Leveraging Machine Learning for Spreading Factor Optimization in Lora WAN Networks

Farhan Nisar[1], Mahad Shah[1], Muhammad Nouman[2], Shahzada Fahim Jan[2], Muhammad Nauman Khan[3], Robina Safi[4], Arshad Farhad[5], Muhammad Touseef Irshad[6] Sana Shafiq[1]
[1]Department of Physical and Numerical Sciences, Qurtaba University, Peshawar
[2]Department of Computer Systems Engineering, UET Peshawar
[3]Department of Computer Science & IT, Agriculture University, Peshawar
[4]Department of Computer Science, Abasyn University, Peshawar
[5]Department of Computer Science, Bahria University, E-8 Islamabad
[6]Department of Computer Science, National University of Modern Languages, Peshawar
***Correspondence:** farhansnisar@yahoo.com, shahmahad077@gmail.com, mnouman@uetpeshawar.edu.pk, shahzadafahimjan2001@gmail.com, nauman.marwat94@gmail.com, robinasafi223@gmail.com, arshadfarhad.buic@bahria.edu.pk, touseef.irshad@numl.edu.pk, sanashafiq454@gmail.com

Citation | Nisar F., Shah M., Nouman M., Jan S. F., Khan M. N., Safi R., Arshad F., Irshad M. T., Shafiq S, "Leveraging Machine Learning for Spreading Factor Optimization in Lora WAN Networks", IJIST, Vol. 07, Issue. 04 pp 2646-2659, November 2025
DOI: https://doi.org/10.33411/IJIST/1635
Received| October 02, 2025 **Revised|** October 17, 2025 **Accepted|** October 25, 2025 **Published|** November 04, 2025.

The Internet of Things (IoT) has witnessed exponential growth and widespread integration across diverse sectors such as agriculture, logistics, smart cities, and healthcare. Among various IoT communication paradigms, the Long-Range Wide Area Network has emerged as a prominent and preferred technology, attributed to its extended transmission range, energy efficiency, and cost-effectiveness. Nevertheless, the escalating proliferation of IoT endpoints has amplified the complexity of efficient resource orchestration, particularly in Spreading Factor (SF) optimization within infrastructures. To mitigate this challenge, this study introduces a Machine Learning–driven Adaptive Data Rate (ML-ADR) framework for dynamic SF management. A Long Short-Term Memory (LSTM) neural network was meticulously trained using a dataset synthesized via ns-3 network simulations to achieve optimal SF classification. The pre-trained LSTM model was subsequently deployed on end-device nodes to enable intelligent and adaptive SF allocation using real-time data during simulation. Experimental evaluations reveal significant enhancements in packet delivery ratio and notable reductions in energy consumption, thereby validating the efficacy and scalability of the proposed ML-ADR approach.

Keywords: Internet of Things (IoT), Machine Learning (ML), LSTM, Spreading Factor (SF), Transmission Power (TP)

Introduction:

The Internet of Things (IoT) has emerged as a disruptive technological paradigm, facilitating the seamless convergence of the physical and digital realms through a vast ecosystem of interconnected intelligent devices. By enabling ubiquitous sensing, communication, and computation, IoT has revolutionized data-driven automation and decision intelligence across multiple domains such as smart cities, precision agriculture, industrial automation, and healthcare [1].

A pivotal technological enabler underpinning this evolution is the Low-Power Wide Area Network (LPWAN), which offers long-range connectivity, minimal power consumption, and cost-effective scalability for large-scale IoT deployments. Among the leading LPWAN standards—Sigfox, Narrowband IoT (NB-IoT), Weightless, and Long-Term Evolution for Machines (LTE-M) [2][3][4][5]—the Long-Range Wide Area Network (LoRa WAN) has attained notable prominence due to its open standardization, architectural flexibility, and compatibility with heterogeneous IoT infrastructures [6].

Table 1 delineates the comparative characteristics of these LPWAN technologies. Sigfox is renowned for its minimalist architecture and economical deployment, while NB-IoT leverages existing cellular infrastructure to offer enhanced data throughput and reliability. The Weightless protocol is distinguished by its scalability and adaptive modulation schemes, and LTE-M excels in mobility support and broad coverage areas. In contrast, LoRa WAN [7] has emerged as the preeminent LPWAN standard, combining long-range transmission, energy efficiency, and operational robustness. This has led to its pervasive adoption across academic research, industrial innovation, and large-scale IoT ecosystems, solidifying its status as a cornerstone technology in the modern IoT landscape.

LoRa and LoRa WAN: An Overview:

Long Range (LoRa) constitutes the physical (PHY) layer foundation of the LoRa WAN protocol stack, leveraging Chirp Spread Spectrum (CSS) modulation to enable resilient, long-distance wireless communication. CSS encodes information using chirp signals that continuously sweep across a broad frequency spectrum, thereby enhancing immunity to interference, multipath fading, and Doppler shifts [8]. This advanced modulation technique yields an exceptionally high link budget exceeding 150 dB, facilitating transmission distances of up to 15 km in rural terrains and 2–5 km in dense urban environments [9], as depicted in Figure 1.

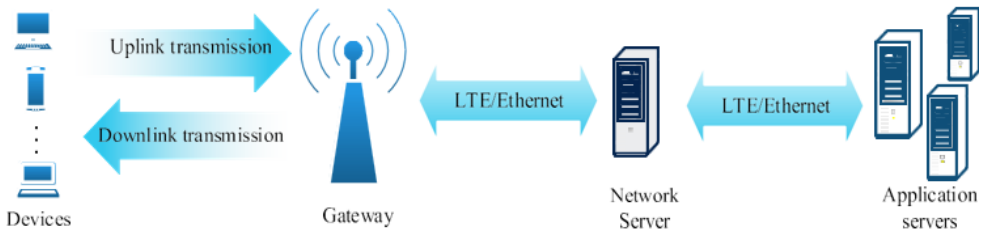

Figure 1. Shows the network and Gateway communication.

At the Medium Access Control (MAC) layer, LoRa WAN orchestrates network-level functionalities, including device authentication, adaptive data rate (ADR) optimization, and bidirectional communication management. Operating within the unlicensed Industrial, Scientific, and Medical (ISM) frequency bands—notably 868 MHz in Europe and 915 MHz in North America—Lora WAN supports variable data rates ranging from 0.3 kbps to 50 kbps, dynamically tuned in response to channel and network conditions [10].

Table 1. Comparative analysis of prominent IoT communication technologies [11][12]

Technology aspect	Lora WAN	NB-IoT	Weightless	LTE-M
Frequency band	Unlicensed ISM bands (e.g., 868 MHz EU, 915 MHz US)	Licensed LTE bands (in-band, guard-band, standalone)	Sub-1 GHz ISM bands	Licensed LTE bands
Channel bandwidth	125 kHz, 250 kHz, 500 kHz	180 kHz	12.5 kHz	1.4 MHz
Modulation scheme	CSS (Chirp Spread Spectrum)	QPSK	GMSK, QPSK	QPSK, 16 QAM
Max. application payload	51 to 242 bytes (region-dependent)	~1,600 bytes	Variable, app-defined	~1,500 bytes
Data throughput	0.3 kbps to 50 kbps	~250 kbps (downlink), ~20 kbps (uplink)	Up to 100 kbps	Up to 1 Mbps
Typical range [km]	Urban ≈ 2–5, Rural > 15	Urban ≈ 1–2, Rural ≤ 10	Urban ≈ 2	Enhanced coverage up to 10 km
Adaptive rate control	Yes	Yes	Yes	Yes
Power profile	Extremely low	Low	Low	Moderate
Mobility	Supported (handovers can be challenging)	Supported in connected mode	Supported	Full, seamless handovers
Positioning method	Uplink TDoA and RSSI [12][13]	OTDOA, E-CID	Supported	OTDOA, E-CID
Private deployment	Yes, fully supported	Yes, via network slicing	Yes	Yes, via network slicing
Two-Way communication	Fully bidirectional	Fully bidirectional	Fully bidirectional	Fully bidirectional
Network model	Public or private	Public (operator-led)	Open standard	Public (operator-led)
Available simulators [public]	Yes [14][15][16][17][18][19][20][21][22]	Yes [23]	Not publicly available	Yes

When benchmarked against competing LPWAN standards such as Sigfox and NB-IoT, Lora WAN demonstrates superior configurability, scalability, and autonomy, making it an ideal choice for private network deployments and customized Quality of Service (QoS) implementations.

Objectives:

Lora WAN Architecture and Components:

Lora WAN adopts a star-of-stars network topology (Figure 2), comprising three fundamental architectural entities that collectively ensure efficient data transmission, network scalability, and reliability.

End Devices (EDs): These are low-power sensor or actuator nodes designed to collect, process, and transmit environmental or operational data using LoRa modulation techniques. End devices are optimized for ultra-low energy consumption, achieving operational lifespans between 2 and 10 years under standard duty-cycle constraints. Gateways (GWs): Functioning as intermediary relay points, gateways receive uplink transmissions from multiple end devices and forward them to the network server via standard IP backhaul connections. Gateways are capable of multi-channel, multi-spreading-factor (SF) reception and utilize the capture effect to demodulate partially overlapping signals, thereby enhancing network throughput and efficiency [24].

Network Server (NS): Serving as the core intelligence hub of the LoRaWAN ecosystem, the network server handles data deduplication, integrity verification, security management, and adaptive data rate (ADR) optimization. It also routes validated payloads to application servers, ensuring end-to-end communication integrity and QoS compliance [1].

Device Classes and Class A Operation:

Lora WAN categorizes end devices into **three operational classes—Class A, Class B, and Class C—**each tailored to distinct communication patterns, latency tolerances, and energy constraints [4]. This hierarchical classification framework empowers device manufacturers and application designers to optimize performance trade-offs between power efficiency and communication responsiveness, aligning configurations with specific IoT application requirements.

Class A devices, representing the fundamental and most energy-efficient mode, operate under an asynchronous, ALOHA-based transmission scheme. Downlink communication is permitted only during two short receive windows immediately following each uplink transmission. This battery-optimized design significantly reduces energy expenditure, making Class A highly suitable for low-duty-cycle applications, such as environmental monitoring or smart metering, where data is transmitted infrequently and moderate latency is acceptable.

Class B devices enhance this architecture by integrating scheduled receive windows, enabled through periodic beacon transmissions from the gateway. These beacons synchronize end devices with the network, facilitating deterministic downlink communication slots. Consequently, Class B is ideal for scenarios requiring timely and predictable data delivery, such as firmware updates or configuration synchronization, albeit at the cost of slightly increased power consumption compared to Class A.

Class C devices constitute the most responsive yet power-intensive configuration, maintaining near-continuous receive capability except during active transmission intervals. This mode is typically deployed in mains-powered systems or mission-critical applications demanding real-time bidirectional communication, such as industrial process control, street lighting management, or smart grid automation.

Overall, the Lora WAN device class hierarchy offers a flexible design continuum, enabling developers to strategically balance energy efficiency, responsiveness, and reliability according to the functional priorities of each deployment scenario.

Problem Statement:

Although Adaptive Data Rate (ADR) and Baseline Adaptive Data Rate (BADR) mechanisms offer foundational strategies for configuring Spreading Factor (SF) and Transmission Power (TP) in Lora WAN networks, they exhibit significant limitations in responsiveness and adaptability. Specifically, ADR tends to adjust parameters sluggishly under dynamic network conditions, whereas BADR lacks adaptability altogether, leading to suboptimal resource utilization and degraded network performance.

This underscores a critical research gap—the absence of an intelligent, context-aware resource allocation mechanism capable of rapidly adapting to fluctuating wireless environments. To address this challenge, we propose a Machine Learning–based Adaptive Data Rate (ML-ADR) framework, wherein a trained predictive model dynamically determines the optimal spreading factor for each end device using real-time contextual and historical data features.

By exploiting data-driven insights and temporal network behavior patterns, the ML-ADR approach aims to minimize packet loss, enhance packet success ratio (PSR), and optimize energy efficiency, thereby effectively mitigating the inherent shortcomings of conventional ADR and BADR schemes.

Contribution of the Paper:

The principal contributions of this research are summarized as follows:

Development of an Intelligent SF Allocation Framework:

We propose a deep neural network–based model capable of learning optimal Spreading Factor (SF) allocation strategies by capturing the intrinsic relationship between network dynamics, device distribution, and communication requirements. This design effectively addresses the long-standing challenge of adaptive SF management in Lora WAN networks.

Simulation-Based Model Training and Integration:

The deep learning model is trained using a comprehensive dataset generated through the ns-3 simulation environment, incorporating parameters such as radio propagation characteristics, end-device locations, gateway proximity, and corresponding successful SF configurations. Once trained, the pre-trained model is deployed at the Network Server (NS) to perform real-time SF optimization for end devices during network operation.

Performance Enhancement via ML-Driven Adaptation:

Through simulation-based evaluation in ns-3, the proposed Machine Learning–based Adaptive Data Rate (ML-ADR) mechanism dynamically assigns the most efficient SF values to end devices. This approach demonstrably improves packet delivery ratio (PDR) and reduces energy consumption, thereby enhancing overall network efficiency and sustainability.

Structure of the Paper:

Section 2 provides a comprehensive review of existing AI-based approaches for resource management in Lora WAN. Section 3 details the dataset collection process, identifies the essential features, and outlines the most suitable ML techniques for resource allocation based on these features. Section 4 describes the functioning of the proposed ML-ADR model. Section 5 offers an in-depth discussion of the experimental setup and offline results, while Section 6 presents the ns-3 simulation results, where the ML algorithm is applied to simulated data. Finally, Section 8 concludes the study with key findings and insights.

Literature Review:

Recent research has extensively explored machine learning (ML) paradigms to enhance Lora WAN resource allocation, particularly in spreading factor (SF) assignment, transmission power (TP) control, and device classification. These studies can be broadly grouped into three domains — reinforcement learning (RL) for dynamic SF optimization, supervised and deep

learning for intelligent decision-making, and hybrid frameworks combining both to exploit their complementary strengths.

Reinforcement learning techniques have demonstrated notable efficiency in adaptive resource allocation. [25] applied a multi-armed bandit (MAB) model, improving packet delivery ratio (PDR) and energy efficiency in simulated single-gateway setups with 100 devices. Similarly, proposed a score table-based RL algorithm, achieving 24–27% energy reduction versus traditional ADR schemes, with minimal computational overhead.

[26] introduced a dual-layer ML framework, integrating centralized supervised ML for TP control with a decentralized EXP4-based RL algorithm for SF allocation. The approach significantly enhanced network throughput and energy efficiency, especially in congested environments. However, it required continuous gateway feedback during training, slightly increasing channel overhead.

Supervised learning has proven effective for device-type classification and signal pattern recognition. A Support Vector Machine (SVM) model accurately distinguished between mobile and static nodes using minimal training data, though it lacked adaptive rate adjustment. [19] leveraged a Gated Recurrent Unit (GRU) network, achieving 96% classification accuracy and 98% PDR in medium-density networks using ns-3 simulations.

[15] implemented Fully Connected (FCNN) and Convolutional Neural Networks (CNN) for smart SF assignment and collision detection, outperforming traditional ML methods in prediction accuracy and energy optimization. However, CNN accuracy declined with increasing node density due to limited spatial correlation.

Furthermore, [13] proposed a proactive ADR mechanism using K-Nearest Neighbors (KNN) for SNR forecasting and dynamic parameter adaptation in mobile IoT nodes. Their model reduced Bit Error Rate (BER) and energy consumption, although a slight overhead occurred with larger SNR buffers.

Hybrid and Emerging Approaches:

Hybrid frameworks combining multiple ML paradigms have emerged as robust and scalable solutions. [11] merged RL-based SF allocation with ML-driven TP optimization, achieving 17% lower estimation error by fusing Lora WAN and environmental sensing data.

The surveyed literature identifies three persistent challenges motivating our proposed ML-ADR model:

Reinforcement learning (RL)-based methods demonstrate strong adaptability in dynamic environments; however, they often face challenges related to slow feedback loops and increased latency, which can hinder real-time decision-making. In contrast, supervised learning models deliver high predictive accuracy due to their reliance on labeled data but generally lack the responsiveness required for real-time network adaptation. To balance these limitations, hybrid solutions have been proposed, combining the strengths of RL and supervised learning approaches. While these hybrid models effectively manage mobility and dynamic conditions, they tend to introduce additional computational complexity, making them less suitable for resource-constrained IoT deployments.

Data Generation and Preprocessing Framework:

The proposed framework efficiently processes Lora WAN transmission data using a 20-step sequential windowing method, where each window captures essential features for optimal Spreading Factor (SF) selection derived from multi-SF transmissions.

Transmission Protocol:

Each End Device (ED) transmits identical packets simultaneously across six SFs (SF7–SF12) in confirmed mode, requiring ACKs from the Gateway (GW). For every transmission, the GW records success/failure status and signal quality metrics, while the ED logs ACK receptions as binary values (1 = received, 0 = not received). This dual logging ensures comprehensive data capture for all SFs during each transmission cycle.

Optimal SF Selection and Feature Extraction:

The optimal SF (SF*) is determined as the smallest SF that successfully receives an ACK; if none are received, SF12 is chosen by default. Each optimal transmission is linked with a feature vector (f) containing parameters such as:

f= [x, y,d,SNR,SNRreq,SNRmargin,dnorm,Prx] f = [x, y, d, SNR, SNR_{req}, SNR_{margin}, d_{norm}, P_{rx}] f=[x,y,d,SNR,SNRreq,SNRmargin,dnorm,Prx]

Where spatial coordinates *(x, y)*, distance (d)**,** signal quality metrics (SNR, SNRreq, SNRmargin)**,** normalized distance (dnorm), and received power (Prx) collectively describe the transmission environment.

Temporal Sequence Construction:

To prepare input for ML analysis, a sliding window technique constructs sequential input matrices (Xi) comprising feature vectors from 20 consecutive transmissions:

The target label (yi) for each sequence corresponds to the optimal SF (SF*_i) of the most recent transmission, enabling the model to learn temporal dependencies in SF adaptation.

The time step i within each window represents the most recent transmission in the sequence. Accordingly, every input matrix (Xi) consists of 20 temporal steps (rows) and 8 features per step (columns), yielding a total of 160 feature values per sample.

Simulations were performed using 500 End Devices (EDs) over 24 hours, where each device transmitted six confirmed uplink messages per hour, generating approximately 72,000 raw transmission events. After applying the 20-step sliding window, the final preprocessed dataset comprised 71,981 sequences, each containing 160 features and an associated optimal SF label, as summarized in Table 3.

Framework Characteristics:

The proposed data generation and preprocessing framework exhibits several notable properties relevant to Lora WAN channel modeling. By employing 20-step temporal windows, the framework inherently captures time-dependent variations in channel conditions and signal quality. The feature vector (f) provides a comprehensive, multidimensional depiction of each communication instance, integrating spatial parameters, signal strength, and quality metrics.

Furthermore, defining the target label (SF*) based on empirically successful transmissions establishes a reliable ground truth representing realistic link performance under observed conditions. The multi-SF transmission protocol enhances data collection efficiency, as each transmission simultaneously produces detailed reception and signal-quality data across all operational SF levels.

Proposed Methodology:

This study introduced a machine learning framework that utilized Long Short-Term Memory (LSTM) networks to optimize Lora WAN communication parameters, with a particular emphasis on dynamic Spreading Factor (SF) selection. The adoption of LSTM was motivated by its proven ability to capture temporal dependencies in sequential data—an essential requirement for modeling time-varying LoRa signal behaviors. Unlike traditional ML models such as Random Forests or Support Vector Machines, which processed samples independently, LSTMs effectively modeled temporal correlations between successive transmissions. This capability was particularly valuable in Lora WAN environments characterized by fluctuating channel conditions, interference variations, and node mobility, all of which influenced optimal SF decisions.

The proposed approach addressed three primary challenges in Lora WAN optimization: (1) the non-stationary nature of wireless IoT channels, (2) the trade-off between data rate and communication range in SF configuration, and (3) the need for energy-efficient

communication mechanisms. The LSTM-based temporal model captured these dynamics through a hierarchical learning architecture that processed sequences of transmission events while retaining long-term contextual memory. This approach contrasts with conventional methods that relied on static SF allocation or instantaneous channel estimation without temporal awareness.

The implemented LSTM architecture was designed to learn and model temporal dependencies across sequential Lora WAN transmission data. The network input was represented as a matrix where TTT denoted the number of time steps and DDD the number of features per step. Based on empirical analysis, T=20T = 20T=20 was chosen to incorporate sufficient temporal history from the last twenty uplink transmissions—balancing representational depth and computational efficiency. Each time step consisted of eight features, including received power (Prx), signal-to-noise ratio (SNR), spatial coordinates (x, y), distance (d), and SNR margin, as defined earlier in Equation (2).

The internal mechanism of the LSTM cell follows the standard gated architecture. At each time step *t*, the input vector *(x□)*, previous hidden state *(h□₋₁)*, and previous cell state *(C□₋₁)* are combined to compute four key components:

The Long Short-Term Memory (LSTM) network operates through four key components that manage information flow within the model. The forget gate (f□) determines which portions of past information should be retained or discarded from the cell state, allowing the model to focus on relevant patterns. The input gate (i□) controls the extent to which new information is incorporated into the memory, ensuring that only significant updates are added. The candidate cell state (Ĉ□) proposes potential modifications to the existing memory content, contributing to the learning of new temporal features. Finally, the output gate (o□) regulates how much of the updated memory is exposed to the next layer, balancing information retention with prediction output. Together, these gates enable LSTM networks to effectively model long-term dependencies in sequential data.

Here, σ and tanh represent the sigmoid and hyperbolic tangent activation functions, ensuring nonlinearity and numerical stability. This gating mechanism allows the network to retain long-term dependencies, filter irrelevant information, and adapt to dynamic signal variations over time, as illustrated in Figure 4.

LSTM Training Mechanism:

The proposed model utilizes a stacked LSTM architecture comprising two layers, each containing 128 hidden units. This configuration was selected for its strong memory capability in modeling long-term temporal dependencies within time-series data—an essential feature for identifying evolving transmission patterns in dynamic wireless channels. The first LSTM layer processes the raw sequential input, while the second layer captures higher-level temporal abstractions from the first layer's output. To ensure continuity across training batches, the network employs stateful processing, where the final hidden and cell states from one batch are propagated as the initial states for the subsequent batch.

Following the LSTM layers, the network integrates fully connected (dense) layers activated by the Rectified Linear Unit (ReLU) function (max (0, x)). These layers convert temporal dependencies learned by the LSTMs into spatial feature representations suitable for final classification. The ReLU activation introduces nonlinearity while mitigating vanishing gradient problems often associated with sigmoid or tanh functions in deeper architectures.

To address the risk of overfitting—a common challenge in Lora WAN datasets of limited size—the model employs two complementary regularization strategies:

To enhance the generalization capability of the model and prevent overfitting, two regularization techniques were employed. Dropout regularization was applied with a rate of p=0.2p = 0.2p=0.2, which randomly deactivated 20% of the neurons during training. This

mechanism compelled the network to develop more generalized and robust feature representations, reducing its dependence on specific neurons. Additionally, L2 weight regularization was incorporated into the loss function as a penalty term to discourage excessively large weight magnitudes. This approach helped prevent the model from over-specializing to the training data, thereby improving its overall stability and performance on unseen samples. These design choices collectively enhance the generalization capability and stability of the model during both training and inference phases.

The final layer uses a SoftMax activation function to generate a probability distribution over the six potential Spreading Factors (SF7–SF12). This enables adaptive decision-making, allowing the model to automatically select the SF with the highest probability or to consider additional constraints such as energy or latency requirements. The SoftMax function normalizes the output as follows:

Where yiy_iyi denotes the one-hot encoded label. The Adam optimizer is used with an initial learning rate of 10−310^ {-3}10 3 and exponential decay rates β1=0.9\beta_1 = 0.9β1=0.9, β2=0.999\beta_2 = 0.999β2=0.999. Early stopping with a patience of 10 epochs is applied to prevent overfitting while ensuring convergence.

The trained system processes Lora WAN transmission sequences through this neural pipeline, learning to predict optimal SFs based on prior channel and transmission patterns. This data-driven, adaptive method significantly outperforms static allocation strategies while remaining computationally efficient for real-world deployment on network servers.

Performance Evaluation of LSTM in Offline Mode:

For training and evaluation, the dataset consisting of 71,981 sequences was divided using a hold-out strategy to maintain temporal independence and avoid data leakage. Approximately 80% (57,585 samples) of the data were allocated for training, while 20% (14,396 samples) were reserved for testing. Furthermore, from the training set, 10% (5,759 samples) was set aside as a validation subset, which was used for hyperparameter tuning and implementing early stopping to enhance the model's generalization and prevent overfitting. Due to the sequential nature of the data, cross-validation was not applied, as retraining LSTM models on large time-series datasets is computationally expensive. Instead, validation monitoring with early stopping (patience = 10) was adopted—consistent with best practices for time-series modeling.

Table 5 compares several machine learning models for Lora WAN SF classification, including Random Forest, Gradient Boosting, SVM, KNN, XGBoost, MLP, and the proposed LSTM. Models were assessed using Accuracy, Precision, Recall, F1 Score, and computational measures such as Training Time (seconds) and Model Size (MB).

The results demonstrated that the LSTM model achieved the highest classification accuracy of 0.7290, outperforming all other models. It was followed by the MLP model (0.7193)**,** SVM (0.7123)**,** and Gradient Boosting (0.7103)**.** The Random Forest (0.6991) and XGBoost (0.6891) models showed moderate performance, while the KNN model exhibited the lowest accuracy (0.6787) among the evaluated approaches. The recall values were largely consistent with accuracy trends, whereas precision scores were generally lower—approximately 0.50–0.51 for LSTM, MLP, and SVM, and around **0.59** for KNN, XGBoost, and Random Forest. These findings confirm the superior capability of the LSTM architecture in capturing temporal dependencies within sequential Lora WAN transmission data. F1 scores showed KNN and XGBoost performing slightly better due to their precision-recall balance, despite lower overall accuracy. In terms of computational efficiency, Gradient Boosting required the longest training time (8,254.48 s), while MLP (20.31 s) and KNN (13.55 s) were the fastest. Regarding storage, Random Forest was the largest (67.76 MB), while MLP (0.21 MB), LSTM (0.84 MB), and Gradient Boosting (0.93 MB) were notably lightweight.

Lora WAN Network Performance Evaluation – Online Mode:

In the ns-3 online simulation, each device maintains a circular buffer that stores its most recent 20 transmission features in real time. After every transmission, this buffer is updated and fed into the model to determine the next Spreading Factor (SF) decision dynamically.

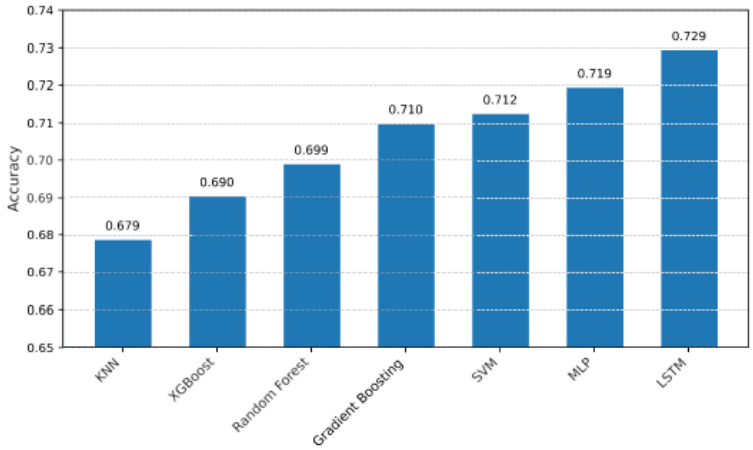

Figure 2. Visualizes the classification accuracies, clearly showing the LSTM model's superior performance, followed by MLP, SVM, and Gradient Boosting. The bar chart confirms the tabulated results, offering a clear comparative insight into model effectiveness and efficiency.

Simulation Setup and Application:

This study evaluates end devices operating in confirmed data mode within a single-gateway Lora WAN network spanning a 5 km radius. To emulate industrial asset monitoring scenarios**,** devices follow a two-dimensional random mobility pattern, changing direction after traveling 200 meters at speeds between 1.0–2.0 m/s**,** consistent with standard IoT mobility models. Each device transmits six confirmed uplink messages per hour over a 24-hour operational period. To ensure statistical robustness, ten independent simulation runs are performed, and results are presented as average performance metrics.

The experimental setup encompasses both static and mobile scenarios. In the static case, between 100 and 1,000 end devices are uniformly distributed across the network's coverage area. In mobile scenarios, the aforementioned random mobility model is applied to represent asset tracking applications.

All simulation configurations employ parameters specified in Table 6, which comply with Lora WAN regional standards for European frequency bands.

Implications of Online Mode Performance:

The online evaluation results, illustrated in Figures 6–9, highlight the operational benefits of the proposed ML-ADR framework across multiple performance metrics. In mobile deployment scenarios (Figure 6), ML-ADR achieves a 22% higher Packet Delivery Ratio (PDR) than the traditional TF baseline at network densities of 1,000 devices. This improvement arises primarily from two mechanisms: dynamic Spreading Factor (SF) adaptation based on real-time channel variations and intelligent retransmission scheduling, which minimizes acknowledgment collisions.

In static deployments (Figure 7), similar trends are observed—ML-ADR consistently outperforms conventional algorithms across all tested network scales. The energy efficiency results (Figures 8–9) further demonstrate ML-ADR's practical advantage, showing a 30% reduction in median energy consumption compared to standard ADR implementations while maintaining superior reliability. This dual gain directly addresses two key LoRaWAN challenges: limited battery capacity of end devices and the need for reliable communication in dense network conditions.

Limitations and Future Directions:

Despite its promising results, the proposed ML-ADR framework has four notable limitations:

Offline Model Training:

The model's dependence on offline training introduces latency when adapting to new environments or network conditions. While efficient for deployment, it may yield suboptimal performance during sudden environmental shifts (e.g., weather changes or unmodeled mobility). Rapidly evolving channel dynamics—such as those caused by vehicular movement in urban areas—may require retraining, delaying adaptation in real-world use.

Simplified Channel Assumptions:

The evaluation assumes ideal channel state information, which doesn't fully capture real-world conditions involving interference and multipath fading. Although ns-3 models-controlled propagation losses, actual deployments face unpredictable interference from WiFi, LPWANs, or industrial systems. These effects, alongside hardware imperfections (e.g., antenna variations or clock drifts), could reduce ML-ADR's real-world performance. Future work will involve hardware-based validation using SX1276 LoRa nodes and large-scale deployments via The Things Network, incorporating real interference data for retraining.

Single-Gateway Limitation:

The current single-gateway model simplifies evaluation but overlooks multi-cell network complexities such as handover delays, inter-gateway interference, and load balancing challenges. In multi-gateway scenarios, overlapping coverage can increase packet loss, particularly under mobility. Future extensions should explore multi-gateway topologies to assess ML-ADR's scalability in distributed environments.

Scalability Constraints:

Simulations are limited to 1,000 devices, whereas large-scale IoT networks may involve tens of thousands. The LSTM's sequence-based computation can impose heavy server-side loads in dense networks, leading to inference delays. Prior studies note similar scalability bottlenecks in ML-based Lora WAN resource allocation, where performance degrades beyond 5,000 devices due to model complexity and contention. Moreover, ns-3 data generation becomes computationally intensive at higher scales.

Future Research Directions:

Overcoming these challenges, future work focused on several key directions. First, federated learning was proposed to enable distributed model updates across gateways without requiring centralized retraining, thereby enhancing scalability and adaptability. Second, Reinforcement Learning (RL) would be employed for real-time channel estimation and interference mitigation to improve robustness under dynamic network conditions. Third, the research aimed to extend multi-gateway and multi-hop simulations to better analyze handover mechanisms and distributed coordination in large-scale Lora WAN environments. Finally, hybrid architectures combining LSTM with lightweight ML models such as TinyML were considered to support scalable edge inference and online learning, allowing the system to dynamically adapt to evolving network densities.

Algorithm for Parsing and Summarization of Lora WAN

Begin
Built table T with rows n
Define
dest-ip=1, sign-ip=2; i=1;
Aler-dscrp-strct = T (1)(signature-name, signature-class-id, priority, score-ip, ip-protocol, source-port, destination-port)
While (Length T 1 and I < length T)
For j=i+1 to Length T do
If (T(I, alter-dscrp-ip) = T(j, alert-descrp-strct))
Add the I record in the table summarized T

Delete i and j records from Table T, set i=1
Else
Merge i and j records of table T and add the resultant merge record in table T. Set i=1;
End if
End if
End for
i=i+1
End while
Add table T to table summarized T
End IF
Return summarized-T

Algorithm: The analysis algorithm for Model C

Begin
Input: test audit data during the current login session
Use CIDs to compute SaaS by aligning against in same machine
If SAS < Φ sas Then
For each cloud node (C _node) containing user I, do
Use CIDs to compute SaS, for the ith user in C _ node
If SaS > Φsas
Not-Masq-flag = True
Exit the loop
End if
End for
End if
If Not-Masq-flag = flag or HIDs instance is fired, then
Run step 2 of model A for each user CIDS instance firing
End if

Conclusions:

This study tackled the challenge of efficient Spreading Factor allocation in LoRaWAN networks, crucial for the scalability of IoT deployments. We introduced a Machine Learning-based Adaptive Data Rate (ML-ADR) mechanism leveraging LSTM models trained on ns-3 simulation data to intelligently predict and assign optimal SFs.

Comprehensive evaluations demonstrated that ML-ADR significantly improves network performance—achieving higher packet delivery ratios and lower energy consumption compared to existing methods like ADR, BADR, and TF baselines. These gains hold across both static and mobile scenarios, emphasizing the method's robustness and adaptability.

References:

[1] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, and D. Pesch, "FREE - Fine-Grained Scheduling for Reliable and Energy-Efficient Data Collection in LoRaWAN," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 669–683, Jan. 2020, doi: 10.1109/JIOT.2019.2949918.

[2] L. Acosta-Garcia, J. Aznar-Poveda, A. J. Garcia-Sanchez, J. Garcia-Haro, and T. Fahringer, "Proactive Adaptation of Data Rate in Mobile LoRa-Based IoT Devices Using Machine Learning," *IEEE Veh. Technol. Conf.*, 2024, doi: 10.1109/VTC2024-SPRING62846.2024.10683082.

[3] M. Ali Lodhi *et al.*, "Tiny Machine Learning for Efficient Channel Selection in LoRaWAN," *IEEE Internet Things J.*, vol. 11, no. 19, pp. 30714–30724, 2024, doi: 10.1109/JIOT.2024.3413585.

[4] T. R. Khola Anwar, "RM-ADR: Resource Management Adaptive Data Rate for Mobile Application in LoRaWAN," *Sensors*, vol. 21, no. 3, p. 7980, 2021, doi: https://doi.org/10.3390/s21237980.

[5] J. Y. Aloÿs Augustin, "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things," *Sensors*, vol. 16, no. 9, p. 1499, 2016, doi:

https://doi.org/10.3390/s16091466.

[6] L. Beltramelli, A. Mahmood, P. Osterberg, M. Gidlund, P. Ferrari, and E. Sisinni, "Energy Efficiency of Slotted LoRaWAN Communication with Out-of-Band Synchronization," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, doi: 10.1109/TIM.2021.3051238.

[7] N. Benkahla, H. Tounsi, Y. Q. Song, and M. Frikha, "Enhanced ADR for LoRaWAN networks with mobility," *2019 15th Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2019*, pp. 514–519, Jun. 2019, doi: 10.1109/IWCMC.2019.8766738.

[8] S. P. Matteo Bertocco, "Estimating Volumetric Water Content in Soil for IoUT Contexts by Exploiting RSSI-Based Augmented Sensors via Machine Learning," *Sensors*, vol. 23, no. 4, p. 2033, 2023, doi: https://doi.org/10.3390/s23042033.

[9] U. R. Martin Bor, "Do LoRa Low-Power Wide-Area Networks Scale?," *MSWiM 2016 - Proc. 19th ACM Int. Conf. Model. Anal. Simul. Wirel. Mob. Syst.*, 2016, doi: https://doi.org/10.1145/2988287.2989163.

[10] A. Bounceur *et al.*, "CupCarbon-Lab: An IoT emulator," *CCNC 2018 - 2018 15th IEEE Annu. Consum. Commun. Netw. Conf.*, vol. 2018-January, pp. 1–2, Mar. 2018, doi: 10.1109/CCNC.2018.8319313.

[11] L. V. der P. Gilles Callebaut, Geoffrey Ottoy, "Cross-Layer Framework and Optimization for Efficient Use of the Energy Budget of IoT Nodes," *Eur. Union*, 2018, [Online]. Available: https://arxiv.org/pdf/1806.08624

[12] C. G. Lluís Casals, "The SF12 Well in LoRaWAN: Problem and End-Device-Based Solutions," *Sensors*, vol. 21, no. 19, p. 6478, 2021, doi: https://doi.org/10.3390/s21196478.

[13] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, "Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 796–799, Apr. 2018, doi: 10.1109/LCOMM.2018.2797057.

[14] M. Naeem, M. Albano, D. Magrin, B. Nielsen, and K. Guldstrand, "A Sigfox Module for the Network Simulator 3," *ACM Int. Conf. Proceeding Ser.*, pp. 81–88, Jun. 2022, doi: 10.1145/3532577.3532599;PAGEGROUP:STRING:PUBLICATION.

[15] A. G. Salah Eddine Elgharbi, Mauricio Iturralde, Yohan Dupuis, "Maritime monitoring through LoRaWAN: Resilient decentralised mesh networks for enhanced data transmission," *Comput. Commun.*, vol. 241, p. 108276, 2025, doi: https://doi.org/10.1016/j.comcom.2025.108276.

[16] S. I. A. Elkarim, M. M. Elsherbini, O. Mohammed, W. U. Khan, O. Waqar, and B. M. Elhalawany, "Deep Learning Based Joint Collision Detection and Spreading Factor Allocation in LoRaWAN," *Proc. - 2022 IEEE 42nd Int. Conf. Distrib. Comput. Syst. Work. ICDCSW 2022*, pp. 187–192, 2022, doi: 10.1109/ICDCSW56584.2022.00043.

[17] "System Reference document (SRdoc); Technical characteristics for Low Power Wide Area Networks Chirp Spread Spectrum (LPWAN-CSS) operating in the UHF spectrum below 1 GHz," 2018, [Online]. Available: https://www.etsi.org/deliver/etsi_tr/103500_103599/103526/01.01.01_60/tr_103526v010101p.pdf

[18] J. Weng, R. H. Deng, X. Ding, C. K. Chu, and J. Lai, "Conditional proxy re-encryption secure against chosen-ciphertext attack," *Proc. 4th Int. Symp. ACM Symp. Information, Comput. Commun. Secur. ASIACCS'09*, pp. 322–332, 2009, doi: 10.1145/1533057.1533100.

[19] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *Adv. Cryptol.*, pp. 10–18, 2000, [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-39568-7_2

[20] F. Nisar, M. Ameen, M. Touseef Irshad, H. Hadi, N. Ahmad, and M. Ladan,

"XGBoost-Driven Adaptive Spreading Factor Allocation for Energy-Efficient LoRaWAN Networks," *Front. Commun. Networks*, vol. 6, p. 1665262, doi: 10.3389/FRCMN.2025.1665262.

[21] J. K. L. Yanjiang Yang, "Extended Proxy-Assisted Approach: Achieving Revocable Fine-Grained Encryption of Cloud Data," *Comput. Secur. -- ESORICS 2015*, pp. 146–166, 2015, [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-24177-7_8

[22] A. K. Sultania, C. Delgado, and J. Famaey, "Implementation of NB-IoT Power saving schemes in ns-3," *ACM Int. Conf. Proceeding Ser.*, pp. 5–8, Jun. 2019, doi: 10.1145/3337941.3337944;PAGE:STRING:ARTICLE/CHAPTER.

[23] W. Sun, X. Liu, W. Lou, Y. T. Hou, and H. Li, "Catch you if you lie to me: Efficient verifiable conjunctive keyword search over large dynamic encrypted cloud data," *Proc. - IEEE INFOCOM*, vol. 26, pp. 2110–2118, Aug. 2015, doi: 10.1109/INFOCOM.2015.7218596.

[24] H. L. Peng Zhang, Zehong Chen, Joseph K. Liu, Kaitai Liang, "An efficient access control scheme with outsourcing capability and attribute update for fog computing," *Futur. Gener. Comput. Syst.*, vol. 78, pp. 753–762, 2018, doi: https://doi.org/10.1016/j.future.2016.12.015.

[25] F. Azizi, B. Teymuri, R. Aslani, M. Rasti, J. Tolvaneny, and P. H. J. Nardelli, "MIX-MAB: Reinforcement Learning-based Resource Allocation Algorithm for LoRaWAN," *IEEE Veh. Technol. Conf.*, vol. 2022-June, 2022, doi: 10.1109/VTC2022-SPRING54318.2022.9860807.

[26] Y. Y. and L. W. Q. Huang, "Secure Data Access Control With Ciphertext Update and Computation Outsourcing in Fog Computing for Internet of Things," *IEEE Access*, vol. 5, pp. 12941–12950, 2017, doi: 10.1109/ACCESS.2017.2727054.