more complex urban mobility models, evaluating the neural network's generalization capability in heterogeneous network topologies. Finally, we aim to perform an experimental validation on embedded hardware to measure real energy consumption and inference latency, comparing these bench results against simulated data.

Acknowledgments

The authors would like to thank the IFMA – Porto Franco Campus for the institutional support and for granting the leave for graduate studies. We also express our gratitude to the Littoral Laboratory for the opportunity to contribute to the developed research. This work was partially supported by UEFS-AUXPPG 2026 and CAPES-PROAP 2026 grants.

References

Ali Lodhi, M., Sun, X., Mahmood, K., Lodhi, A., Park, Y., and Hussain, M. (2025). AI-Enhanced Resource Allocation for LPWAN-Based LoRaWAN:A Hybrid TinyML and Deep Learning Approach. *IEEE Internet of Things Journal*, 12(14):28950–28963.

Bhat, S. A., Huang, N.-F., Hussain, I., and Sajjad, U. (2023). 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*, 20(3):3413–3427.

Cheng, Y., Wang, D., Zhou, P., and Zhang, T. (2020). A Survey of Model Compression and Acceleration for Deep Neural Networks. *IEEE Signal Processing Magazine*, 37(5):160–174.

Farhad, A. and Pyun, J.-Y. (2023). AI-ERA: Artificial Intelligence-Empowered Resource Allocation for LoRa-Enabled IoT Applications. *IEEE Transactions on Industrial Informatics*, 19(12):11640–11653.

González-Palacio, M., Tobón-Vallejo, D., Sepúlveda-Cano, L. M., Luna-Delrisco, M., Röhrig, C., and Le, L. B. (2024). Machine-Learning-Assisted Transmission Power Control for LoRaWAN Considering Environments With High Signal-to-Noise Variation. *IEEE Access*, 12:54466–54487.

Jang, J.-S. R. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665–685.

Kufakunesu, R., Hancke, G., and Abu-Mahfouz, A. (2022). A Fuzzy-Logic Based Adaptive Data Rate Scheme for Energy-Efficient LoRaWAN Communication. *Journal of Sensor and Actuator Networks*, 11:65.

Lodhi, M. A., Wang, L., Farhad, A., Qureshi, K. I., Chen, J., Mahmood, K., and Das, A. K. (2025a). A Contextual Aware Enhanced LoRaWAN Adaptive Data Rate for mobile IoT applications. *Computer Communications*, 232:108042.

Lodhi, M. A., Wang, L., Mahmood, K., Farhad, A., Chen, J., and Kumari, S. (2025b). Enhanced adaptive data rate strategies for energy-efficient Internet of Things communication in LoRaWAN. *International Journal of Communication Systems*, 38(2):e5966.

Prakash, A., Choudhury, N., Hazarika, A., and Gorrela, A. (2025). Effective Feature Selection for Predicting Spreading Factor with ML in Large LoRaWAN-Based Mobile IoT Networks. In *2025 National Conference on Communications (NCC)*, pages 1–6. IEEE.

Santos, F. J. V., da Silva, T. A. R., Sarmento Neto, G. A., Mendes, L. H. O., Abreu, P. F. F., Sanches, A. L., and dos Reis Jr., J. V. (2025). FADR-MLP: A Lightweight Neural Network-Based Surrogate Model for Adaptive Data Rate Control in LoRaWAN. In *Extended Proceedings of the XV Symposium on Computing Systems Engineering (SBESC)*, pages 41–44, Campinas, SP, Brazil. SBC.

Santos, F. J. V., Neto, G. A. S., da Silva, T. A. R., de Abreu, P. F. F., de O. Mendes, L. H., and dos Reis Junior, J. V. (2026). ADR Based on Fuzzy Logic with Dynamic Number of Uplink Transmissions Adjustment for LoRaWAN Networks in Mobility Scenarios. In de Freitas, R. and Furtado, D., editors, *Intelligent Systems. BRACIS 2025*, pages 74–88, Cham. Springer Nature Switzerland.

Sarmento Neto, G. A., da Silva, T. A. R., Abreu, P. F. F., Veloso, A. F. D. S., Mendes, L. H. D. O., and dos Reis Junior, J. V. (2024). A Fuzzy Logic Approach to Adaptive Data Rate for Mobile LoRaWAN Environments. In *2024 XIV Brazilian Symposium on Computing Systems Engineering (SBESC)*, pages 1–6.

Wang, H., Pei, P., Pan, R., Wu, K., Zhang, Y., Xiao, J., and Yang, J. (2022). A Collision Reduction Adaptive Data Rate Algorithm based on the FSVM for a Low-Cost LoRa Gateway. *Mathematics*, 10(21):3920.

Yang, H., Wu, X., Ji, H., Huang, Z., and Fang, J. (2025). A Topology-Aware GNN Learning Approach for Energy Optimization in Multihop LoRa Networks. *IEEE Internet of Things Journal*, 12(22):46596–46610.