

Research article



ADRL: A reconfigurable energy-efficient transmission policy for mobile LoRa devices based on reinforcement learning

Laura Acosta-Garcia ^a, Juan Aznar-Poveda ^b*, Antonio Javier Garcia-Sanchez ^b, Jakob Hollenstein ^c, Joan Garcia-Haro ^a, Thomas Fahringer ^b

^a Department of Information and Communication Technologies, Universidad Politécnica de Cartagena, Plaza del Hospital 1, Cartagena, 30202, Spain

^b Distributed and Parallel Systems group, University of Innsbruck, Technikerstrasse 21a, Innsbruck, 6020, Austria

^c Intelligent and Interactive Systems group, University of Innsbruck, Technikerstrasse 21a, Innsbruck, 6020, Austria

ARTICLE INFO

Keywords:

Internet of Things
Long range
Wireless communications
Transmission parameters
Energy efficiency

ABSTRACT

LoRa is a wireless communication technology for low-power and long-range Internet of Things (IoT) networks. Although it was not initially designed for mobile IoT devices, its applicability has quickly expanded to support mobility. LoRaWAN, its network protocol, includes an Adaptive Data Rate (ADR) mechanism to minimize energy consumption while optimizing data rates and Time-on-Air values. However, ADR struggles to adapt effectively in highly dynamic and unpredictable environments, leading to slow convergence and performance degradation. These challenges are particularly evident in scenarios with frequent changes in signal conditions, such as urban areas with non-line-of-sight (NLoS) or environments with rapid channel variations. To address these limitations, we introduce ADRL, a novel transmission mechanism that proactively selects the most suitable transmission configuration of LoRa devices under dynamic and variable conditions. To this end, the proposed method (i) forecasts the received signal strength of end devices using a lightweight k-nearest neighbors (KNN)-based approximation, and (ii) employs deep reinforcement learning to create a reconfigurable controller on the network server side for selecting the transmission parameters that minimize energy consumption and ensure a given probability of successful packet decoding. Moreover, we leverage duty-cycle information to maximize the delivery ratio and comply with the associated regulations. We compare our method to state-of-the-art works using a realistic simulator in various urban environments. The results show that ADRL effectively balances energy consumption and performance requirements, reducing average energy expenditure by up to 15% and increasing the packet delivery ratio by up to 57%.

1. Introduction

Long-range (LoRa) communications, based on the Long-Range Wide-Area Network (LoRaWAN) protocol, have become a leading technology for large-scale Internet of Things (IoT) and Machine-to-Machine (M2M) applications. LoRa devices offer cost-effective deployment, low energy consumption, and long-range communication, making them suitable for many industrial applications [1,2]. Although LoRa was not initially conceived for mobile devices, its applicability has expanded to various applications with on-board sensors, such as fleet management, remote diagnostics, public transportation optimization, environmental monitoring, and

* Corresponding author.

E-mail address: juan.aznar-poveda@uibk.ac.at (J. Aznar-Poveda).

<https://doi.org/10.1016/j.iot.2025.101577>

Received 31 October 2024; Received in revised form 31 January 2025; Accepted 12 March 2025

Available online 22 March 2025

2542-6605/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

emergency responses during natural disasters [3–6]. In this context, optimizing energy consumption and dynamically adapting to environmental conditions are crucial for deploying and operating IoT systems, whether in battery-constrained or OBD-II (On-Board Diagnostics) interfaced devices [7].

LoRaWAN addresses this issue using the Adaptive Data Rate (ADR) mechanism. ADR enhances energy efficiency by dynamically adjusting transmission parameters such as the spreading factor (SF) and transmission power (p_t) while ensuring successful message reception at gateways. Once established, the link is periodically maintained, reducing the need to downlink messages to the end node. However, the default ADR has shown limited adaptability to changing channel conditions due to its reliance on historical average link quality measurements (i.e., the signal-to-noise ratio (SNR) over multiple transmissions) and its slow decision-making process, [4,8–11]. Specifically, ADR only updates transmission parameters after a fixed number of uplink frames, which increases delays in the response to sudden fluctuations in channel conditions, such as mobility-induced fading, interference, or changes in network topology. Consequently, ADR performance can significantly degrade when dealing with mobile nodes [12], as channel fluctuations greatly affect their connectivity with the gateways. Numerous efforts have been made to improve LoRaWAN's default ADR mechanism's slow adaptation [13–16]. For example, an adaptive model for improving energy efficiency and reliable packet delivery has been proposed in [16]. In [14], an intelligent routing policy for LoRa networks is employed to optimize transmission parameters while diverging from traditional star topology to address performance concerns. However, most of these studies overlook node mobility, significantly limiting their applicability in real-world settings where mobile devices also require efficient transmission configurations.

In this paper, we propose ADRL, a reconfigurable mechanism to optimize energy consumption in mobile devices under highly variable and unpredictable conditions, including non-line-of-sight (NLoS) scenarios. Our approach ensures a user-specific probability of packet decoding while dynamically achieving a custom trade-off between energy consumption and delivery ratio over time. To this end, we leverage device dynamics and environmental information, together with an efficient supervised machine-learning technique, to predict the received signal strength indicator (RSSI) of end devices, using a buffer of previous RSSI samples based on device positions. The problem of selecting the transmission configuration based on RSSI samples is formulated as a Markov Decision Process (MDP), which is solved using Deep Reinforcement Learning (DRL) [17]. This approach employs neural networks to approximate and learn the optimal behavior of LoRa nodes, allowing us to select the most suitable transmission configuration for minimizing energy consumption while satisfying the required probability of successful packet decoding and complying with duty cycle constraints (e.g., 1% in sub-band M (868–868.6 MHz)). The main contributions of this article are summarized as follows:

- We propose ADRL, a novel proactive ADR mechanism for LoRaWAN that selects suitable transmission parameters for mobile LoRa nodes.
- Channel conditions and path loss variations are swiftly captured and predicted using the k-nearest neighbors algorithm on buffered RSSI samples.
- A reconfigurable deep reinforcement learning controller is employed to select the most efficient transmission parameters, guaranteeing a user-defined desired probability of packet decoding and minimizing energy consumption.
- Duty cycle information is tracked and leveraged to reduce energy consumption and determine whether transmission should be carried out.
- The proposed model is evaluated using a realistic and representative simulator under several urban scenarios and compared to relevant related works. The results show that our approach increases the packet delivery ratio by up to 57% while reducing energy consumption by up to 15%.

The rest of this paper is organized as follows. Section 2 describes and discusses related work. Section 3 introduces the basic background about LoRa networks and duty-cycle regulations. The proposed model, including the MDP formulation, analytical development, and training environment, is detailed in Section 4. Section 5 evaluates the model, comparing it with state-of-the-art parameters adaptations in LoRaWAN, and examines the results. Finally, Section 6 summarizes the major findings of this study and outlines future work.

2. Related work

One of the simplest approaches considering node mobility is Blind ADR [18], introduced by the semiconductor manufacturing company Semtech. This mechanism sends frames blindly (hence the name), reducing the application logic complexity of the IoT devices needed to decide when to transmit. Other approaches [10,19] proposed the use of Gaussian filters (G-ADR) and the Exponential Moving Average (EMA-ADR) as low-pass filters to resist rapid changes in the signal-to-noise ratio (SNR). In [11], the authors proposed Resource Management ADR (RM-ADR) for end devices and the network server. The end-device algorithm minimizes retransmission attempts, whereas the network server side leverages the preceding frame's reception power to dictate the subsequent frame's transmission parameters. The algorithm introduced in [20] dynamically adjusts the transmission settings of mobile LoRa nodes to fulfill specific communication requirements related to reliability and energy constraints. Other approaches employed Artificial Intelligence methods to dynamically select the transmission parameters of LoRa devices. For example, the work in [21] applied Deep Reinforcement Learning (DRL) to find the best combination of transmission parameters in terms of energy consumption and an upper bound of the Bit Error Rate (BER). DRL was also used in [15] to minimize energy consumption by dynamically selecting transmission parameters. Despite the benefits of the previous reactive solutions in extending battery life and improving packet delivery rates for mobile IoT devices, highly dynamic and unpredictable environments may still pose significant challenges.

Table 1
Comparison of related works.

Related work	Transmission parameters optimization	Energy reduction	Performance optimization	End-device mobility	Priority management	TDC integration
ADR [26]	✓	✓	✓	✗	✗	✗
BADR [18]	✗	✓	✗	✓	✗	✗
I-ADR [27]	✓	✗	✓	✗	✗	✗
G/EMA-ADR [10]	✓	✓	✓	✓	✗	✗
HADR [19]	✓	✓	✓	✓	✗	✗
LR-ADR [4]	✓	✓	✓	✓	✗	✗
MADERE [22]	✓	✓	✓	✓	✗	✗
VHMM [23]	✓	✓	✓	✓	✗	✗
E-ADR [9]	✓	✓	✗	✓	✗	✗
MDP [24,25]	✓	✗	✓	✗	✓	✗
ADRL (Ours)	✓	✓	✓	✓	✗	✓

In this regard, some approaches determined the transmission parameters based on the predicted performance and position of mobile devices [4,22]. In [4], the LR-ADR and LR+ADR methods were introduced. LR-ADR focus on stabilizing the received SNR of packets and predicting it for subsequent transmissions. This method assumes fixed periodic transmission patterns from end devices, supporting the SNR estimation by incorporating a constant margin to simulate device movement away from the gateway. However, the proposed linear prediction approach may not effectively adapt to non-linear changes in the communication environment, such as signal and noise variations due to obstacles or atmospheric conditions. LR+ADR operates at the end-device level to quickly restore connectivity in case of failure. The authors of [22] recently introduced MADERE (Mobile Adaptive Datarate for LoRaWAN), an ADR that estimates the location of end devices by computing the probability of their position at various points in space. A discretized version employing a grid of points is used to represent the LoRa network area. This approach enables location-based adjustments within LoRaWAN networks but relies heavily on the propagation model assumed to estimate the distance between an end device and the gateway and does not provide a generic position estimation in terms of end-device speed. E-ADR is an Enhanced Adaptive Data Rate [9] designed for scenarios with predefined mobility patterns. This method was extended [23] to adapt to IoT applications with unpredictable trajectories, such as smart cattle ranching, achieving a solution that adapts transmission parameters based on an estimated position, node-to-gateway distance, and estimated RSSI, enhancing its suitability for multiple environmental conditions. While they address mobility and enhance the performance of the default ADR mechanism, the approaches mentioned above overlook duty cycle regulations for selecting transmission parameters. This might increase energy consumption and cause harmful interference to other devices operating in the same frequency band.

As shown in Table 1, which summarizes the related works considered so far, only a few works took into account the duty cycle limitation [24,25]. In [24], the authors formulated a Markov Decision Process (MDP) including Time Duty Cycle (TDC) information to determine a transmission policy of IoT devices that maximizes the number of reported events using a given priority. Similarly, the study introduced in [25] enhanced node performance based on power and spectrum efficiency. It also formulated optimal transmission policies as constrained optimization problems to maximize throughput within given energy consumption limits. However, these papers did not directly employ the duty-cycle information for optimizing transmission parameters and disregarded the mobility of end nodes. To fill this research gap, our proposal integrates duty-cycle information into the selection of transmission parameters based on the RSSI prediction, optimizing the performance and energy efficiency of the LoRa network.

3. Background

3.1. LoRaWAN adaptive data rate

LoRaWAN operates on a star network topology consisting of end-devices (EDs) communicating with gateways (GWs). The LoRaWAN gateways transform LoRaWAN packets into IP packets for secure transmission to the Network Server (NS), which is responsible for managing the ADR mechanism. When the NS processes a new packet, it identifies the GW that recorded the highest SNR among the received signals. This particular GW is then used to transmit an acknowledgment message. Minimal metadata from the IP packets, such as reception time and SNR, are then used by the NS to select the optimal transmission parameters. The SNR values are stored in a queue associated with the specific ED or LoRa node. The NS periodically checks this queue and computes the SNR margin (SNR_{margin}) for a particular node using Eq. (1), as follows:

$$SNR_{margin} = SNR_{max} - SNR_{req} - SNR_{th} \quad (1)$$

where SNR_{max} is the maximum SNR value of the last 20 frames received, SNR_{req} represents the SNR value of the signal requested, and SNR_{th} represents a threshold constant, which is set by default to 10 dBm. Then, the N_{step} variable is calculated as $N_{step} = \left\lfloor \frac{SNR_{margin}}{3} \right\rfloor$. N_{step} is used to adjust the SF and p_i of the ED to minimize energy consumption and enhance network capacity. If N_{step} is positive, the NS reduces the SF by N_{step} . If the SF reaches its minimum value, the NS decreases the p_i in 3 dBm steps, provided that it is not yet at the minimum level. Conversely, when N_{step} is negative, the NS increases the p_i by steps of 3

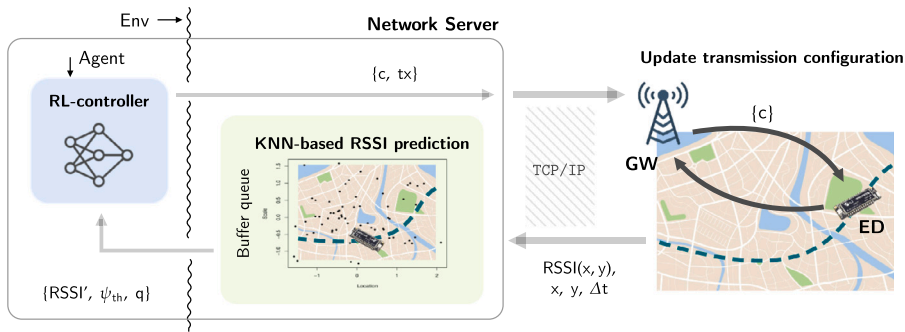


Fig. 1. Diagram of the proposed transmission parameter selection mechanism (ADRL) for LoRa-based mobile devices using trajectory-based RSSI estimation and an RL controller in the NS.

dBm, unless it has already reached the maximum value. If the maximum transmission power is attained, the NS increases the SF. A TX_CONFIG command is dispatched to the ED, indicating the new SF and p_t parameters for the subsequent uplink message. There are two major recommendations for updating the transmission parameters: (i) when the ADRAckReq bit is set by an ED (e.g., every time the RSSI drops), and (ii), periodically, when every N uplink frames [8,28–30].

3.2. Duty cycle constraint

The duty cycle constraint is a fundamental regulatory requirement in LoRaWAN networks, governing the permissible transmission time of an ED over a specific frequency band within a defined period [26]. This constraint is vital to ensure fair spectrum use and minimize interference with other devices sharing the same frequency band. In LoRaWAN systems, the duty cycle is typically expressed as the fraction of time an ED can transmit relative to a specific period. The European regulation defined by the ETSI EN300.220-2 standard imposes duty cycle limitations that can vary from 0.1% to 10% across different sub-bands. For instance, in sub-band M (868–868.6 MHz), a duty cycle limitation of 1% is imposed. This means that an ED operating in this frequency band can transmit for no more than 1% of a defined time interval (e.g., 36 s per hour). Compliance with duty cycle restrictions is critical for maintaining network integrity.

4. System model

In this paper, we propose ADRL, a novel transmission mechanism that proactively considers signal degradation and node mobility [9–11] to select optimal transmission parameters in LoRa networks. Unlike the traditional ADR mechanism, which struggles to adapt in highly dynamic environments due to its reliance on collecting SNR values and long convergence times (causing suboptimal transmission parameter adjustments that lead to increased packet loss and inefficient energy use [8,9]), ADRL is designed to address these limitations. This is particularly true in urban and semi-urban environments with mobile end devices and highly variant and unpredictable communication channels, including NLoS conditions. The main goal of this paper is to provide a reconfigurable method that reduces energy consumption while ensuring a given probability of packet decoding, leveraging information about the remaining TDC available. Considering the duty cycle information allows us to improve energy efficiency by selecting transmission configurations with less Time-on-Air (ToA) that still fulfill performance requirements. A diagram of the proposed model is shown in Fig. 1. First, we estimate the trajectory of the end devices to gain insight into potential obstacles and signal variations using RSSI values. This is essential for optimizing energy consumption and maintaining reliable communication. Then, we employ Deep Reinforcement Learning (DRL) to create a controller that dynamically adjusts the transmission parameters for each transmission from the end device to the gateway, enhancing the adaptability and robustness of the system in varying conditions.

4.1. KNN-based RSSI prediction

As in the traditional ADR, we also use a buffer in the NS to periodically store some metadata of previously received messages from the EDs. However, instead of only using them from the past to determine the next parameters, we introduce a novel approach by dynamically estimating the next device position and predicting the associated RSSI for this position. Rather than relying only on past data, predicting the next position and the associated RSSI enables the proactive adaptation of transmission parameters. Therefore, we anticipate changes in signal quality due to mobility and environmental variations, leading to more accurate and efficient adaptations. By better understanding the expected RSSI within the area where the ED is located, the system can optimize energy use by selecting the most appropriate transmission parameters. This ensures that the ED transmits with the minimum necessary power while maintaining reliable communication. We consider GPS-plus-LoRaWAN devices that can regularly update their geographic coordinates [31–33]. The NS buffer receives the ED locations over time and stores the associated RSSI values in a circular buffer ($\mathcal{B}$). This data structure stores the most recent data while replacing the oldest entries, facilitating continuous monitoring with a reduced memory footprint.

4.1.1. Trajectory estimation

In this study, we estimate the next position of mobile EDs by dynamically calculating the radius of the curvature of their trajectory [22,34,35]. Estimating the future position of EDs is crucial for predicting the RSSI at that point, which enables the system to anticipate obstacles and variations in signal quality. With this information, we can make more informed decisions about the transmission parameters, ensuring both energy efficiency and communication reliability in dynamic and unpredictable environments. For this purpose, we record each ED's last three GPS positions, namely A (oldest), B , and C (newest). The radius of the curvature is calculated using the perpendicular bisectors of the chords formed by these positions [36]. First, we calculate $\overline{AB}$ and $\overline{BC}$ as the chords describing the vehicle's trajectory and determine their midpoints M^{AB} and M^{BC} . Next, we compute the gradients of these chords, $\nabla(\overline{AB})$ and $\nabla(\overline{BC})$, to understand their directional changes. By deriving perpendicular bisectors $\overline{L_{AB}}$ and $\overline{L_{BC}}$, we use their gradients and midpoints to formulate the equations of the bisectors:

$$O_y - M_y^{AB} = \nabla(\overline{L_{AB}})(O_x - M_x^{AB}) \quad (2a)$$

$$O_y - M_y^{BC} = \nabla(\overline{L_{BC}})(O_x - M_x^{BC}) \quad (2b)$$

Solving these equations gives us the central point, denoted as O . The radius R , representing the distance between a position point and the estimated center, is calculated as $R = \sqrt{(O_x - A_x)^2 + (O_y - A_y)^2}$. To predict the next position along the vehicle's trajectory, we calculate velocity v , and then we compute displacement angle $\theta = \frac{v \times \Delta t}{R}$ in radians, where Δt represents the time between consecutive transmissions, calculated using a simple moving average of the last three transmission times. Finally, we predict the coordinates of the next position based on the current estimated center.

4.1.2. RSSI prediction

Once the next position (x, y) has been estimated, we predict the RSSI at this position employing the k -nearest neighbors (KNN) algorithm. KNN is a non-parametric supervised learning classifier widely used across various domains, including localization applications with LoRa technology for indoor and outdoor scenarios [37]. KNN is particularly suited for regression tasks, where it predicts values for new data points by identifying the k -closest points in the training dataset. In our application, we formulate the RSSI prediction as a regression problem leveraging the KNN algorithm. We use it to predict RSSI values based on existing buffered RSSI measurements collected on the NS from nearby devices or vehicle locations. The predicted RSSI at a given location and transmission power is estimated by averaging the RSSI values of its closest neighboring points so that the network can proactively adjust transmission parameters to optimize communication. For example, suppose the predicted RSSI indicates a potential drop in signal strength due to an upcoming obstacle. In this case, the network can increase the transmission parameters to ensure reliable communication. Conversely, if the predicted RSSI suggests strong signal strength, the network can change the communication modulation to save energy. This approach is mathematically described as:

$$RSSI'(x, y) = \frac{1}{k} \sum_{i=1}^k RSSI(x_i, y_i) \quad (3)$$

where $RSSI'(x, y)$ is the predicted RSSI at position (x, y) , and k is the number of nearest neighboring samples considered for prediction. The k -nearest positions and their associated RSSI values allow us to compute the estimated RSSI, as specified by Eq. (3). The output of the KNN algorithm enables the effective prediction of the RSSI in the next location, considering both vehicle or node dynamics and the expected RSSI variations influenced by environmental factors and channel conditions.

4.2. RL controller

After predicting the future RSSI value, we determine whether the LoRa nodes within the coverage area of the gateways should initiate a transmission and, if so, select the most appropriate transmission parameters in terms of energy and communication performance. Our proposal aims to reduce energy consumption while satisfying a desired decoding probability, specified by the user or application layer. As introduced in Section 3, we also leverage the TDC information to reduce energy consumption and guarantee that the transmission adheres to duty cycle constraints. We formulate this problem as a Markov Decision Process (MDP) due to its suitability in uncertain scenarios and dynamic environments where prior knowledge is unavailable [38]. In this framework, an agent interacts with its surrounding environment, perceiving observations and taking actions to achieve certain goals. The agent perceives the environment as a set of states, and the agent's actions influence the transition between these states. To guide the agent to the desired behavior, the agent receives rewards from the environment based on its actions. In our case, the environment is a LoRa network, including all the components of the LoRa network, and communication effects, where the agent is a virtual ED entity represented on the NS side that is optimizing transmission parameters. When the agent associated with a particular ED takes an action, the NS sends a downlink message to that ED, updating its transmission parameters. Further details about the environment used and the ADRL algorithm can be found in Sections 4.4 and 4.5, respectively.

4.2.1. States and actions

The state space of our model includes (i) the predicted signal strength indicator $RSSI'$ obtained by Eq. (3); (ii) the desired decoding probability specified by the user or application ψ_{th} at a given time; and (iii) the remaining TDC (q) in seconds relative to the maximum allowed (Q_{max}), defined as follows:

$$s = (RSSI'(x, y), \psi_{th}, q) \quad (4)$$

The probability threshold provides a reconfigurable method to dynamically control the desired communication quality over time, directly impacting energy consumption and the packet delivery ratio. For the duty cycle calculations, we assume Class A devices [26], which transmit whenever an event arrives, while Q_{max} is reset at regular intervals T . The duty cycle is assumed to be subject to European regulation, which is limited to 1%. For example, if total transmission period T is 1 h, $Q_{max} = 36$ s. Regarding the RSSI, it influences transmission decisions to maintain adequate link quality, reflecting communication performance. We assume LoRa-plus-GPS devices, and therefore, both the gateway and network server (where the model is trained) know the position of end-devices or IoT nodes. By providing the RSSI in the state and excluding the current position, (i) the learning problem is simplified into a fully observable problem, and (ii) the controller is more generally applicable since the learning cannot overfit the RSSI prediction to a given training scenario that directly depends on GPS coordinates. With the state information, the NS determines whether to initiate a transmission on the ED and, if so, which transmission parameters to employ. We define action a of the proposed MDP as follows:

$$a = (c, tx) \quad (5)$$

where $c \in C$ represents one of the available transmission configurations stated in the LoRaWAN standard [39]. A transmission configuration is comprised of the Spreading Factor (SF), Coding Rate (CR), Data Rate (DR), Sensitivity of receptor (S), and Bandwidth (BW), and $tx \in \{0, 1\}$ is a binary variable that controls the packet transmission on the ED. If $tx = 1$, the ED will transmit, otherwise, the packet (p) will be discarded.

4.2.2. Reward function

The reward function plays a fundamental role by providing a feedback signal that guides the agent toward actions that maximize performance objectives. To effectively capture the decision-making process of our particular problem, we define the following reward function:

$$r(\psi, \psi_{th}, Q_{MAX}, q) = \pi_P f(\psi, \psi_{th}) + \pi_T g(Q_{MAX}, q) \quad (6)$$

As can be observed, the proposed reward function is based on two distinct functions: function $f(\cdot)$ has been designed to ensure performance compliance regarding signal quality while minimizing power consumption as much as possible, whereas function $g(\cdot)$ is associated with the TDC and transmission constraints.

$$f(\psi, \psi_{th}) = \begin{cases} -1 & \text{if } \psi < \psi_{th} \\ 1 - |\psi - \psi_{th}| & \text{otherwise} \end{cases} \quad (7)$$

As shown in Eq. (7) and Fig. 2, when the actual decoding probability ψ for the chosen transmission configuration is less than the established minimum decoding probability threshold ψ_{th} , the reward function returns -1. This reflects the critical importance of meeting the minimum decoding probability requirement to ensure a reliable signal reception. In contrast, when ψ is greater than or equal to the minimum threshold chosen by the application or user ψ_{th} , the reward function is modeled as a decreasing function based on the difference between the estimated decoding probability and the chosen threshold. This function returns the maximum value for configurations with ψ closer to ψ_{th} and decreases as the difference between these variables increases, encouraging the selection of parameters that maintain high signal quality while minimizing energy consumption and enhancing the overall performance of the LoRa communication system. There is a trade-off between decoding probability and energy consumption for different transmission configurations, mainly caused by spreading factor variations. The higher the SF, the longer the transmission ranges with successful decoding, at the cost of increased energy consumption. However, this is not entirely predictable, as it can be significantly affected by obstacles and environmental conditions that alter the path loss. This trade-off can be controlled dynamically by using the ψ_{th} parameter as state input. Function $g(Q_{MAX}, q)$ of the reward is associated with TDC compliance, and it is calculated as follows:

$$g(Q_{MAX}, q) = \text{sign}(Q_{MAX} - q) \quad (8)$$

where q is the consumed ToA once action a is applied, and Q_{MAX} is the maximum ToA available for a given transmission period T , assuming a TDC limitation of 1%. This function is based on a $\text{sign}(\cdot)$ function, evaluating the difference between the consumed ToA and the maximum ToA available. If the difference is greater than or equal to 0, sufficient TDC is available to transmit using the chosen transmission parameters, and the function returns +1, and -1 otherwise. This prioritizes actions with lower energy consumption that can be completed within the remaining time, contributing to effective TDC management. Weights π_P and π_T in reward Eq. (6) determine the relative importance of each term in the reward function. Weight π_P emphasizes the priority of achieving performance objectives related to signal quality and decoding probability, while π_T underscores the significance of efficient duty cycle use and regulatory compliance. In this paper, we have assigned the values of $\pi_P = 2$ and $\pi_T = 1$, highlighting our priority of optimizing signal quality to ensure reliable communication while minimizing energy consumption and maintaining a balance with duty cycle efficiency. The practical selection of these weights requires careful tuning based on the specific objectives of the system.

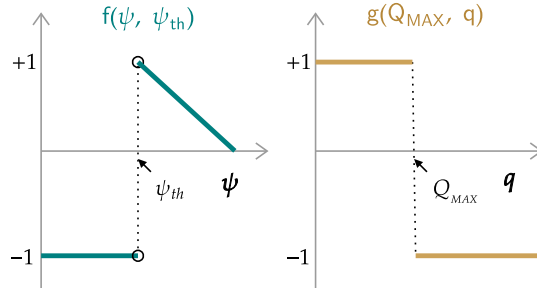


Fig. 2. Illustration of the functions $f(\psi, \psi_{th})$ and $g(Q_{MAX}, q)$, which compose the reward function $r(\psi, \psi_{th}, Q_{MAX}, q)$.

4.3. Analytical simulation model

ADRL has been built atop a custom simulation framework based on analytical models of LoRa networks formulated in the literature [21,40–43]. We consider a set of EDs, GWs, and an NS that exchange uplink and downlink messages upon the availability of sensing data. Since ADRL focuses on optimizing the transmissions from the EDs to the GWs, we assume that the probability of a downlink message being decoded is similar to its predecessor uplink message [44]. In other words, we assume that channel conditions do not vary within the time window between uplink messages and their subsequent response downlink messages. Reconnection or retransmission mechanisms will be considered in future works. Regarding gateways, they are modeled to operate in semi-duplex mode [45–47]. In Sections 4.3.1 and 4.3.2, we provide a detailed discussion of the transmission and energy models employed in this paper.

4.3.1. Transmission reliability and performance model

To determine whether a packet is successfully decoded or not [4,16,48–50], we employ the received signal strength indicator (RSSI). The RSSI represents the power level that a receiver perceives from the transmitted signal. Analytically, the RSSI (dBm) in LoRaWAN networks can be estimated as follows:

$$RSSI = p_t - PL(d_0) - 10\gamma \log_{10} \left(\frac{d}{d_0} \right) - N_\sigma \quad (9)$$

where p_t represents the transmission power of the ED, $PL(d_0)$ is the path loss at reference distance d_0 , γ is the path loss exponent characterizing signal attenuation with distance, d is the distance between the ED and GW, and N_σ denotes the noise component, typically modeled as a zero-mean Gaussian random variable with a standard deviation σ . This expression can be simplified as $RSSI = z - N_\sigma$, where $z = p_t - PL(d_0) - 10\gamma \log_{10} \left(\frac{d}{d_0} \right)$ and $N_\sigma = N(0, \sigma^2)$. Although this model primarily formulates the path loss as a function of the distance, this might not be true in urban scenarios with NLoS. For this reason, in the evaluation Section 5, we consider a wide range of random noise. This reflects the environmental variability beyond the distance between end nodes and gateways, ensuring a more realistic representation of signal propagation conditions. From the RSSI, we can easily derive the signal-to-noise ratio as $SNR(dB) = RSSI - P_{noise}$, where P_{noise} denotes the noise floor in dBm. The SNR represents the quality of the received signal and can be used to derive the Bit Error Rate (BER). The BER also quantifies the rate of errors in the received data. Assuming Rayleigh fading channels, it can be estimated using a closed-form approximation [40,42]. Furthermore, we can obtain the probability of successful decoding (ψ), which refers to the likelihood that the received signal strength (RSSI) at the GW exceeds the receiver sensitivity threshold (η) for a given SF. The formula to calculate ψ accounts for the inherent variability in received signal strength due to channel fading and path loss:

$$\psi = \mathbb{P}(N \leq RSSI - \eta) \approx \frac{1}{2} + \frac{1}{2} \operatorname{erf} \left(\frac{z - \eta}{\sqrt{2}\sigma} \right) \quad (10)$$

where η is the specific GW receiver sensitivity for a given SF, and $\operatorname{erf}(\cdot)$ is the error function used to compute the integral of the Gaussian distribution. Then, we can obtain the Packet Delivery Ratio (PDR) for the transmission link as $PDR = \psi \cdot \zeta$, where ζ is a factor representing the probability that the packet is not corrupted due to co-channel interference.

4.3.2. Energy model

To characterize the energy consumed when transmitting a packet (p) from the ED, we employ the total time it takes to transmit a packet from the ED to the GW, usually referred to as Time-on-Air (ToA). This time includes the transmission of preamble symbols (N_{PR}) and payload symbols (N_{PL}) within the data frame, as follows:

$$ToA = T_s \cdot (N_{PR} + N_{PL} + 4.25) \quad (11)$$

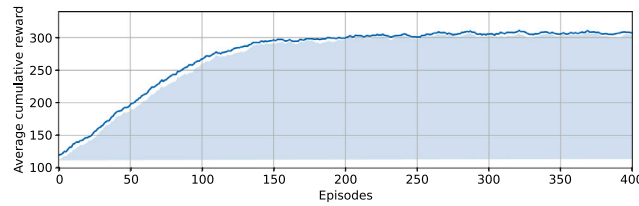


Fig. 3. Average reward over training episodes across 10 different seeds. The shadow area indicates the 95% bootstrapped confidence intervals.

where $T_s = \frac{2^{SF}}{BW}$ is the symbol period, determined by the SF and the BW , N_{PR} is the number of preamble symbols (typically 8 symbols for the EU868 band), and N_{PL} is the number of symbols transmitted in the frame payload, estimated using the following equation:

$$N_{PL} = 8 + \max\left(\left\lceil \frac{8PL - 4SF + 28 + 16CRC - 20H}{SF - 2DE} \right\rceil \cdot \left(\frac{CR + 4}{4}\right), 0\right) \quad (12)$$

where PL represents the length of the payload in bytes, SF is the spreading factor used in the transmission, CRC is the Cyclic Redundancy Check used for error detection, H denotes the presence of a physical header (0 or 1), DE indicates the implementation of low data rate optimization (0 or 1), and CR is the coding rate. With Eqs. (11) and (12), we can estimate the energy consumption (E) associated with a specific transmission power (p_t) and the calculated ToA [51]:

$$E = p_t \cdot ToA \quad (13)$$

As shown in Eq. (12), energy consumption is influenced by several factors, including transmission power, the modulation scheme, and data rate. Optimizing ToA is essential for maximizing the battery life of wireless end-devices, particularly in LoRa-based IoT nodes powered by batteries. In this paper, we calculate the energy consumption using uplink messages [11,19,52]. To optimize transmission parameters, we consider downlink messages with low payloads, thus making the energy consumption of the reception mode of the EDs almost negligible. It should be highlighted that this is only valid for reduced downlink messages. Note that the frequent updates of transmission parameters carried out by ADRL would be counterproductive with large downlinks in terms of energy consumption. In such a case, we would need to restrict the number of downlink transmissions to optimize energy expenditure. In future work, we will explore this aspect and further optimize the usage of downlink messages.

4.4. Training environment

For training the RL controller, which is critical for optimizing the transmission parameters of end devices in ADRL, we developed a custom Gymnasium environment [53] which interacts with the simulation framework introduced in Section 4.3. We employ the KNN-based model and the RSSI buffer ($\mathcal{B}$) to predict the RSSI values of mobile EDs based on historical measurements, capturing the dynamic behavior of the wireless environment, including mobility and environmental changes that affect signal quality and reliability. The predicted RSSI is then fed into the RL agent, which uses this information to optimize the transmission parameters.

We derive suitable transmission policies using the Proximal Policy Optimization (PPO) [54] algorithm. PPO is an on-policy gradient algorithm that iteratively updates a parameterized policy to maximize the expected cumulative reward. It addresses the issue of unstable policy updates, common in many RL algorithms, by constraining changes to the policy distribution during each iteration, which ensures a proximal policy update. PPO has demonstrated effectiveness across various RL domains, offering stability, improved sample efficiency, and straightforward implementation compared to other policy optimization algorithms. The main parameters used during the training are plotted in Table 2. The training has been carried out 10 times with different seeds to prove the consistency of the results, as seen in Fig. 3. The following sections provide further details about the analytical model used in our training environment [55] and about the resulting ADRL transmission mechanism employing the trained model.

4.5. ADRL transmission mechanism

This section describes the proposed ADRL algorithm for the efficient selection of transmission parameters in dynamic LoRaWAN environments. ADRL is executed on the ED (Algorithm 1) and the NS (Algorithm 2) to ensure effective communication and adjustment to changing channel conditions. The ED side runs continuously and manages sensor data collection and downlink commands from the NS. At the start of each period T , the ED resets its currently consumed ToA q to zero to adhere to the duty cycle limitations imposed by the network (line 3). Since Class A EDs are considered, transmissions from the ED to the NS via the GW are prepared when the ED receives data from connected sensors. The ED reads its current GPS coordinates (x, y) (line 8) and generates a new uplink packet p_u . This packet includes the device's location, the sensor data, the consumed ToA (q), and the ED's unique identifier. Once the uplink packet is created, the ED transmits it to the NS. After the transmission, q is updated to account for the time spent transmitting (line 12).

On the NS side, the algorithm processes incoming packet p_u from the ED and determines suitable transmission parameters for future communications. When the NS receives an uplink packet from an ED, it records the ED's location and the measured RSSI in a

Table 2
Parameters used during the training phase based on the European LoRaWAN specifications [56].

Parameter	Value
Header (H)	0, 1
Payload (PL)	26 Bytes
Bandwidth (BW)	125 kHz
Coding Rate (CR)	4/5
Noise figure (NF)	6 dB
LoRa frequency (f)	868 kHz
Spreading factor (SF)	7, 8, 9, 10, 11, 12
Transmission power (p_t)	14 dBm
Cyclic Redundancy Check (CRC)	1
Low data rate optimization (DE)	0, 1
Path loss exponent (γ)	2
Reference distance (d_0)	1 m
Standard deviation of the Path Loss (σ)	2 dB
Path loss at reference distance ($PL(d_0)$)	70 dB
Total transmission period of the ED (T)	1 h
Time Duty Cycle (TDC)	1%
RSSI range for training	-140 – 0 dBm
ψ_{th} range for training	0.3 – 1
Weight π_p	2
Weight π_r	1
TDC range for training	-6 – 36 s
Learning rate (α)	0.01
Discount factor (γ)	0.9
Number of iterations	7×10^6

Algorithm 1 ADRL (ED side)

```

1: loop
2:   if  $t \% T == 0$  then
3:      $q = 0$ 
4:      $allowed\_tx = true$ 
5:   end if
6:   if sensor data  $d$  available then
7:     if  $allowed\_tx$  or  $q < Q_{MAX}$  then
8:        $(x, y) \leftarrow$  Read GPS coordinates
9:       Create new uplink packet  $p_u$ 
10:       $p_u.append(x, y, d, q, id)$ 
11:      Send uplink packet  $p_u$ 
12:       $q \leftarrow q + ToA(c)$ 
13:    end if
14:  end if
15:  if downlink packet  $p_d$  received then
16:    if  $p_d.id = local\_id$  then
17:      if  $p_d.tx = 1$  then
18:         $allowed\_tx = true$ 
19:        Apply  $p_d.c$ 
20:      else
21:         $allowed\_tx = false$ 
22:      end if
23:    end if
24:  end if
25: end loop
26:

```

circular or ring buffer (B) of size (n). This buffer is updated every time a new packet is successfully transmitted and acknowledged by the NS, ensuring that the most recent data is available for future predictions. If the ED's position already exists in the buffer, the NS replaces the previous entry with the new RSSI measurement, maintaining only the latest value for that location. If the position does not exist, the NS pushes the new data into the buffer, discarding older data. As input, the ADRL algorithm also requires a user-defined decoding probability threshold ψ_{th} , which can be dynamically updated every time ADRL is executed on the NS and the number of nearest neighbors (k). Then, we predict the next transmission time using a simple moving average of the last three

Algorithm 2 ADRL (NS side)

```

1: Input:  $\psi_{th}, k$ 
2: loop
3:   if uplink packet  $p_u$  received then
4:      $\mathcal{B}.\text{update}(p_u.x, p_u.y, RSSI(p_u.x, p_u.y))$ 
5:     Estimate time of next transmission  $t$ 
6:     Estimate next position  $(x, y)$  for ED  $p_u.id$ 
7:      $RSSI'(x, y) = \frac{1}{k} \sum_{i=1}^k RSSI_i(x_i, y_i)$  ▷ Use RSSI samples from buffer  $\mathcal{B}$ 
8:      $s \leftarrow (RSSI'(x, y), \psi_{th}, p_u.q)$ 
9:      $(c, tx) \leftarrow \theta(s)$ 
10:    Create new downlink packet  $p_d$ 
11:     $p_d.\text{append}(c, tx)$ 
12:    Send  $p_d$  to ED  $p_u.id$ 
13:  end if
14: end loop
15:

```

transmission times. This approach helps to handle the irregular transmission patterns of Class A devices. It should be noted that even though the time of the next transmission is important for estimating the position, the number of RSSI samples available in the NS is the factor that introduces the largest error in the RSSI prediction. This time is used to calculate the next position of the ED (line 6) and predict the RSSI associated with that position using the nearest- k RSSI samples and the KNN model trained with historical data (line 7). The KNN algorithm has a linear computational complexity $\mathcal{O}(n)$, where n is the number of previously recorded positions and their associated RSSI values stored in the NS ring buffer. The predicted RSSI, user-defined decoding probability threshold ψ_{th} , and q constitute the state s (line 8). Forecasting the RSSI is helpful in areas with high fading or NLoS, where the device might save a significant amount of energy by avoiding transmissions that will likely not reach any gateway or guarantee a higher packet delivery by increasing transmission robustness. The specified ψ_{th} parameter directly induces this trade-off. This state information is used to predict the best action (i.e., transmission configuration c and transmission variable tx) by the trained model (θ). The transmission parameters c are included in downlink packet p_d , which is then sent back to the ED $p_u.id$. This downlink packet keeps a low payload by only containing the selected configuration for the next transmission from the ED, enabling the dynamic adaptation of the ED based on the current channel conditions and duty-cycle limitations.

5. Performance evaluation

In this section, we evaluate the following aspects of ADRL:

- **Packet decoding requirements:** ADRL enables users to dynamically specify the desired packet decoding probability (ψ_{th}), allowing them to reach a custom trade-off between energy consumption and the packet delivery ratio. To assess this contribution, we evaluate the performance of ADRL under a range of different required ψ_{th} values.
- **Mobility and adverse channel conditions:** The RSSI fluctuations and adverse channel conditions partially derived from ED mobility are captured using buffered RSSI samples and predicted with KNN. We emulate heterogeneous channel conditions by splitting the evaluation area where the ED moves along into clusters with different RSSI levels, which are explained in detail in Section 5.2. Then, we evaluate the trajectory estimation error of ADRL.
- **Transmission configuration update frequency:** We also evaluate how varying the number of received frames required before updating transmission parameters influences ADRL performance and energy efficiency, comparing these results with the default ADR mechanism.

The evaluation was carried out using a custom simulation environment, wrapped as a Gymnasium environment, as defined in Section 4.4. We employ the well-known Semtech SX1276 transceivers [57] and 3.6 V batteries and a packet payload of 26 Bytes. Note that the trained models would still apply to any device and battery compliant with the European standard [26] since commercial devices support the variables included in the proposed observation space. In this evaluation, we implement a simulation environment consisting of a 5×5 km urban area and a *MultiTech Conduit IP67* LoRa gateway located at the center. We consider moving EDs along random trajectories within this area at speeds of between 25 and 55 km/h. The ED transmits packets with a 26-byte payload following an exponential distribution with an average period of 100 s. We employ 32 EDs, each moving along individual trajectories, to cover a wide range of RSSI values throughout the simulated area where the EDs are moving along. We compare ADRL with state-of-the-art mechanisms, including the default ADR mechanism of LoRaWAN [26], Blind ADR (BADR), and G-ADR [10]. Semtech developed blind ADR to economize on battery life without relying on continuous feedback. The transmission parameters are predefined and applied in a round-robin way [18]. G-ADR [10] adjusts transmission parameters by predicting the SNR using a Gaussian filter, enhancing the SNR estimation based on its Gaussian distribution and enabling efficient adaptation to dynamic environments. The metrics employed in this evaluation include energy consumption and the packet delivery ratio.

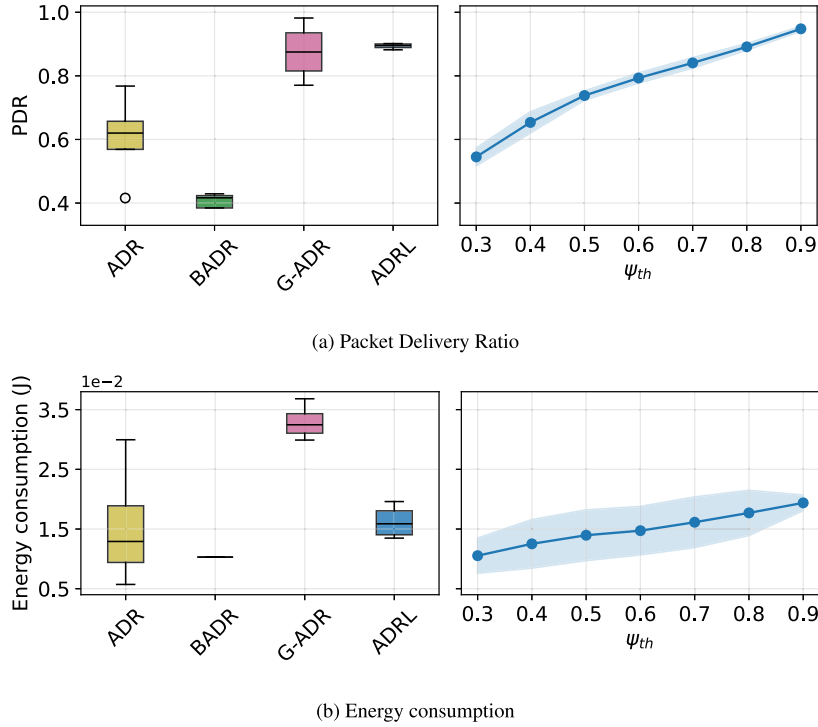


Fig. 4. Analysis of mean PDR and mean Energy consumption per packet for different baselines using $\psi_{th} = 0.8$, $k = 3$, and $n = 1000$. ADRL demonstrates superior performance, achieving, on average, the highest PDR with the lowest mean energy consumption per packet compared to the other algorithms.

5.1. Communication performance with varying requirements

In this first scenario, we evaluate the performance of ADRL in an urban environment under different required probabilities of successful packet decoding (ψ_{th}). As shown in Fig. 4(a) (right), ADRL provides a reconfigurable and flexible way of adjusting transmission parameters based on the required probability of successful packet decoding. As this probability threshold increases, the PDR also increases, demonstrating that ADRL effectively meets higher performance demands. The PDR consistently exceeds the required probability, reflecting the robustness of our model in surpassing the performance targets set. Meanwhile, as seen in Fig. 4(b) (right), energy consumption increases with higher probabilities due to the necessity of using different transmission parameters to maintain high performance. Energy consumption peaks for a decoding probability of 0.9. Despite this increase, the system's energy consumption remains lower than that of ADR across all the scenarios, highlighting its efficiency in optimizing energy use while meeting performance requirements. Specifically, in Fig. 4(b) (left), we set the probability of successful packet decoding to a fixed value of ϕ to compare it with previous approaches. The default ADR exhibits intermediate to low performance, achieving a mean PDR approximately 32% lower than our ADRL and with considerable variability, indicating that performance can fluctuate significantly.

Regarding energy consumption per packet (left side of Fig. 4(b)), ADRL shows slightly higher median energy consumption but more consistent results than ADR, which exhibits substantial variations and reflects ADR's inefficient energy use. In contrast, BADR shows very low energy consumption, saving, on average, approximately half the energy per packet compared to ADRL, but with the poorest performance in terms of packet delivery, achieving, on average, a 57% lower PDR than ADRL. This is because BADR allocates transmission configurations “blindly”, without considering specific transmission requirements, as can be seen in Fig. 5, showing the SF because it is the most significant transmission parameter. Regarding G-ADR, it attains a relatively high PDR, around 3.5% lower, on average, than ADRL, but it also shows significant variability, suggesting that its performance is effective only for specific cases. Additionally, G-ADR's energy consumption per packet is $1.95\times$ higher than ADRL's, which is a substantial increase. This is because of its predominant use of SF12, as detailed in Fig. 5, indicating the algorithm's lack of energy efficiency in its transmission parameter selection. The ADRL algorithm stands out with a mean PDR of 0.90 and low data variance, showing high and consistent packet delivery performance. Energy consumption is relatively low, with a mean value of 0.017 J per packet. However, this average increase in energy consumption is due to the need to select higher spreading factor SFs depending on node positions and environmental conditions. Fig. 5 illustrates how ADRL effectively uses all the SFs according to the conditions, achieving a strong PDR while maintaining a balance with energy efficiency. This contrasts with the ADR mechanism's behavior, as shown in the same figure. While it also employs a range of SFs, its selection appears less optimal, as evidenced by the lower PDR values achieved.

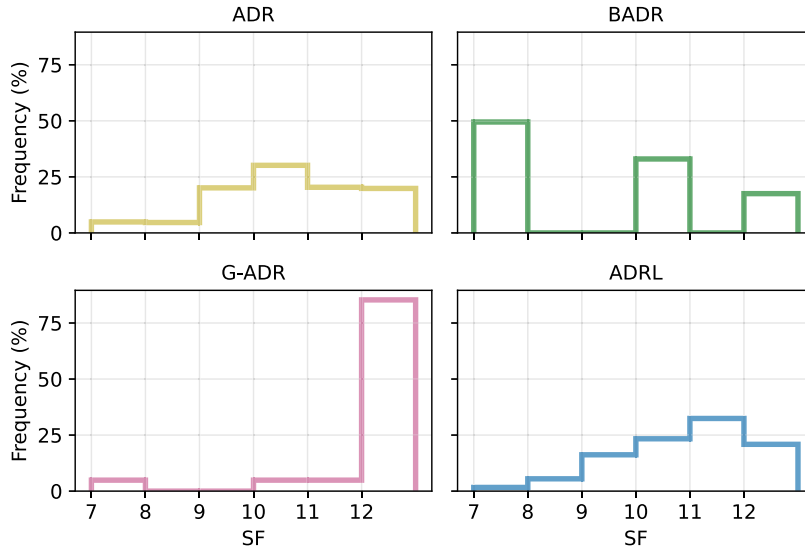


Fig. 5. Frequency distribution of the spreading factors used by each method.

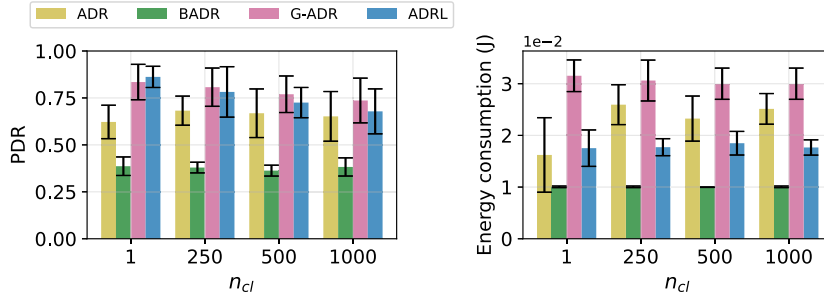


Fig. 6. Evaluation of the PDR and Energy consumption, on average, per packet for different baselines as a function of the number of clusters (n_{cl}), using $\psi_{th} = 0.8$, $k = 3$, and $n = 1000$.

5.2. Impact of adverse channel conditions with moving EDs

In this scenario, we aim to assess the robustness of different algorithms under the varying channel conditions that are typical of urban environments, influenced by obstacles, buildings, and other sources of interference while the ED is moving. To simulate these conditions, we divide the evaluation area where the ED is located into different n_{cl} clusters. A cluster is defined as a group of neighboring locations with similar RSSI levels. The more clusters with different RSSI levels assigned, the more heterogeneous and unpredictable the communication becomes. Each cluster is generated atop the path loss from the ED to the gateway using k-means clustering, with a normally distributed random noise component ranging from -5 to 5 dB, uniformly applied to all the points of each area or cluster. In this experiment, we evaluate different scenarios ranging from clusters $n_{cl} = 1$ to $n_{cl} = 1000$.

As shown in Fig. 6, ADR initially has a relatively low PDR for $n_{cl} = 1$, averaging approximately 32% lower than our ADRL. Although its performance improves slightly as n_{cl} increases, with the PDR reaching values around 7% lower than ADRL for $n_{cl} = 1000$, this improvement comes at the cost of significantly higher energy consumption, around $1.5\times$ higher than ADRL. Therefore, ADRL consistently outperforms ADR in terms of the PDR and energy consumption. BADR, while having the lowest energy consumption among all the evaluated algorithms, saving slightly less than half the energy per packet compared to ADRL, shows the lowest PDR and remains almost invariant to the changes in n_{cl} due to its “blind” use of SFs. In contrast, G-ADR maintains a high PDR across all values of n_{cl} , though it slightly decreases with a higher number of clusters (7.3% higher than ADRL for $n_{cl} = 1000$). However, it has the highest energy consumption, with values that far exceed those of ADRL. This high energy use is due to its predominant selection of SF12, leading to inefficient energy consumption. Lastly, our ADRL mechanism demonstrates a relatively high PDR across all values of n_{cl} , with a notable decrease as n_{cl} increases, but with values still consistently higher than those of ADR, while energy consumption increases slightly with higher n_{cl} values due to the presence of more noise. As we observe, ADRL strikes a good balance between the PDR and energy efficiency compared to the other mechanisms, even under varying levels of channel conditions and mobility.

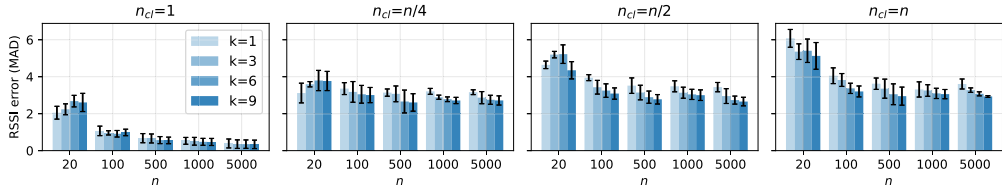


Fig. 7. Analysis of the accuracy of the RSSI estimation, including the Mean Absolute Deviation (MAD) of RSSI between the measured and estimated values for each position where the ED transmits to the GW, as a function of the number of buffered RSSI samples in the network server (n), the number of neighboring RSSI samples (k), and the number of clusters (n_{cl}).

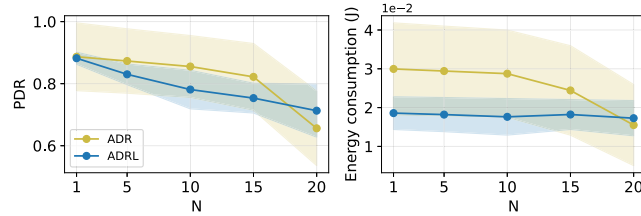


Fig. 8. Evaluation of the mean PDR and mean Energy consumption per packet for different numbers of received SNR and RSSI frames before updating the transmission configuration, using $\psi_{th} = 0.8$, $k = 3$, and $n = 1000$.

5.3. Analysis of KNN accuracy for RSSI estimation

In this experiment, we analyze the impact of channel conditions on the accuracy of the KNN-based RSSI estimation used by ADRL. We also evaluate the effect of varying the number of clusters on these estimations, examining how different cluster configurations influence overall performance. As shown in Fig. 7, when $n_{cl} = 1$, the RSSI error decreases as n_{cl} increases, indicating greater accuracy in position estimation with more samples in the buffer. The lowest k value, $k = 1$, results in the highest error, since it relies on a single nearest neighbor, which can lead to more variability and less accurate estimates. Conversely, higher k values lead to less error, as the estimation is based on a larger number of neighbors, providing a more robust average and reducing the impact of outliers. For $n_{cl} = n/4$, the RSSI error also decreases with increasing n , and the error decreases with increasing k , although the variation is not as pronounced as in the case of $n_{cl} = 1$. This suggests that while increasing the number of neighbors generally improves accuracy, the impact is influenced by the number of clusters. In the case of $n_{cl} = n/2$, we observe a similar trend, where the error decreases with increasing n and k , although the overall error remains higher compared to the previous two cases. This indicates that as the number of clusters increases, position estimation becomes more challenging. For $n_{cl} = n$, the RSSI error continues to decrease with n , but there is less difference in error for different values of k . The error is the highest compared to the other scenarios, indicating that having a large number of clusters that equal the number of samples can introduce more variability and difficulty in achieving precise position estimation.

5.4. Analysis of transmission parameter update frequency

The default ADR typically updates its transmission configuration every 20 samples of SNR data [8,28–30]. However, as suggested by Semtech [58], this could also be done dynamically using the ADRackReq bit in EDs. In this experiment, we further evaluate the performance of the default ADR mechanism by analyzing the effect of varying the number of required SNR frames on transmission parameter updates (N). In particular, we investigate how varying the frequency of transmission configuration updates, in terms of the number of received SNR frames for ADR or RSSI frames for ADRL before updating the transmission parameters, affects system performance and energy efficiency. We compare these results to demonstrate that ADRL still outperforms the default ADR even while reducing N . As shown in Fig. 8, the ADR mechanism achieves a higher PDR for lower values of N due to its frequent use of the highest spreading factor (SF12), which increases reliability but leads to significantly higher energy consumption. In contrast, ADRL strikes a balance between PDR and energy consumption, compensating a slight reduction in PDR with a significant improvement in energy efficiency. For $N = 1$, ADR consumes significantly more energy for achieving a similar PDR than ADRL. In addition, as confirmed in Fig. 8, ADRL demonstrates significantly lower energy consumption across all N values. This reduction in energy consumption validates ADRL's strategy of selecting transmission parameters that balance PDR and consumption, resulting in a more efficient overall performance.

6. Conclusions

This paper introduces a novel reconfigurable transmission mechanism (ADRL) that dynamically adapts transmission parameters to enhance performance and energy efficiency in mobile LoRa-based IoT networks, complying with duty-cycle constraints. ADRL

ensures the probability of packet decoding required by the mobile IoT application, dynamically balancing energy consumption and the delivery ratio. We employ supervised machine learning to predict RSSI and Deep Reinforcement Learning (DRL) to create a controller that determines the most suitable transmission configuration. The experimental results for different scenarios with varying channel conditions and mobile EDs demonstrate the effectiveness of ADRL compared to state-of-the-art approaches, increasing the packet delivery ratio up to 57% while reducing energy consumption up to 15%. In future work, we will investigate the congestion and interference effects of an increasing number of end devices, the Doppler effect, which might have a considerable impact on mobile LoRa communications, and explore multi-agent approaches where nodes collaboratively influence each other's decisions to solve complex and shared problems.

CRedit authorship contribution statement

Laura Acosta-Garcia: Writing – review & editing, Writing – original draft, Visualization, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Juan Aznar-Poveda:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Antonio Javier Garcia-Sanchez:** Writing – review & editing, Visualization, Validation, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization. **Jakob Hollenstein:** Writing – review & editing, Visualization, Validation, Software, Methodology, Investigation, Formal analysis. **Joan Garcia-Haro:** Writing – review & editing, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis. **Thomas Fahringer:** Writing – review & editing, Visualization, Validation, Supervision, Methodology, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the grants PID2023-148214OB-C21, PCI2024-153485, and TED2021-129336B-I00, funded by MICIU/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR. This work was also funded by Fundación Séneca (22236/PDC/23), and a result of the ThinkInAzul and AgroAINext programs, funded by Ministerio de Ciencia, Innovación y Universidades (MICIU) with funding from European Union NextGeneration EU/PRTR-C17.I1 and by Fundación Séneca with funding from Comunidad Autónoma Región de Murcia (CARM). This research was also contextualized to DAIMon, a cascade funding action deriving from the Horizon Europe project aerOS, funded by the European Commission, under grant number 101069732.

Data availability

Data will be made available on request.

References

- [1] J.P.S. Sundaram, W. Du, Z. Zhao, A survey on lora networking: Research problems, current solutions, and open issues, *IEEE Commun. Surv. Tutorials* 22 (1) (2019) 371–388.
- [2] P. Gkotsiopoulos, D. Zorbas, C. Douligeris, Performance determinants in LoRa networks: A literature review, *IEEE Commun. Surv. Tutorials* 23 (3) (2021) 1721–1758.
- [3] P. Ferrari, E. Sisinni, D.F. Carvalho, A. Depari, G. Signoretti, M. Silva, I. Silva, D. Silva, On the use of LoRaWAN for the internet of intelligent vehicles in smart city scenarios, in: 2020 IEEE Sensors Applications Symposium, SAS, IEEE, 2020, pp. 1–6.
- [4] V. Moysiadias, T. Lagkas, V. Argyriou, A. Sarigiannidis, I.D. Moscholios, P. Sarigiannidis, Extending ADR mechanism for LoRa enabled mobile end-devices, *Simul. Model. Pr. Theory* 113 (2021) 102388.
- [5] A.P.A. Torres, C.B. Da Silva, H. Tertuliano Filho, An experimental study on the use of LoRa technology in vehicle communication, *IEEE Access* 9 (2021) 26633–26640.
- [6] H. Soy, Coverage analysis of LoRa and NB-IoT technologies on LPWAN-based agricultural vehicle tracking application, *Sensors* 23 (21) (2023) 8859.
- [7] D. Saibannavar, M.M. Math, U. Kulkarni, A survey on on-board diagnostic in vehicles, in: International Conference on Mobile Computing and Sustainable Informatics: ICMCSI 2020, Springer, 2021, pp. 49–60.
- [8] S. Li, U. Raza, A. Khan, How agile is the adaptive data rate mechanism of LoRaWAN? in: 2018 IEEE Global Communications Conference, GLOBECOM, IEEE, 2018, pp. 206–212.
- [9] N. Benkahla, H. Tounsi, S. Ye-Qiong, M. Frikha, Enhanced ADR for LoRaWAN networks with mobility, in: 2019 15th International Wireless Communications & Mobile Computing Conference, IWCMC, IEEE, 2019, pp. 1–6.
- [10] A. Farhad, D.-H. Kim, S. Subedi, J.-Y. Pyun, Enhanced LoRaWAN adaptive data rate for mobile internet of things devices, *Sensors* 20 (22) (2020) 6466.
- [11] K. Anwar, T. Rahman, A. Zeb, I. Khan, M. Zareei, C. Vargas-Rosales, Rm-adr: Resource management adaptive data rate for mobile application in LoRaWAN, *Sensors* 21 (23) (2021) 7980.
- [12] A. Augustin, J. Yi, T. Clausen, W.M. Townsley, A study of LoRa: Long range & low power networks for the internet of things, *Sensors* 16 (9) (2016) 1466.
- [13] M. Jouhari, K. Ibrahim, J.B. Othman, E.M. Amhoud, Deep reinforcement learning-based energy efficiency optimization for flying LoRa gateways, 2023, arXiv preprint [arXiv:2302.05214](https://arxiv.org/abs/2302.05214).

- [14] M.S.A. Muthanna, A. Muthanna, A. Rafiq, M. Hammoudeh, R. Alkanhel, S. Lynch, A.A. Abd El-Latif, Deep reinforcement learning based transmission policy enforcement and multi-hop routing in QoS aware LoRa IoT networks, *Comput. Commun.* 183 (2022) 33–50.
- [15] J. Mhatre, A. Lee, Dynamic reinforcement learning based scheduling for energy-efficient edge-enabled LoRaWAN, in: 2022 IEEE International Performance, Computing, and Communications Conference, IPCCC, IEEE, 2022, pp. 412–413.
- [16] Y. Yazid, A. Guerrero-González, I. Ez-Zazi, A. El Oualkadi, M. Arioua, A reinforcement learning based transmission parameter selection and energy management for long range internet of things, *Sensors* 22 (15) (2022) 5662.
- [17] R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 2018.
- [18] A. Farhad, D.-H. Kim, J.-S. Yoon, J.-Y. Pyun, Feasibility study of the LoRaWAN blind adaptive data rate, in: 2021 Twelfth International Conference on Ubiquitous and Future Networks, ICUFN, IEEE, 2021, pp. 67–69.
- [19] A. Farhad, J.-Y. Pyun, HADR: A hybrid adaptive data rate in LoRaWAN for internet of things, *ICT Express* 8 (2) (2022) 283–289.
- [20] B. Robbe, W. Danny, A qos-aware adaptive mobility handling approach for lora-based iot systems, in: 2018 IEEE 12th International Conference on Self-Adaptive and Self-Organizing Systems, SASO, IEEE, 2018, pp. 130–139.
- [21] L. Acosta-Garcia, J. Aznar-Poveda, A.J. Garcia-Sanchez, J. Garcia-Haro, T. Fahringer, Dynamic transmission policy for enhancing LoRa network performance: A deep reinforcement learning approach, *Internet Things* 24 (2023) 100974.
- [22] A. Durand, N. El Rachkidy, A. Guitton, MADERE: Mobile adaptive datarate for LoRaWAN, in: 2023 IEEE Wireless Communications and Networking Conference, WCNC, IEEE, 2023, pp. 1–6.
- [23] N. Benkahla, H. Tounsi, Y.-Q. Song, M. Frikha, VHMM-based E-ADR for LoRaWAN networks with unknown mobility patterns, in: 2021 International Wireless Communications and Mobile Computing, IWCMC, IEEE, 2021, pp. 86–91.
- [24] R.M. Sandoval, A.-J. Garcia-Sanchez, J. Garcia-Haro, T.M. Chen, Optimal policy derivation for transmission duty-cycle constrained LPWAN, *IEEE Internet Things J.* 5 (4) (2018) 3114–3125.
- [25] R.M. Sandoval, A.-J. Garcia-Sanchez, J. Garcia-Haro, Performance optimization of LoRa nodes for the future smart city/industry, *EURASIP J. Wirel. Commun. Netw.* 2019 (1) (2019) 1–13.
- [26] L. Specification, LoRaWAN specification v1.1, LoRaWANtm- Specif.- V11 (2017).
- [27] A. Farhad, D.-H. Kim, D. Kwon, J.-Y. Pyun, An improved adaptive data rate for LoRaWAN networks, in: 2020 IEEE International Conference on Consumer Electronics-Asia, ICCE-Asia, IEEE, 2020, pp. 1–4.
- [28] M. Slabicki, G. Premsankar, M. Di Francesco, Adaptive configuration of LoRa networks for dense IoT deployments, in: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2018, pp. 1–9.
- [29] K.Q. Abdelfadeel, V. Cionca, D. Pesch, Fair adaptive data rate allocation and power control in LoRaWAN, in: 2018 IEEE 19th International Symposium on “a World of Wireless, Mobile and Multimedia Networks”, WoWMoM, IEEE, 2018, pp. 14–15.
- [30] J. Finnegan, R. Farrell, S. Brown, Analysis and enhancement of the LoRaWAN adaptive data rate scheme, *IEEE Internet Things J.* 7 (8) (2020) 7171–7180.
- [31] J.G. Panicker, M. Azman, R. Kashyap, A LoRa wireless mesh network for wide-area animal tracking, in: 2019 IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT, IEEE, 2019, pp. 1–5.
- [32] M.I. Arsyad, A.A. Pekerti, B.A. Nugraha, Time-slotted LoRaWAN implementation of GPS tracker system for post-disaster utilization, in: 2020 International Conference on Information Technology Systems and Innovation, ICITSI, IEEE, 2020, pp. 172–178.
- [33] G.L. Tracker, L. Millor, K. Ruckstuhl, E. Alan Jin, C. Alan Ko, C. Christina Zhao, C. Sheel Bhardwaj, Open-source GPS long range (LoRa) tracker for rural environments, 2020.
- [34] J. Aznar-Poveda, E. Egea-López, A.-J. García-Sánchez, Cooperative awareness message dissemination in EN 302 637-2: An adaptation for winding roads, in: 2020 IEEE 91st Vehicular Technology Conference, VTC2020-Spring, IEEE, 2020, pp. 1–5.
- [35] L. Acosta-Garcia, J. Aznar-Poveda, A.-J. Garcia-Sanchez, J. Garcia-Haro, T. Fahringer, Proactive adaptation of data rate in mobile lora-based IoT devices using machine learning, in: 2024 IEEE 99th Vehicular Technology Conference, VTC2024-Spring, IEEE, 2024, pp. 1–5.
- [36] J. Aznar-Poveda, E. Egea-López, A.-J. García-Sánchez, J. García-Haro, Advisory speed estimation for an improved V2X communications awareness in winding roads, in: 2020 22nd International Conference on Transparent Optical Networks, ICTON, IEEE, 2020, pp. 1–4.
- [37] J. Purohit, X. Wang, S. Mao, X. Sun, C. Yang, Fingerprinting-based indoor and outdoor localization with LoRa and deep learning, in: GLOBECOM 2020-2020 IEEE Global Communications Conference, IEEE, 2020, pp. 1–6.
- [38] A.R. Cassandra, A survey of POMDP applications, in: Working Notes of AAAI 1998 Fall Symposium on Planning with Partially Observable Markov Decision Processes, vol. 1724, 1998.
- [39] LoRaWAN[®] specification v1.1, 2023, <https://resources.lora-alliance.org/technical-specifications/lorawan-specification-v1-1> (Accessed 13 December 2023).
- [40] J. Courjault, B. Vrigneau, O. Berder, Fast performance evaluation of LoRa communications over Rayleigh fading channels, in: 2019 IEEE Wireless Communications and Networking Conference Workshop, WCNW, IEEE, 2019, pp. 1–5.
- [41] A. Lavric, V. Popa, A LoRaWAN: Long range wide area networks study, in: 2017 International Conference on Electromechanical and Power Systems, SIELMEN, IEEE, 2017, pp. 417–420.
- [42] T. Elshabrawy, J. Robert, Closed-form approximation of LoRa modulation BER performance, *IEEE Commun. Lett.* 22 (9) (2018) 1778–1781.
- [43] X. Zhang, Z. Lin, S. Gong, B. Gu, D. Niyato, Multiagent reinforcement learning with an attention mechanism for improving energy efficiency in LoRa networks, in: GLOBECOM 2023-2023 IEEE Global Communications Conference, IEEE, 2023, pp. 4152–4157.
- [44] U. Coutaud, M. Heusse, B. Tourancheau, LoRa channel characterization for flexible and high reliability adaptive data rate in multiple gateways networks, *Computers* 10 (4) (2021) 44.
- [45] A. Trunov, A. Malcheniuk, The Mesh technology for improving of end nodes updating for LoRaWAN network, in: 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, vol. 1, IEEE, 2018, pp. 164–167.
- [46] D. Zorbas, Improving LoRaWAN downlink performance in the EU868 spectrum, *Comput. Commun.* 195 (2022) 303–314.
- [47] Y. Sun, J. Chen, S. He, Z. Shi, High-confidence gateway planning and performance evaluation of a hybrid LoRa network, *IEEE Internet Things J.* 8 (2) (2020) 1071–1081.
- [48] Z. Xu, S. Tong, P. Xie, J. Wang, From demodulation to decoding: Toward complete lora phy understanding and implementation, *ACM Trans. Sens. Netw.* 18 (4) (2023) 1–27.
- [49] M.A.A. Khan, H. Ma, S.M. Aamir, C.A. Baris, Experimental comparison of SNR and RSSI for LoRa-ESL based on machine clustering and arithmetic distribution, 2022, arXiv preprint arXiv:2210.15122.
- [50] R. Kufakunesu, G.P. Hancke, A.M. Abu-Mahfouz, A survey on adaptive data rate optimization in LoRaWAN: Recent solutions and major challenges, *Sensors* 20 (18) (2020) 5044.
- [51] M.S. Philip, P. Singh, Energy consumption evaluation of LoRa sensor nodes in wireless sensor network, in: 2021 Advanced Communication Technologies and Signal Processing, ACTS, IEEE, 2021, pp. 1–4.
- [52] R. Serati, B. Teymuri, N.A. Anagnostopoulos, M. Rasti, ADR-lite: A low-complexity adaptive data rate scheme for the lora network, in: 2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob, IEEE, 2022, pp. 296–301.
- [53] Gymnasium documentation, 2024, <https://gymnasium.farama.org/index.html>. (Accessed 06 November 2024).
- [54] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, 2017, arXiv preprint arXiv:1707.06347.
- [55] ADRL GitHub repository, 2024, <https://github.com/LauraAcostaG/ADRL-LoRa>.
- [56] L. Alliance, RP002-1.0. 0 LoRaWAN regional parameters, 2019, LoRa Alliance: Fremont, CA, USA.
- [57] Lora connect transceiver, SX1276, 137 MHz to 1020 MHz | semtech, 2025, URL <https://www.semtech.com/products/wireless-rf/lora-connect/sx1276>. [Online; Accessed 27 January 2025].
- [58] S. Corporation, LoRaWAN—Simple rate adaptation recommended algorithm, 2016.