Journal of Network and Computer Applications 250 (2026) 104460

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Research paper

PSO-Enhanced Reinforcement Learning for Resource Allocation in LoRaWAN IoT Network Slicing

Fatima Zahra Mardi [a,*], Yassine Hadjadj-Aoul [b], Miloud Bagaa [c], Nabil Benamar [a,d]

[a] Moulay Ismail University, Meknes, Morocco
[b] University of Rennes, Inria, IRISA, Rennes, France
[c] Department of Electrical and Computer Engineering, Université du Québec, Trois-Rivières, QC, Canada
[d] Al Akhawayn University, Ifrane, Morocco

ARTICLE INFO

Keywords:
Internet of Things
LoRaWAN
Network slicing
Particle swarm optimization
Deep Q-network
Resource allocation

ABSTRACT

Recent years have witnessed drastic growth in the use of wireless sensor networks, mainly due to the increasing adoption of Internet of Things (IoT) applications in various sectors. However, efficiently allocating transmission parameters in these networks, particularly within the LoRaWAN framework, poses a significant challenge due to the diverse requirements of multiple services, each demanding varying levels of bandwidth and reliability. To address this challenge, this paper introduces three innovative resource allocation approaches for LoRaWAN network slicing: Deep Q-Network-3 (DQN-3), Particle Swarm Optimization (PSO), and the hybrid Particle Swarm Optimization–Deep Q-Network (PSO–DQN). These methods aim to dynamically assign transmission parameters, including transmission power (TP), spreading factor (SF), and coding rate (CR), to optimize network performance while meeting the Service Level Agreement (SLA) requirements of the supported applications. The DQN-3 approach focuses solely on learning optimal policies through experience, while the standalone PSO efficiently explores the parameter space with low complexity but lacks adaptability in dynamic environments. However, the proposed PSO–DQN approach combines the exploratory capabilities of PSO with the decision-making strengths of DQN. The PSO facilitates the exploration of the parameters' configurations, enabling the algorithm to identify optimal resource allocations even in complex and dynamic environments. These configurations are then passed to the DQN, which refines them to enhance efficiency and performance. Our findings reveal that all three approaches significantly improve reliability and energy efficiency while meeting the Service Level Agreement (SLA) requirements for various services in LoRaWAN networks. The combined method, PSO–DQN, surpasses both the DQN-3 and PSO strategies, highlighting the benefits of merging heuristic techniques and deep learning. Our simulation results demonstrate that the proposed framework outperforms existing methods.

1. Introduction

Today, the rapid growth of the Internet of Things (IoT) technology has led to billions of devices connecting and communicating through standardized communication protocols, creating networks that meet the needs of a wide range of applications (Haxhibeqiri et al., 2018). However, this significant increase in connected devices has made network management more complex than ever. The expanding network not only complicates management tasks, but also introduces new challenges in maintaining reliable connectivity and ensuring optimal performance (Farhan et al., 2017). To address these challenges, innovative technologies are being deployed to effectively manage these dynamic environments, optimize resource utilization, and meet the diverse demands of various applications. Among these solutions, Low Power Wide Area Network (LPWAN) technology has emerged as one of the most promising options for IoT applications (Gu et al., 2020).

The Long Range “LoRa” technology, developed by Semtech, employs chirp spread spectrum (CSS) modulation techniques to facilitate long-range communication (Van den Abeele et al., 2017). In LoRa's CSS modulation, each data symbol is represented as a chirp, which is a signal that changes its frequency linearly over time (Nguyen et al., 2019). This modulation technique allows for efficient and reliable data transmission over significant distances, making LoRa an ideal solution for various applications in IoT environments.

LoRaWAN is an open source LPWAN protocol developed by the LoRa Alliance (Alliance, 2015), built on top of the physical layer of LoRa. It provides an efficient communication framework for IoT

---

* Corresponding author.
E-mail address: fa.mardi@edu.umi.ac.ma (F.Z. Mardi).

https://doi.org/10.1016/j.jnca.2026.104460
Received 14 August 2025; Received in revised form 26 December 2025; Accepted 24 February 2026
Available online 5 March 2026

devices, offering long-range connectivity, low power consumption, and cost-effective deployment. These features make LoRaWAN an ideal choice for a diverse array of IoT applications, ensuring reliable connectivity across various use cases.

As the number of connected devices continues to grow, managing the network efficiently becomes more challenging. The proliferation of IoT devices, coupled with their diverse traffic characteristics, has highlighted the importance of traffic classification in tackling the challenges of IoT applications (Tahaei et al., 2020). Network slicing offers a promising solution, allowing the creation of multiple virtual networks within a single physical infrastructure; each can be customized to meet the specific needs of various applications (Afolabi et al., 2018). This approach not only divides the network, but also significantly improves overall efficiency and improves the quality of service for individual applications. By incorporating network slicing into LoRaWAN, the framework can allocate resources in a highly effective and isolated manner, addressing the diverse needs of IoT applications. Thus, network slicing is recognized as an essential strategy for optimizing performance in LoRaWAN environments.

In addition to network slicing, efficient management of transmission parameters such as Spreading Factor (SF), Transmission Power (TP), and Coding Rate (CR) within LoRaWAN is crucial for ensuring efficient network operations, as the number of connected devices continues to rise. As multiple devices attempt to transmit data simultaneously, the probability of data collision increases significantly. These collisions can lead to network congestion, resulting in a substantial degradation of key performance metrics, such as Packet Delivery Ratio (PDR), energy consumption (EC), latency, throughput, and overall network reliability.

To address these dynamic resource-allocation challenges while respecting computational complexity constraints, this paper proposes and evaluates three complementary approaches for LoRaWAN network slicing by dynamically adjust the SF, TP, and CR to improve reliability and energy efficiency across different slices while meeting service-level agreement requirements. The first approach, DQN-3, is a DRL-based standalone solution, allowing systems to dynamically learn optimal strategies through trial-and-error interactions with the environment (Alwarafy et al., 2021), to dynamically adjust transmission parameters. DQN-3 is particularly well suited to highly dynamic environments where density, interference, or traffic change rapidly. After offline training, it enables very fast online decision-making, making it relevant for large-scale networks requiring real-time adaptation with minimal latency. The second approach, PSO, meanwhile, a population-based metaheuristic inspired by swarm intelligence, enables devices to efficiently explore and exploit the parameter space. This process identifies optimal configurations for the transmission parameters, including the SF, TP, and CR. Is a lightweight, low-complexity solution that is effective for static or slightly variable networks where optimization can be performed periodically. Its low computational cost makes it a good choice for gateways with limited resources or for the initial planning phase.

While metaheuristic algorithms and RL-based approaches have their respective strengths, they also face distinct limitations. Metaheuristics excel in rapid convergence and global optimization but may struggle to adapt to dynamic changes in real-time. Conversely, RL models are highly adaptable but require extensive training and exploration to achieve optimal performance. Recognizing these complementary strengths, our third approach, PSO–DQN, is a hybrid method that combines Particle Swarm Optimization (PSO) with Deep Reinforcement Learning (DRL). In this hybrid framework, PSO is employed to optimize the initial set of transmission parameters for each LoRa node, while DRL continuously adapts these parameters to changing network conditions. This integration effectively balances exploration and exploitation, leading to greater adaptability and improved overall performance. PSO–DQN targets dense, heterogeneous scenarios that are highly susceptible to interference, where it is necessary to combine effective global exploration (PSO) with adaptive online refinement (DQN). Thanks to the high-quality initial solutions provided by PSO, DQN avoids lengthy and costly random exploration, greatly reducing the number of learning iterations required and improving convergence stability.

Each of these approaches optimizes the parameter allocation to meet the specific requirements of each supported service, achieving significant improvements in PDR and energy efficiency. These solutions are highly suitable for addressing the diverse needs of IoT environments within LoRaWAN.

The remainder of this paper is organized as follows. Section 2 presents the related work and paper contributions, Section 3 provides the system model, and Section 4 presents the solution framework. Then we evaluate our simulation results in Section 5. Finally, Section 6 concludes the paper.

2. Related work and paper contributions

In this section, we present the relevant research context for our work. Section 2.1 details the related work. Following this, Section 2.2 discusses the shortcomings of these works and outlines the main contributions of our proposed solutions.

2.1. Related work

The research community has been heavily involved in extensive studies and discussions about the LoRaWAN network. A series of research initiatives have focused on optimizing resource parameters within LoRa networks.

In Hamdi et al. (2020), the authors propose a new dynamic spreading factor assignment scheme for LoRaWAN that adapts to instantaneous channel conditions rather than relying solely on the distance from the gateway, as commonly considered in many methods. This approach aims to minimize the average Symbol Error Rate (SER) and improve spectral efficiency, allowing for a greater number of connected devices. Similarly, the papers (Minhaj et al., 2023; Navas et al., 2023) present machine learning-based approaches for resource allocation in LoRaWAN, addressing the inefficiencies of traditional methods in managing the growing number of IoT devices. Although these studies offer valuable insights into the parameter optimization, they do not provide the comprehensive resource management offered by network slicing.

Recently, several studies have started exploring the application of network slicing within LoRaWAN. In Mardi et al. (2022), the authors present a strategy for network slicing in LoRa networks that employs a centralized coalition game to efficiently manage LoRa nodes within the LoRaWAN framework. They begin by forming coalitions through the K-Means clustering algorithm, which facilitates the dynamic reassignment of LoRa nodes among coalitions to improve reliability for each network slice. However, this static framework can lead to inefficiencies in resource utilization and may struggle to adapt to the varying demands of different IoT applications, ultimately affecting overall network performance. Meanwhile, in Mardi et al. (2023), the authors examine network slicing in LoRaWAN using the Heuristic-Deep Q-Network (H-DQN) method for resource allocation. They propose an inter-service allocation strategy that employs the deep Q-Network (DQN) algorithm to distribute virtual resource blocks to services. Furthermore, a heuristic algorithm is used to assign transmission probabilities to LoRa nodes for each service, optimizing the Packet Delivery Rate (PDR) while maintaining service priorities. In Mardi et al. (2024) the authors provide valuable insights into resource allocation strategies for LoRaWAN, focusing on the potential of Multi-Armed Bandit (MAB)-based methods to optimize network performance in the context of IoT applications, with the main objective to dynamically and efficiently allocate resources to network slices while maximizing the PDR and ensuring compliance with each service's Service Level Agreement (SLA). Additionally, the authors in Fall et al. (2023) propose a resource reservation mechanism that utilizes device trajectory prediction and traffic differentiation to enhance the Quality of Service (QoS) for both priority and non-priority traffic. By reserving resources in advance based on predicted vehicular movements, the approach aims to reduce rejection rates and improve overall network efficiency. While these studies provide valuable insights into network slicing and resource allocation in LoRaWAN, they are limited in their focus on optimizing transmission parameters and do not sufficiently address other critical factors, such as energy efficiency and latency.

Research efforts have advanced LoRaWAN slicing performance through sophisticated parameter tuning. Dawaliby et al. (2021) developed an innovative multi-faceted approach that combines Maximum Likelihood Estimation (MLE), the Geometric Mean Method (GMM), and the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). Their system dynamically optimizes two critical transmission parameters — spreading factors and transmission power — across different network slices. This optimization framework ensures that Quality of Service (QoS) requirements are consistently met while maintaining efficient resource allocation across the network.

Nevertheless, they do not leverage the adaptive and exploratory capabilities of other techniques, such as machine learning approaches, which can dynamically adjust to changing network conditions. In Messaoud et al. (2020), Messaoud et al. utilize a Deep Federated Q-Learning (DFQL) approach for channel assignment in LoRaWAN networks, focusing on adjustments to transmission power and spreading factors based on previous agent experiences. This strategy aims to enhance the QoS for each network slice. However, the authors focus only on the allocation of two parameters, while these parameters are important, limiting the optimization to only two dimensions restricts the system's adaptability and performance, especially in dense IoT environments. Moreover, in Ossongo et al. (2024), the authors propose a network slicing architecture specifically designed for IoT services, leveraging Software-Defined Networking (SDN) and Network Functions Virtualization (NFV) technologies. It adopts a decentralized architecture in which each device makes its own decisions, followed by a federated reinforcement learning process to effectively adjust transmission power and spreading factor parameters. In spite of that, this method is less realistic, as it relies on end devices, which generally have very limited capabilities and cannot support heavy computational loads. Similarly, the authors in Tellache et al. (2022) propose a real LoRaWAN platform using a Deep Reinforcement Learning (DRL)-based approach that uses multi-agent DQN to allocate spreading factor and transmission power to IoT devices, demonstrating superior performance over the traditional ADR scheme in meeting QoS requirements across various network slices. Nonetheless, this approach remains limited in scope due to the optimization of only two parameters. In addition, resource allocation occurs at the nodes level, each node has a DRL agent that decides the appropriate values of SF and TP for that specific node, which can lead to high computational costs, additional energy consumption due to real-time learning and decision-making processes, potentially restricting use on energy-constrained devices.

Existing studies on LoRaWAN slicing rely exclusively on pure learning, leading to longer convergence times. However, recent works in other wireless fields have demonstrated that combining metaheuristics with reinforcement learning (RL) can significantly accelerate convergence in complex wireless environments. For instance, DQN and PSO has been coupled for resource allocation and task offloading in 6G edge-intelligence systems in which they integrate deep reinforcement learning (DQN) for adaptive policy learning and then the PSO for fine-grained resource search (Tang, 2025). Similarly, cooperative RL-metaheuristic models have been studied for routing/clustering in wireless sensor networks, in which reinforcement learning is used to guide a metaheuristic in the joint search for the best cluster heads and routing paths (Wang et al., 2023). Finally, a hybrid metaheuristic-RL-based approach have been proposed for frequency resource management in 6G User-Centric cell-free mMIMO, where the authors used Aquila Optimizer (AO) for its exploratory capabilities, while Actor–Critic Reinforcement Learning (AC-RL) is chosen for its adaptive learning strengths in handling complex multi-objective problems (Cheggour and Loscri, 2025).

A critical gap remains in the application of such hybrid approaches to LoRaWAN slicing, which presents a highly constrained environment. Our PSO–DQN framework is specifically designed to meet these unique challenges. In contrast to the works in Tang (2025) and Wang et al. (2023), our approach first utilizes a metaheuristic to provide a 'warm-start,' effectively eliminating the initial exploration phase of the RL process. This is particularly critical because our architecture deals with the discrete and combinatorial action space of LoRaWAN parameters (SF, TP, CR), where pure RL is typically overwhelmed by infeasible solutions, leading to slow convergence and high cost. Once a feasible baseline is established by metaheuristic (PSO), the reinforcement learning component (DQN) takes over to refine these solutions. This allows the system to continuously adapt to time-varying traffic in real-time while ensuring the SLA. Table 1 presents a comparison of the selected solutions related to this study.

2.2. Limitations and contributions

Based on a comprehensive literature review and thorough analysis of resource allocation, particularly in optimizing transmission parameters within LoRaWAN networks, managing overall network performance remains an evolving field of research. Existing studies still exhibit significant limitations.

In which, all existing resource optimization solutions for LoRaWAN network slicing rely on the allocation of the spreading factors and the transmission power, neglecting the critical role of the Coding Rate (CR). This may lead to inefficiencies in data transmission and energy usage, resulting in suboptimal network performance, particularly as the number of services increases, which negatively impacts reliability and efficiency. Moreover, some solutions fail to address energy efficiency, which is crucial for the effective operation of LoRaWAN communications. This limits their ability to provide a comprehensive framework that optimizes performance in diverse and demanding scenarios, where energy efficiency is essential for maintaining functionality and extending the operational life of the devices. Furthermore, in some solutions, the importance of network slicing is neglected, even though it is essential for optimizing resource allocation and performance across multiple services. Without it, resources cannot be effectively adapted to the diverse needs of different services, leading to inefficiencies and compromising performance in supporting various applications on a single infrastructure.

In response to these limitations, our main contributions are the following:

- We focus on enhancing energy efficiency while ensuring high reliability in IoT environments, by introducing resource allocation solutions, that include the allocation of coding rate along with the spreading factor and the transmission power. In contrast to conventional two-parameter strategies, incorporating CR adjustment along with SF, and TP enables our methods to effectively balance error robustness and data transmission efficiency. This crucial capability allows the proposed approach to maximize overall network performance, while achieving significant energy savings and mitigating interferences more effectively in high-density scenarios.
- We present and analyze two standalone solutions with the allocation of SF, TP, and CR three parameters (DQN-3 and PSO) to evaluate the robustness of our new hybrid solution. The DQN-3 solution learns optimal resource-allocation policies from experience but can be computationally heavy and slow to converge due to extensive exploration. The second solution employs PSO alone, which efficiently explores the parameter space to find

Table 1
Summary of resource allocation approaches.

Ref.	Allocation			Slicing	Performance metric		Limitation
	SF	TP	CR		Reliability	Energy	
Hamdi et al. (2020)	✓	✗	✗	✗	✓	✗	Lacks orchestration for isolated slices with distinct QoS requirements.
Minhaj et al. (2023)	✓	✓	✗	✗	✓	✓	
Navas et al. (2023)	✓	✗	✗	✗	✓	✓	
Mardi et al. (2022)	✗	✗	✗	✓	✓	✓	Lacks transmission parameter allocation, resulting in an incomplete analysis.
Mardi et al. (2023, 2024), Fall et al. (2023)	✗	✗	✗	✓	✓	✗	
Dawaliby et al. (2021), Messaoud et al. (2020), Ossongo et al. (2024), Tellache et al. (2022)	✓	✓	✗	✓	✓	✓	Lacks Coding Rate (CR) optimization, reducing reliability and efficiency as traffic increases.

near-optimal transmission configurations with low computational complexity. Although effective in improving energy efficiency and reliability, it lacks the depth needed for full optimization in dynamic environments.

- We introduce a new hybrid framework, PSO-DQN, which combines the strengths of both PSO and DQN. In this method, PSO generates promising initial combinations of transmission parameters for each device, establishing a strong starting point aimed at achieving an optimal balance between reliability and energy efficiency. These PSO-generated combinations serve as a foundation for training the DQN, which further refines the configurations. By combining PSO's rapid convergence capabilities with DQN's fine-tuning strengths, the hybrid PSO–DQN approach offers a significantly more efficient and effective resource allocation process compared to either method used alone. This collaboration not only reduces the time and computational resources needed for convergence but also results in enhanced performance, making it a highly compelling choice for optimizing transmission parameters and addressing the challenges of IoT environments in LoRaWAN.

3. System model

This section details the comprehensive system framework adopted in our study and outlines the problem formulation. We first describe the slicing architecture in Section 3.1. Then, we present the network and communication models in Sections 3.2 and 3.3, respectively. Finally, we introduce the performance metrics in Section 3.4 and formally state the problem formulation in Section 3.5.

3.1. Slicing architecture

Network slicing in LoRaWAN aims to efficiently manage resources by creating multiple virtual networks on a shared physical infrastructure. Each network slice allows for custom resource allocation to meet the specific needs of different applications. This approach improves flexibility and optimizes performance across the network. In this context, we consider a network slicing architecture comprising a set $\mathbb{S} = \{1, \dots, S\}$ of IoT network slices. We adopt a strict one-to-one mapping strategy, where each slice $s \in \mathbb{S}$ is exclusively dedicated to a specific IoT service class. These slices operate on a shared physical infrastructure and utilize the same resources provided by the LoRa gateways.

Without loss of generality, the proposed network slicing architecture comprises three distinct network slices, each serving a specific IoT service class. The first slice, dedicated to the Ultra High Reliability Service (UHRS), is given the highest priority and is designed to support mission-critical applications, including use cases related to security, data protection, and e-health. By carefully managing the transmission parameters and optimizing resource allocation, we can improve the performance, reliability, and energy efficiency of these vital services. The second slice, corresponding to the High-Reliability Service (HRS), has medium priority and serves applications that require consistent data transmission, such as environmental monitoring and industrial control systems. However, it does not have the same stringent reliability demands as the UHRS. The third slice, dedicated to the Best Effort Service (BES), is assigned the lowest priority and is reserved for applications such as scale readings. This service is suitable for less critical applications where some delays or data loss are allowed.

3.2. Network model

In this study, we focus on the uplink scenario with a single LoRa gateway and a set of LoRa nodes, denoted as $\mathbb{N} = \{1, \dots, N\}$, which are randomly positioned around the LoRa gateway g. The network architecture follows the standard LoRaWAN star topology, where nodes transmit directly to the gateway to minimize energy consumption. The single gateway setup is modeled for the cost-constrained IoT cells. To achieve efficient resource allocation and maintain isolation, each service $s \in \mathbb{S}$ contains a set of LoRa nodes, referred to as $\mathbb{N}_s$. It is essential to note that every LoRa node $n \in \mathbb{N}$ is assigned to a specific network service.

$$\forall k, l \in \mathbb{S}, \mathbb{N}_k \cap \mathbb{N}_l = \emptyset \qquad \mathbb{N} = \bigcup_{k \in \mathbb{S}} \mathbb{N}_k \tag{1}$$

The services are virtually integrated over the LoRa gateway. This LoRa gateway g has its physical resources, including a set of $\mathbb{C} = \{1, \dots, C\}$ channels, which are virtually divided and reserved for the various services. Each service $s \in \mathbb{S}$ has a specific number of resource channels, denoted as $\mathbb{C}_s$, which represents the radio resources available for that particular service. Noting that a given channel $c \in \mathbb{C}$ is specifically reserved for one and only one service.

$$\forall c, d \in \mathbb{S}, \mathbb{C}_c \cap \mathbb{C}_d = \emptyset \qquad \mathbb{C} = \bigcup_{s \in \mathbb{S}} \mathbb{C}_s \tag{2}$$

3.3. Communication model

The model used to describe the path loss between the LoRa nodes and the LoRa gateway is based on the log-distance path model with shadowing (Bor et al., 2016), as follows:

$$P_L(d) = \overline{P_L}\left(d_0\right) + 10n \log\left(\frac{d}{d_0}\right) + X_\sigma \tag{3}$$

where $P_L(d)$ represents the path loss at the distance d measured in decibels [dB], while $\overline{P_L}\left(d_0\right)$ indicates the mean path loss at a reference distance d_0. The variable n denotes the path-loss factor, and X_σ follows

Table 2
Required signal-to-noise ratio (SNR) of the LoRa demodulator.

SF	Required SNR (dB)
12	−20
11	−17.5
10	−15
9	−12.5
8	−10
7	−7.5

a normal distribution $N\left(0,\sigma^2\right)$, with zero-mean and a variance of σ^2 to account for shadowing.

The received signal power, denoted as $P_{\mathrm{rx},n}$, for the LoRa node n at the gateway, can be defined as follows.

$$P_{\mathrm{rx},n} = P_{\mathrm{tx},n} + G - L - P_L(d) \tag{4}$$

whereby, $P_{tx,n}$ represents the transmitted power in dB, of the LoRa node, G denotes the antenna power gains, assuming the use of omni-directional antennas for both the transmitter and receiver. L indicates the power losses at the transmitter, and $P_L(d)$ refers to the path loss.

Definition 1. In wireless networks, receiver sensitivity is given by the following equation (Liu and Ball, 2021):

$$RXS = 10\log_{10}(kTB) + NF + S_r \tag{5}$$

In this logarithmic equation, k is Boltzmann's constant, T is the temperature in Kelvin, B represents the bandwidth, and NF is the receiver noise figure, while S_r is the required signal-to-noise ratio (SNR) necessary for successful demodulation.

This equation describes the general receiver sensitivity in wireless networks, which defines the minimum signal power required by the receiver to successfully demodulate a signal. However, in certain wireless technologies like LoRaWAN, additional factors specific to the modulation scheme influence the sensitivity.

LoRaWAN operates on the principle of CSS modulation, which allows communication over long distances and under conditions of very low signal-to-noise ratio. For example, at $SF12$, the required SNR is approximately −20 dB, which translates into a very high sensitivity whereas $SF7$ requires around −7.5 dB.

The equation for LoRaWAN sensitivity is similar to the general wireless network sensitivity but incorporates specific parameters for LoRa's spread spectrum technology:

$$S_{\mathrm{LoRa}} = -174 + 10\log_{10}(B) + NF + SNR \tag{6}$$

Here, bandwidth B typically takes values of 125 kHz, 250 kHz, or 500 kHz, and SNR is the required signal-to-noise ratio for a given SF, as shown in Table 2.

To ensure successful communication, the received signal power P_{rx} must exceed the calculated receiver sensitivity S_{LoRa}. This requirement is particularly crucial in environments where signal degradation can occur due to obstacles, interference, or the distance from the gateway. In LoRaWAN networks, a major cause of packet loss is collisions, which happen when two or more nodes transmit simultaneously. In this study only co-SF interference is considered, as the impact of inter-SF interference is negligible due to the orthogonality of the spreading factors (SFs).

The performance of the network is influenced by various factors, including TP, SF and CR. Our goal is to dynamically adapt these parameters to ensure that the LoRaWAN system remains robust and efficient. This adaptability will facilitate reliable communication, even in varying environmental conditions, ultimately enhancing the overall effectiveness of IoT applications.

3.4. Performance metrics

Packet collisions are a significant source of data loss in LoRaWAN networks, negatively impacting performance when multiple LoRa nodes transmit simultaneously. Factors such as SF, TP, and CR influence the likelihood of these collisions. Effective management of these parameters is essential for optimizing network performance and enhancing communication reliability. Additionally, maintaining a high Packet Delivery Rate (PDR) and minimizing energy consumption are crucial for ensuring reliable communication and extending the battery life of devices. By focusing on these aspects, LoRaWAN can optimize its performance and efficiency across various services, which makes it particularly suitable for IoT applications.

3.4.1. Packet delivery rate model

To evaluate whether a packet has been successfully received, we use the PDR metric. This metric helps estimate the success rate of packet transmissions by evaluating how many packets are correctly received out of those sent. It provides a simple way to measure the reliability of communication in LoRa networks that should be maximized for each service on LoRa gateway. A higher PDR indicates better performance in terms of data transmission.

$$PDR_i = \left(\frac{N_r}{N_s}\right) \times 100 \tag{7}$$

$$PDR_s = \sum_{i\in\mathbb{N}_s} \alpha_i PDR_i, \quad \forall i \in \mathbb{N}_s \tag{8}$$

$$\max U^s_{REL} = PDR_s, \quad \forall s \in \mathbb{S} \tag{9}$$

Where α_i is a binary indicator variable representing the association of LoRa node i with service s. Specifically, $\alpha_i = 1$ if the device is currently active in the service, and $\alpha_i = 0$ otherwise. U^s_{REL} denotes the reliability utility of a slice $s \in \mathbb{S}$ that should be maximized, and N_s denotes the total number of packets sent by the LoRa node i associated with a particular service $s \in \mathbb{S}$, and N_r is the number of packets received at the gateway g. PDR_i represents the PDR of LoRa node $i \in \mathbb{N}_s$ and PDR_s represents the PDR of a service $s \in \mathbb{S}$.

To theoretically quantify the impact of CR on reliability, we adopt the packet delivery model proposed in Li et al. (2020). The theoretical Packet Delivery Ratio is defined as the product of the survival probabilities of the preamble (P_{pre}), the header (P_h), and the payload (P_p):

$$PDR_{theo} = P_{pre} \times P_h \times P_p \tag{10}$$

First, the Bit Error Probability (P_b) is derived from the Signal-to-Noise Ratio (SNR) and Spreading Factor (SF) in an AWGN channel as:

$$P_b = 0.5 \times Q\left(\sqrt{2^{SF+1}\cdot SNR} - \sqrt{1.386\cdot SF + 1.154}\right) \tag{11}$$

where Q(⋅) is the tail function of the standard normal distribution. The header always employs a fixed CR of 4/8, allowing for error correction. Its decoding probability depends on the header length L_h:

$$P_h = \left((1-P_b)^4 + 3(1-P_b)^7 P_b\right)^{\lceil L_h/4SF\rceil} \tag{12}$$

Crucially, the Payload Decoding Probability (P_p) varies according to the dynamic CR selected by our algorithm. The model distinguishes between two regimes:

$$P_p(SNR,SF) = \begin{cases} (1-P_b)^{\lceil L_p/SF\rceil}, & CR \in \{1,2\} \\ \left((1-P_b)^4 + 3(1-P_b)^{3+CR}P_b\right)^{\lceil L_p/4SF\rceil}, & CR \in \{3,4\} \end{cases} \tag{13}$$

where L_p is the payload length, and $CR \in \{1..4\}$ corresponds to rates $\{4/5, 4/6, 4/7, 4/8\}$. For lower rates ($CR \in \{1,2\}$), the system operates in a detection-only regime, In contrast, higher rates ($CR \in \{3,4\}$) activate a correction regime via Forward Error Correction (FEC). The second case of the equation explicitly incorporates an additional term, $3(1-P_b)^{3+CR}P_b$, which represents the probability gain provided by the Forward Error Correction (FEC) mechanism.

3.4.2. Energy efficiency model

Energy Efficiency is a crucial metric that measures the effectiveness of a LoRa node in consuming its energy during communication. It is calculated as the ratio of successfully transmitted packets to the energy consumed during those transmissions. Higher energy efficiency indicates that a device achieves more successful transmissions with less energy consumption.

$$EE_i = \frac{\phi_i}{EC_i} \tag{14}$$

$$EE_s = \sum_{i \in \mathbb{N}_s} \alpha_i EE_i, \quad \forall i \in \mathbb{N}_s \tag{15}$$

$$\max U^s_{EE} = EE_s, \quad \forall s \in \mathbb{S} \tag{16}$$

where U^s_{EE} denotes the utility of the energy efficiency (EE) of a slice $s \in \mathbb{S}$ that should be maximized, ϕ_i denote the throughput achieved by the LoRa node i. EC_i represents the energy consumption of the LoRa node $i \in \mathbb{N}_s$, which refers to the total energy consumed by the LoRa node i during data transmission. EE_s represents the energy efficiency of a service $s \in \mathbb{S}$. This value should be maximized for each service. As defined in Lima et al. (2021), the energy consumption EC_i is expressed as:

$$EC_i = \text{ToA}^i \times I \times V \times N^i_s, \tag{17}$$

whereby N^i_s is the total number of packets sent by the LoRa node $i \in \mathbb{N}$, V is the operating voltage, I is the drain current for each transmitted packet in amperes (A), and ToA_i denotes the Time on Air (ToA), which corresponds to the duration that a packet occupies the communication channel during transmission in seconds (s). Moreover, the ToA of a packet is calculated as the sum of the preamble duration and the payload duration (Semtech, 2025):

$$\text{ToA} = T_{\text{payload}} + T_{\text{preamble}} \tag{18}$$

where

$$T_{\text{preamble}} = \left(n_{\text{preamble}} + 4.25\right) \times T_{\text{sym}} \tag{19}$$

$$T_{\text{payload}} = n_{\text{payload}} \times T_{\text{sym}} \tag{20}$$

whereby $T_{\text{sym}} = \frac{2^{SF}}{BW}$. Also, n_{preamble} and n_{payload} are the number of preamble symbols and the number of payload symbols, respectively, with n_{payload} defined as:

$$\begin{aligned} n_{\text{payload}} = 8 + \max\Bigg(\Bigg\lceil \frac{1}{4(SF - 2DE)} \times ((8PL - 4SF \\ +28 + 16CRC - 20H) \times (CR+4)) \Bigg\rceil, 0 \Bigg) \end{aligned} \tag{21}$$

where n_{payload} depends on the number of payload bytes PL, and the transmission parameters SF and CR. Furthermore, n_{payload} also depends on the Boolean variables H (which indicates the use of an explicit header) and DE (which represents the low data rate optimization). Finally, CRC indicates whether the CRC field is included in the physical message. If the CRC field is absent, CRC is set to 0; otherwise, it is set to 1.

The explicit n_{payload} formula reveals the mechanical impact of the Coding Rate on energy consumption. The term $(\mathbf{CR}+\mathbf{4})$ represents the total number of encoded bits transmitted for every 4 bits of raw data, where $CR \in \{1,2,3,4\}$. This creates a direct multiplier effect on the packet duration, for example, Coding Rate 4/5 ($CR = 1$): The redundancy factor is $(1+4) = 5$. This means 5 bits are transmitted for every 4 data bits (minimal overhead). However, the redundancy factor increase to 8 for using the Coding Rate 4/8 ($CR = 4$), meaning that increasing in the coded payload length. Since the Energy Consumption (EC_i) is a linear function of the Time-on-Air ($EC_i \propto ToA_i$), increasing the CR to maximize reliability mechanically increases the energy cost of the payload transmission.

Therefore, we define the throughput and delay for each LoRa node i assigned to service s on LoRa gateway g as:

$$\begin{aligned} \phi_i &= SF \cdot \frac{R_c}{2^{SF}} \cdot CR, \forall i \in \mathbb{N}_s, \\ d_i &= \frac{L_i}{\phi_i}, \forall i \in \mathbb{N}_s, \end{aligned} \tag{22}$$

where ϕ_i and d_i denote the throughput, and the delay achieved by the LoRa node i, and R_c denote the chip rate, CR denote the coding rate, and L_i the packet size.

3.5. Problem formulation

Our primary objective is to maximize the network's PDR and the energy efficiency, while ensuring the SLA for each individual service, by carefully selecting the appropriate transmission parameters. In this context, the SLA is defined by the PDR targets for each service, representing the desired percentage of successfully delivered packets. By meeting these objectives, we aim to achieve a balance between reliability and energy efficiency, ensuring optimal and high-quality performance across all services. Therefore, the multi-objective optimization for the resources allocation problem is formulated as follows:

$$\begin{aligned} \max U^s_{\text{total}} = \sum_s \left(w_{s(REL)} \cdot U^s_{REL} + w_{s(EE)} \cdot U^s_{EE} \right) \\ - \sum_s \max\left(0, PDR^{\text{target}}_s - U^s_{REL}\right), \forall s \in \mathbb{S}. \end{aligned} \tag{23}$$

subject to the following constraints:

$$\sum \alpha_i = 1, \quad \forall i \in \mathbb{N}_s, \forall s \in \mathbb{S} \tag{24a}$$

$$N_s \cap N_{s'} = \emptyset, \quad \forall i \in \mathbb{N}_s, \forall s, s' \in \mathbb{S} \tag{24b}$$

$$PDR_s \geq PDR^{\text{target}}_s, \quad \forall s \in \mathbb{S} \tag{24c}$$

$$EE_s \geq EE_{s-1}, \quad \forall s \in \mathbb{S}, \tag{24d}$$

Where $w_{s(REL)}$ represents the weight for different services regarding reliability, and $w_{s(EE)}$ represents the weight for different services regarding energy efficiency. The Constraint (24a) ensures that every LoRa node is allocated exactly to one and only one service. The Constraint (24b) plays a crucial role in ensuring the successful coalition formation for the association of LoRa nodes into services. In particular, constraint (24b) ensures that LoRa nodes assigned to slice $s \in \mathbb{S}$ are not the same as those in the service $s' \in \mathbb{S}$ within the coverage area of gateway g. Constraint (24c) ensures that the PDR of a service meets or exceeds the target threshold defined for that service, thus ensuring the SLA. Finally, constraint (24d) ensures that the energy efficiency for each higher-priority service s should still be greater than lower-priority service $s-1$ at the gateway g.

4. Proposed solution

In our proposed resource allocation scheme, we present three solutions to address the complexities of resource allocation in LoRaWAN network slicing. The first solution uses DQN independently to allocate SF, TP, and CR based solely on its learning from real-time interactions within the network. By adapting its strategies based on the observed outcomes, the DQN aims to refine and enhance the configurations to achieve the desired performance metrics. The second solution utilizes PSO, an effective optimization technique that explores the parameter space to find optimal transmission configurations for each LoRa node. By simulating the social behavior of swarms, PSO allows for efficient evaluation of critical transmission parameters of SF, TP, and CR. Its ability to quickly converge on effective solutions with minimal computational overhead makes PSO particularly valuable, enhancing both energy efficiency and reliability in resource allocation for the network. The third solution is a hybrid PSO–DQN approach, designed to leverage the complementary strengths of both PSO and DQN to maximize the

PDR and enhance energy efficiency. The initial phase of this approach uses PSO to explore the extensive search space of possible combinations of critical transmission parameters, including (6 SF × 5 TP × 4 CR per LoRa node/slice). By leveraging the collective intelligence of the swarm, PSO identifies the most promising combinations that lead to optimal performance for each LoRa node. After identifying potential solutions through the PSO process, the second stage employs Deep Reinforcement Learning (DRL) to refine and enhance these configurations. The DRL model learns continuously, adapting its strategies based on the network performance outcomes derived from the PSO-generated configurations. This hybridization reduces the inefficient random exploration, slow convergence; it accelerates convergence and ultimately maximizes the PDR while enhancing energy efficiency. Overall, these three solutions provide flexible options to optimize resource allocation while ensuring high PDR and energy efficiency.

4.1. PSO-DQN resource allocation solution

In the proposed PSO–DQN solution, we utilize a hybrid approach that combines PSO and DQN to address resource allocation challenges in LoRaWAN networks. The primary goals are to maximize PDR and energy efficiency. Further details on the PSO and DQN algorithms are provided in Appendix A and Appendix B, respectively.

As presented in Algorithm 1, the first phase employs PSO to explore the search space of transmission parameters, including SF, TP, and CR. In the PSO process, each particle represents a candidate solution, where a configuration (i.e., CR, TP, and SF) is assigned to every LoRa node.

Algorithm 1 PSO–DQN Resource Allocation

Input: I: Iterations, P: Particles, E: Episodes, T: Time steps, N: LoRa nodes, ϵ: Exploration–exploitation, ω, c_1, c_2: PSO params, α: Learning rate, γ: Discount factor

Output: Optimal parameters for each node n

```
PSO Phase:
Initialize positions $X_p^n$ and velocities $V_p^n$ for all $p, n$.          $\mathcal{O}(P \cdot N)$
for $i = 1$ to $I$ do
  for $p = 1$ to $P$ do
    Simulate network using particles configurations. $\mathcal{O}(C_{\text{sim}})$
    Get $PDR_n$ and $EE_n$ from simulation output.
    for $n = 1$ to $N$ do
      Compute fitness F = $f(\text{PDR}_n, \text{EE}_n)$. $\mathcal{O}(1)$
    end for
    Update $V_p^n, X_p^n$, and bests $P^n_{p,p_{best}}, P^n_{g_{best}}$. $\mathcal{O}(N)$
  end for
end for                                                                        $\mathcal{O}(I \cdot P \cdot C_{\text{sim}})$
for $n = 1$ to $N$ do
  Sort particles by fitness. $\mathcal{O}(PlogP)$
  Select top $K$ particles as $P^n_{top}$ $\mathcal{O}(K)$
end for                                                Top-K extraction: $\mathcal{O}(N.PlogP)$
DQN Phase:
Initialize DQN and replay buffer. $\mathcal{O}(1)$
for each episode $e = 1$ in $E$ do
  Initialize state $S_0$. $\mathcal{O}(1)$
  for each time step $t = 1$ in $T$ do
    for each LoRa node $n = 1$ to $N$ do
      Select action $a_t^n = (SF_n, TP_n, CR_n)$ from $P^n_{top}$ with $\epsilon$-greedy
        policy and noise $\mathcal{O}(|\theta| + K)$.
      Apply action $a_t^n$ to LoRa node $n$. $\mathcal{O}(1)$
    end for
    observe $R_t$, and store $(S_t, a_t, R_t, S_{t+1})$. $\mathcal{O}(1)$
    Train DQN and update $Q(S_t, a_t)$. $\mathcal{O}(|\theta|)$
  end for
end for                                                                   $\mathcal{O}(E \cdot T \cdot N \cdot |\theta|)$.
```

The process starts with an initialization phase, in which each particle is assigned a random position within the search space, and its initial velocity is set to zero. Additionally, each particle's personal best position is initialized to its current position, and the global best position is determined based on the best personal position among all particles. After this setup, the algorithm proceeds into a loop in which, at each iteration, for every particle, a network simulation is performed using the current configurations of all LoRa nodes proposed by the particle. The simulation outputs include the Packet Delivery Ratio PDR_n and the energy efficiency EE_n for each LoRa node, which are then used to compute the fitness value individually for each LoRa node as: $F = w_s \cdot \text{PDR}_n + w_s \cdot \text{EE}_n - \max(0, \text{PDR}_s^{\text{target}} - \text{PDR}_n)$, where w_s represents the weight corresponding to the service s to which the LoRa node belongs. This formulation ensures that the fitness function respects the SLA and enforces priority between services. The particle then updates its personal best for each LoRa node $P^n_{p,p_{best}}$ if it finds a better configuration than previously found. Similarly, the global best $P^n_{g_{best}}$ is also updated, if this new configuration is better than any found by the swarm for that LoRa node. The positions and velocities are also updated according to Eqs. (A.1) and (A.2), respectively, which are influenced by three key components: an inertia component, which retains some of the particle's previous velocity; a cognitive component, which directs the particle towards its personal best-known configuration; and a social component, which guides the particle towards the global best-known configuration identified by the swarm. This collaborative process allows particles to improve their configurations by learning both from their own experiences and from the successes of other particles in the swarm, enabling efficient optimization of LoRa node transmission parameters.

After completing all iterations, the top K = 10 configurations with the highest fitness scores are selected for each LoRa node, ensuring that the best-performing parameter sets are retained individually for every LoRa node in the network.

Since PSO has already identified the most promising parameter combinations for each LoRa node, in the second phase, we use DRL to refine these identified configurations. Specifically, the DQN uses the optimal parameter combinations identified by PSO for each LoRa node as a starting point to further refine the solutions. These parameter combinations are treated as the actions that the DQN should evaluate and improve. The DQN operates in a dynamic environment, learning to enhance its decisions through continuous interactions and feedback. The DQN framework can be seen in Fig. 1. By focusing its learning on the more refined and relevant set of actions provided by PSO, the DRL model avoids the inefficiencies of navigating through an expansive and often irrelevant action space. This enables the DQN to adjust the parameters in real-time, based on performance metrics, facilitating continuous improvement.

Consequently, this hybrid architecture establishes a robust hierarchy for handling network dynamics. The DQN agent learns a state-conditioned switching policy, allowing it to dynamically select the most appropriate configuration from the existing Top-K pool to match the instantaneous network state for standard network variations such as traffic fluctuations, temporary peaks, and moderate load changes. However, in scenarios where there is a massive traffic of new devices connected, or in the case of the creation of new slices, or the infrastructure change (e.g., gateway displacement), or when the SLA changes, the original Top-K configurations provided by PSO are not suitable; the PSO in our system is re-executed to generate a fresh set of Top-K configurations adapted to the new network regime.

One of the key advantages of employing PSO in this context is its ability to significantly reduce the time required to explore the parameter space compared to using DQN alone. While DQN can effectively learn optimal strategies through trial and error, it often requires extensive exploration of the action space, which can be time-consuming and computationally expensive. In contrast, PSO efficiently narrows down the search by quickly converging on the most promising parameter combinations, thereby minimizing the overall exploration time. This is particularly beneficial in scenarios where rapid decision-making is crucial, as it allows faster adaptations to changing network conditions.

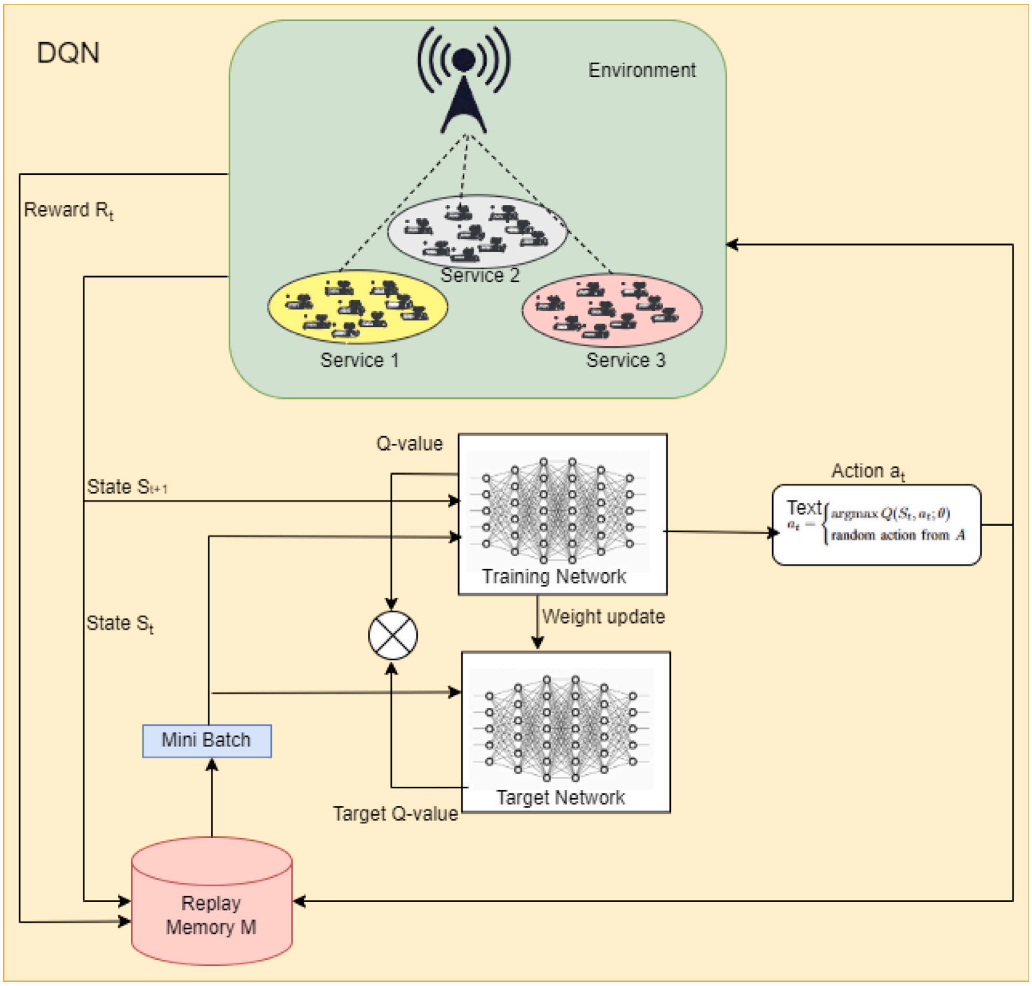


Fig. 1. DQN architecture.

Moreover, utilizing PSO, we effectively minimize the action space that the DRL model needs to explore. Since PSO has already identified the most promising parameter combinations for each LoRa node, the DQN model can focus its learning efforts on a more refined set of actions, rather than navigating through a vast and potentially irrelevant action space. This targeted approach not only accelerates the learning process, but also enhances the overall efficiency of resource allocation. This collaboration leads to a more efficient and adaptive allocation of resources, significantly enhancing overall network performance. Consequently, our framework provides a robust solution for optimizing resource allocation in communication networks, contributing to improved reliability, energy efficiency, and user satisfaction in data transmission.

As training progresses, the DQN agent becomes better at finding the best resource allocations that maximize service reliability and energy efficiency. By using the parameter combinations pre-selected by PSO, the DQN can focus only on the most relevant options. This reduces the complexity of the learning process, allowing the agent to learn faster and achieve the best results more efficiently.

4.2. DQN resource allocation model

To effectively optimize the performance of LoRaWAN network Slicing, it is crucial to define the key components that will guide the agent's DQN decision-making process. In this framework, we consider a single gateway as an intelligent agent. By defining the state, action, and reward elements precisely, we enable the agent to assign the most suitable transmission parameters to each LoRa node, enhancing both PDR and EE across the network. We define the state, action, and reward as follows:

Action: The primary function of the agent is to assign optimal transmission parameters SF, TP, and CR to the LoRa node within the LoRaWAN network slicing.

The action space, denoted as A, is defined by the discrete set of all possible parameter combinations:

$$A = \{(SF_i, TP_j, CR_k) \mid SF_i \in SF, TP_j \in TP, CR_k \in CR\}$$

Here, SF, TP, and CR represent the sets of possible values for spreading factor, transmission power, and coding rate, respectively. For instance:

- Spreading Factor (SF): SF_7, SF_8, SF_9, SF_{10}, SF_{11}, SF_{12}.
- Transmission Power (TP): $TP_2, TP_5, TP_8, TP_{11}, TP_{14}$.
- Coding Rate (CR): $CR_{4/5}, CR_{4/6}, CR_{4/7}, CR_{4/8}$.

In each episode, the agent selects an action $a \in A$ for each LoRa node, which is a tuple of the form:

$$a = (SF_i, TP_j, CR_k)$$

where i, j, and k are indexes in the respective sets. The goal is to optimize the overall network performance by choosing the best combination of *SF*, *TP*, and *CR* for each LoRa node.

State: In our framework, the state of each LoRa node n is defined by two critical performance metrics: *PDR* and the Energy Efficiency (*EE*).

We can represent the state of each LoRa node n as:

$$\mathrm{S}_n = (PDR_n, EE_n)$$

where:

- PDR_n represents the PDR for LoRa node n.
- EE_n denotes the energy efficiency metric for LoRa node n.

Reward: After allocating transmission parameters to the LoRa nodes by selecting the optimal actions, the environment provides a reward to the system. This reward guides the gateway agent in making decisions that achieve its objectives: maximizing both *PDR* and *EE* for each service while prioritizing critical services and ensuring priority among them. This can be represented as shown in (23), as follows:

$$R = \sum_s \left(w_{s(REL)} \cdot U^s_{REL} + w_{s(EE)} \cdot U^s_{EE} \right) - \sum_s \max\left(0, PDR_s^{\text{target}} - PDR_s\right), \forall s \in \mathbb{S}. \tag{25}$$

4.3. Computational complexity analysis

The complexity of the proposed approach PSO–DQN combines the complexity of both DQN and PSO.

The overall training complexity of the DQN is high (Zhang et al., 2023) and depends, directly, on the number of episodes E, the number of time steps considered T, the number of LoRa nodes N, and the execution of a forward and backward pass on the neural network constituting the DQN, which is represented by its parameters θ. Thus, the DQN complexity can be approximated as $\mathcal{O}(E \cdot T \cdot N \cdot |\theta|)$.

The complexity of PSO is determined by the number of particles P, the number of iterations I, and the simulation cost C_{sim} required to obtain performance metrics — namely PDR and energy efficiency (EE) — for each LoRa node. In each iteration, every particle represents a complete configuration assigned to all LoRa nodes, and the simulation outputs PDR and EE values individually for each node. Thus, the PSO complexity can be approximated as $\mathcal{O}(I \cdot P \cdot C_{\text{sim}})$.

To summarize, standalone DQN-3 can be computationally expensive to train due to its extensive exploration requirements. PSO offers a less complex learning mechanism but relies heavily on the cost of fitness evaluations. The PSO–DQN hybrid leverages PSO's exploratory strength to reduce the search space for DQN, potentially offering the best trade-off in terms of achieving high-quality solutions with manageable computational effort, especially in complex, dynamic environments.

5. Simulation results and analysis

In this section, we present the effectiveness of our proposed LoRaWAN network slicing architecture, which introduces three advanced solutions for the allocation of transmission parameters: DQN-3, PSO, and a PSO–DQN hybrid approach. To validate the robustness of our hybrid approach, we extended the comparative analysis to PSO–PPO(PSO–Proximal Policy Optimization). These solutions are designed to optimize the allocation of three key transmission parameters, such as spreading factors, transmission power, and coding rate based on the specific requirements of various services within the LoRa network. To demonstrate their effectiveness, we compared these approaches with a baseline method DQN-2 that allocates only two parameters, typically spreading factors and transmission power, aligning with the core structure of studies proposed in the related work.

To evaluate the performance of our proposed solutions, we used a model implemented in LoRaSim (Lancaster University, 2025), a discrete event simulator developed by Bor et al. using the SimPy library. LoRaSim is specifically designed to analyze scalability and collision issues in LoRa networks. The simulation scenario considers an uplink configuration where a LoRa gateway (GW) serves as the central communication hub, with LoRa nodes uniformly distributed across the gateway's coverage area. To support data transmission, the LoRa gateway is equipped with 8 receiving channels, each characterized by a bandwidth of 125 kHz within the standard 868 MHz European sub-band. This configuration ensures sufficient frequency diversity to accommodate multiple transmissions while minimizing collisions and enhancing scalability.

Table 3
Simulation parameters.

Parameters	Values
Spreading factors	[7 – 12]
Transmission power (dBm)	2, 5, 8, 11, 14
Coding rate	4/5, 4/6, 4/7, 4/8
Bandwidth (kHz)	125
Number of slices	3
Channels band (MHz)	European ISM Band : [863 – 870]
Voltage (V)	3
Epoch /Iteration Number	2000
Discount factor γ	0.99
Learning rate	0.003
Batch size	64

The resources of the LoRa gateway (channels) are shared among three services. Each channel is reserved for one service. The LoRa nodes are assigned to these three distinct services, with each LoRa node dedicated to only one specific service.

The simulations were replicated 10 times with a confidence interval of 95% to ensure statistical reliability. The model operated over a period of 10000 ms. A Poisson process, with an average packet transmission rate (λ) of 1×10^{-3} milliseconds (ms), was employed to model the traffic generation process. During the simulation, devices were assumed to upload small packet payloads of 20 Bytes. Each LoRa node initiated its communication with an initial transmission of packets to the gateway. Following this initial transmission, the LoRa nodes performed a single retransmission of their packets without receiving an acknowledgment (ACK). This approach simulates a scenario in which LoRa node continuously attempt to send their data.

The DQN was implemented and trained using Python 3.9 with the PyTorch framework. It consists of two fully connected hidden layers, each with 256 neurons, along with input and output layers. The Rectified Linear Unit (ReLU) was used as the activation function, and the Adam optimizer (Kingma and Ba, 2015) was employed with a learning rate (β). However, the PSO algorithm relies on several key parameters to optimize solutions effectively. These include a population size of 300 particles and a maximum number of 2000 iterations for convergence. The cognitive and social coefficients (c_1 and c_2) are set to 1.5, guiding the particles' movements based on their own best position and the swarm's global best. The inertia weight (ω) is set to 0.5, helping to balance exploration and exploitation during the optimization process. Table 3 represents the parameters of the LoRa simulation environment.

5.1. Average cumulative reward evaluation

Fig. 2 illustrates the average Cumulative rewards over episodes for four methods: PSO–DQN, PSO, DQN-3, and DQN-2.

The PSO–DQN method achieves significantly higher rewards compared to the other methods, demonstrating its superior effectiveness in optimizing transmission parameters. These results highlight the advantages of integrating PSO with DQN to improve solution quality in this context. While PSO alone performs well, it slightly underperforms compared to the hybrid PSO–DQN solution, indicating that although PSO is effective on its own, its performance is notably improved when integrated with DQN. The green curve, representing DQN-3, achieves lower rewards compared to both PSO and PSO–DQN. However, it incorporates the allocation of CR into its resource allocation strategy, which allows it to outperform DQN-2 by leveraging the redundancy provided

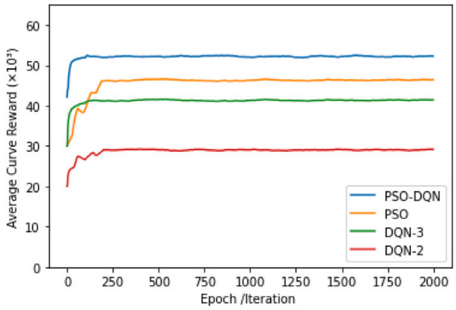


Fig. 2. Average cumulative reward evaluation.

by CR to improve metrics such as reliability and energy efficiency. Meanwhile, DQN-2, represented by the red curve, exhibits the lowest performance. This is because it does not utilize CR allocation, limiting its ability to effectively optimize key metrics.

5.2. Packet delivery rate analysis

Fig. 3 compares the PDR for three services using our hybrid approaches (PSO–DQN and PSO–PPO) and the standalone solutions (PSO and DQN-3), against the standard DQN model (DQN-2), which only allocates SF and TP. The comparative analysis has been extended to include the PSO–PPO framework, specifically focusing on the allocation of SF, TP, and CR.

The results show that Service 1 consistently maintains a PDR above its target of 90%. Similarly, Service 2's PDR consistently exceeds its target of 70%, while Service 3's PDR remains above its target of 50% for all solutions. This indicates that these solutions have effectively met the service level agreement (SLA) for these services. However, DQN-2 fails to meet the Service 1 target as the number of LoRa nodes increases, highlighting its limitations in meeting the SLA.

Among the various approaches evaluated, hybrid approaches stand out as the most effective, followed closely by PSO and then DQN-3. This hierarchy highlights the critical role of including the CR in resource allocation to enhance reliability, especially in dense networks. In contrast, DQN-2 experiences a significant decrease in performance as node density increases, illustrating its limitations in managing network congestion.

As shown in Fig. 3 PSO–DQN outperforms PSO–PPO for Service 1. However, for Services 2 and 3, both hybrid models exhibit comparable performance with negligible differences. While PSO provides the same high-quality initialization for both agents, the difference in final performance stems from the refinement phase. PPO is an on-policy algorithm, meaning it cannot reuse historical data from a replay buffer, leading to lower sample efficiency and longer training times, it requires significantly more resources due to its Actor–Critic architecture. In contrast, DQN, being a value-based method, operates with lower computational overhead. Consequently, PSO–DQN remains preferable from a system perspective, as it achieves the same result with much lower computational costs and resource requirements than PSO–PPO.

The hybrid approaches achieve a superior PDR compared to standalone methods. This performance stems from the synergy between PSO and reinforcement learning. In LoRaWAN, the search space defined by $\{SF, TP, CR\}$ is both vast and discrete. A standalone DRL agent typically starts 'blind,' frequently selecting parameter configurations that result in packet collisions during its exploration phase. This leads to poor initial rewards and slow, sub-optimal convergence. The hybrid approach addresses this critical issue by employing PSO as a global pre-selection mechanism. Leveraging its swarm intelligence, the PSO rapidly identifies near-optimal initial configurations, allowing the agent to start not from scratch, but from a stable and high-performing state. Consequently, the DQN can focus exclusively on fine-tuning and adapting to dynamic network changes, thereby ensuring high reliability (PDR) from the very first epochs.

In comparison, the PSO method also demonstrates strong performance by leveraging its global optimization capabilities to allocate SF, TP, and CR. Its efficiency in exploring the search space allows for rapid convergence on near-optimal solutions, making it particularly effective for resource allocation in dense network environments. DQN-3, on the other hand, relies solely on DRL for resource allocation. This limitation hinders its ability to thoroughly explore the search space for optimal configurations. Although DQN-3 allocates the CR alongside SF and TP, its performance is limited due to the absence of PSO's global optimization capabilities, provided by PSO in the PSO–DQN approach. Furthermore, DQN-3's learning-based approach takes longer to converge and may overlook some optimal configurations. Similarly, DQN-2 only allocates SF and TP without taking CR into account, limiting its flexibility and adaptability. CR is essential for enhancing communication reliability, particularly in crowded networks where interference and congestion are higher. By adjusting the CR, the system can control the amount of redundancy added to transmitted data, which directly affects the reliability of data delivery. Without CR, the DQN-2 solution has difficulty managing network congestion, resulting in a significant drop in performance as the number of LoRa node increases. In contrast, including CR in PSO–DQN, PSO, and DQN-3 improves their ability to adapt, leading to better throughput and higher PDR, even in dense network environments.

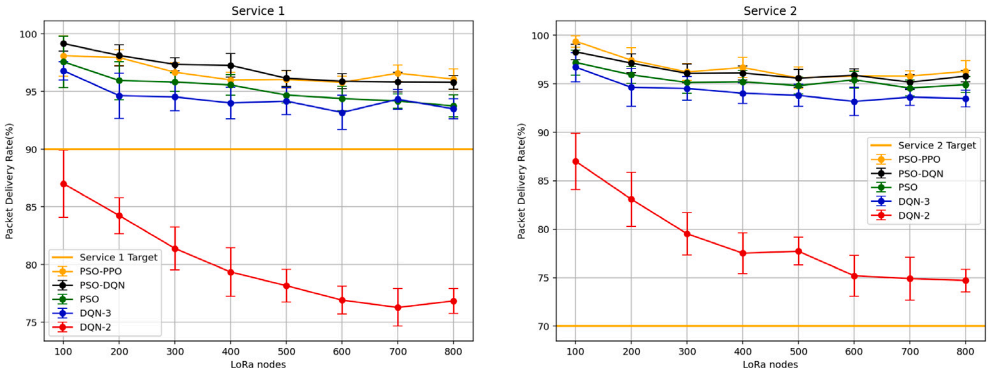
We observe in Fig. 3(a), Fig. 3(b), and Fig. 3(c) that the higher-priority services, such as Service 1, consistently experience higher PDR compared to lower-priority services like Service 2 and Service 3. This prioritization reflects the system's capability to meet SLAs by allocating resources in a way that satisfies the reliability constraints of critical services. As a result, Service 1 maintains the best PDR across all solutions, highlighting the system's ability to meet the needs of IoT applications.

5.3. Delay analysis

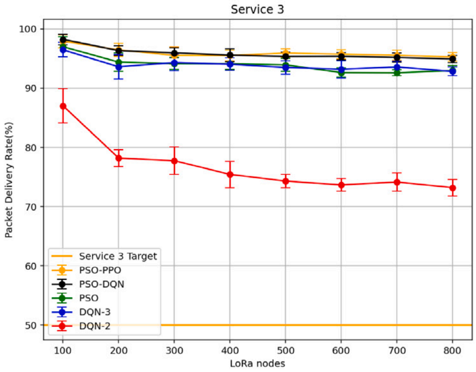
In this section, we analyze the mean delay of LoRa nodes in each service. The delay metric, as illustrated in Fig. 4, shows a significant increase as the number of LoRa nodes assigned to each service increases.

Among the approaches, PSO excels in minimizing delay by optimizing transmission parameters such as SF, TP, and CR. However, while PSO effectively reduces queuing delays, it may struggle to adapt to rapidly changing network conditions. This is where the hybrid approach of PSO–DQN shines, as it combines the global exploration capabilities of PSO with the dynamic adaptability of DQN, ensuring efficient communication even under dense network conditions. Since the PSO pre-selects configurations leading to high performance, the network avoids the retransmission cycles that typically cause delay. By providing these optimal initial settings, the PSO effectively shifts the DRL agent's starting point from a suboptimal region to a zone of high performance, allowing the reinforcement learning to focus solely on fine-tuning for dynamic network changes rather than basic configuration from scratch. Consequently, PSO–DQN effectively minimizes queuing and retransmission delays, ensuring efficient communication across all services.

To validate the robustness of our hybrid approach, we extended the comparative analysis to PSO–PPO with the allocation of 3 parameters (SF, TP, CR). The experimental results demonstrate that PSO–PPO and PSO–DQN yield comparable performance, suggesting that the PSO initialization effectively reduces the complexity of the solution space for both agents. However, their resource requirements differ significantly. PPO's on-policy nature and complex architecture lead to longer training times and higher resource consumption. In contrast, DQN is computationally lighter. Since LoRaWAN network slicing demands low-latency decision-making on constrained hardware (gateways), PSO–DQN is the



(a) UHRS Packet Delivery Rate (b) HRS Packet Delivery Rate



(c) BES Packet Delivery Rate

Fig. 3. Packet delivery rate evaluation.

more practical choice, offering the same high performance but at a fraction of the computational cost. However, DQN-3, which integrates CR alongside SF and TP, demonstrates improved performance over DQN-2 by adapting more effectively to network congestion and enhancing communication reliability. However, the lack of PSO's global optimization in DQN-3 leads to slightly higher delays, particularly as node density increases.

In contrast, DQN-2 experiences significant delays as node density increases. This is due to the inability to allocate CR, which limits its adaptability to network congestion and its capacity to manage redundancy. Without the flexibility to adjust redundancy levels, DQN-2 faces prolonged queuing delays, which severely impact its performance in dense scenarios. As observed in Fig. 4, UHRS consistently experiences lower delays compared to other services in the three frameworks, indicating that the algorithms effectively guarantee the SLA.

5.4. *Throughput analysis*

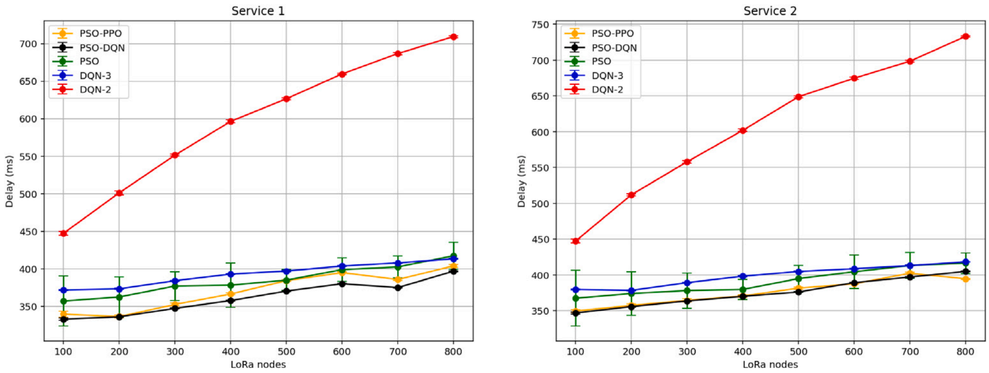
In this section, we evaluate the mean throughput of LoRa nodes for the services of the proposed solutions, as illustrated in Fig. 5. These results are compared to the throughput achieved using the standard DQN-2 approach. Moreover, to clearly demonstrate the robustness of our hybrid approach, we extended the comparative analysis to PSO–PPO with the allocation of 3 parameters (SF, TP, CR). The figures show that as the number of LoRa nodes deployed within each service increases, the throughput decreases. This decrease is primarily due to the increased congestion resulting from the higher density of LoRa nodes, which leads to greater competition for the available resources within each service.

Despite the challenges posed by this congestion, the services managed by the PSO–DQN, PSO–PPO, PSO, and DQN-3 frameworks consistently achieve higher throughput compared to the DQN-2 scheme, as demonstrated in Fig. 5. This notable improvement in performance is largely due to the allocation of the CR parameter employed in these frameworks, which enhances the ability to recover lost or corrupted packets.

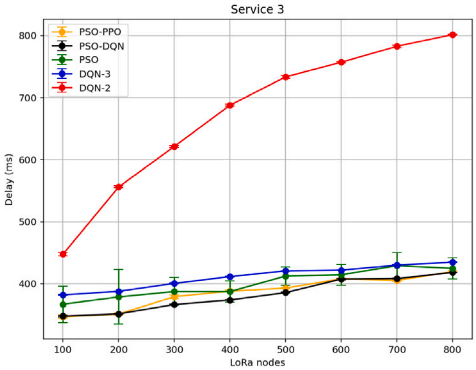
The experimental results demonstrate that hybrid approaches (PSO–PPO and PSO–DQN) show a significant improvement in the total amount of data successfully delivered. This gain is a direct consequence of the improved Packet Delivery Ratio (PDR). By intelligently distributing parameters, the hybrid approaches effectively mitigates congestion and reduces packet collisions. Unlike static allocation strategies, which suffer from the benefits of this hybridization.

Moreover, the PSO–DQN framework demonstrates superior throughput compared to standalone methods. While a standard DRL agent typically begins with an exploration phase, resulting in frequent collisions and high latency, the hybrid approach fundamentally alters this dynamic. The PSO acts as a global navigator, pre-selecting a safe and efficient starting configuration for the network. This allows the DQN agent to skip the initial trial-and-error phase and focus immediately on fine-tuning parameters to handle dynamic changes, such as interference. Consequently, the framework excels in its ability to dynamically allocate transmission parameters, minimizing interference and maximizing throughput even under increasing network density. Its adaptability enables it to respond effectively to varying network conditions, which is essential to maintain high performance in real-world scenarios.

Alternatively, the PSO framework, achieves remarkably high throughput levels, even in the presence of increased LoRa node density. This performance is mainly due to PSO's ability to dynamically optimize the transmission parameters of SF, TP, and CR. On the other hand, DQN-3 outperforms DQN-2, with a slight decrease in throughput due to the incorporation of error correction mechanisms provided by the coding rate. In contrast, DQN-2 suffers a significant performance



(a) UHRS Delay (b) HRS Delay



(c) BES Delay

Fig. 4. Delay evaluation.

degradation under congested conditions, primarily due to its lack of effective error correction. This limitation increases the probability of packet loss, further exacerbating the negative impact of congestion on throughput.

5.5. Energy consumption analysis

In this section, we evaluate the energy consumption (EC) of each service. As illustrated in Fig. 6, the EC metric is evaluated for the PSO–DQN, PSO, and DQN-3 approaches and compared to the standard DQN-2 method. The comparative analysis has been extended to include the PSO–PPO framework, specifically focusing on the allocation of SF, TP, and CR. Fig. 6(a), Fig. 6(b), and Fig. 6(c) clearly demonstrate that the EC for all services increases as the number of LoRa nodes increases. This is due to the higher network demands associated with larger LoRa nodes deployments.

The DQN-2 approach uses a fixed lower CR of 4/5 that helps to reduce energy consumption. However, this lack of flexibility in adjusting the CR parameter results in a lower reliability. On the other hand, the DQN-3 framework adjusts the CR parameter, as well as the SF and the TP. This flexibility enhances the network's reliability by using higher CRs when needed. However, this increase in reliability comes with higher energy costs, as the use of higher CRs usually leads to greater energy consumption. In contrast, the PSO approach results in lower energy consumption.

The experimental results demonstrate that the hybrid approaches (PSO–PPO and PSO–DQN) achieve a significant improvement in terms of energy consumption. However, PSO–DQN emerges as the more practical solution. While the PSO–PPO algorithm is highly robust, its Actor–Critic architecture and on-policy nature require intensive calculations, compared to the DQN.

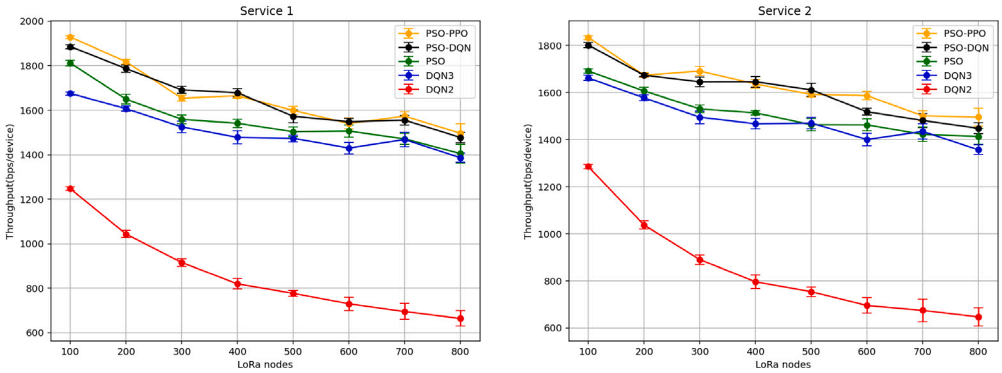
Additionally, the hybrid PSO–DQN approach stands out by integrating PSO with DQN. Since the standalone DRL agent tends to explore the entire action space, often selecting high TP or high SF or high CR unnecessarily during the learning process, leading to more energy consumption. The hybrid approach mitigates this by initializing the agent with the appropriate configuration parameters identified by PSO. As a result, the DRL agent starts from an optimized state, avoiding wasteful high-power transmissions and ensuring a significantly lower energy. This combination enables the framework to allocate optimal configurations for each LoRa node, achieving a balance between energy efficiency and reliability.

By adaptively allocating the SF, TP, and CR, the PSO–DQN approach ensures consistently low EC and high reliability. This performance surpasses both the PSO, DQN-3 and standard DQN-2 approaches in balancing these critical trade-offs. Consequently, PSO–DQN offers a robust and effective solution to the unique challenges of IoT slice-specific applications, enabling scalable, reliable, and energy-efficient network operations even with increasing LoRa nodes density.

5.6. The CR impact on system performance

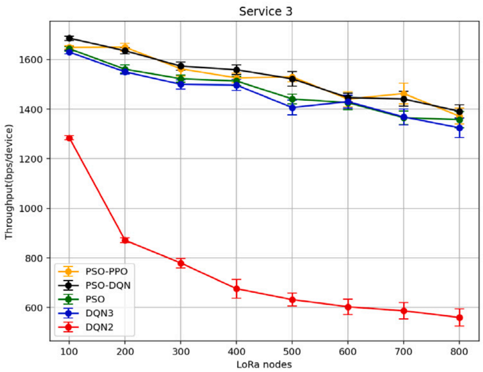
In this section, we investigate the impact of the CR on the overall system performance. We compare the proposed adaptive PSO–DQN approach (depicted by the black line) against a baseline PSO–DQN configuration with a fixed CR of 4/5 (green line). This comparison highlights the trade-offs between reliability and energy consumption.

Fig. 7 illustrates the impact of CR on PDR. It is observed that the PSO–DQN with adaptive CR allocation consistently outperforms the approach PSO–DQN with a fixed CR of 4/5. This improvement is attributed to the Forward Error Correction capability provided by the coding rate. By introducing redundancy bits as demonstrated in Section 3.4.1, the mechanism effectively recovers corrupted packets caused



(a) UHRS Throughput

(b) HRS Throughput



(c) BES Throughput

Fig. 5. Throughput evaluation.

by channel interference or collisions, thereby significantly enhancing overall network reliability. This stands in contrast to the PSO–DQN with fixed CR, which lacks the flexibility to adjust the CR parameter, resulting in lower reliability.

In terms of energy consumption, as shown in Fig. 8, the PSO–DQN approach with adaptive CR (black line) demonstrates superior performance, achieving slightly lower energy consumption than the fixed CR method. This indicates that our algorithm successfully balances the trade-off between reliability and energy. It identifies the optimal transmission parameters to maximize PDR while keeping energy costs to a minimum, ensuring robust communication without unnecessary power drain.

5.7. *Traffic variation analysis*

In this section, we evaluate the impact of traffic variation on each service of the proposed PSO–DQN solution for 300 LoRa nodes. In LoRaWAN networks, traffic variation is commonly modeled using the parameter lambda λ, which represents the average packet transmission rate.

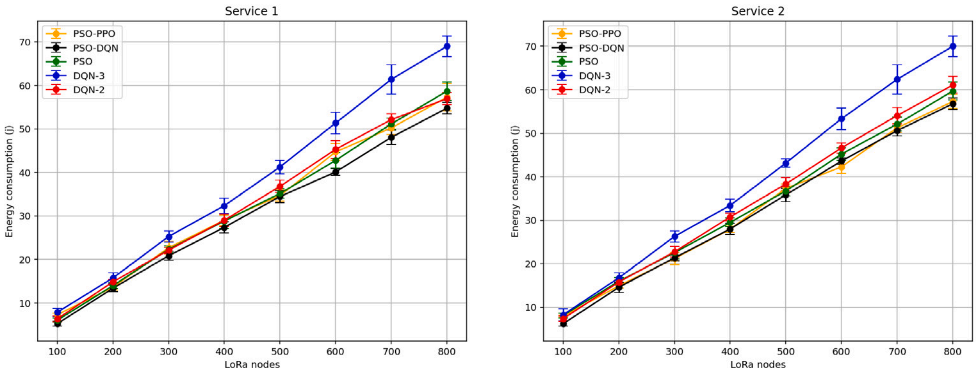
Specifically, as the value of λ increases, the average time between two consecutive packet transmissions decreases. In this scenario, LoRa nodes transmit more frequently, significantly increasing the probability of simultaneous transmissions, thereby also increasing the risk of packet collisions and overall network congestion. As a result, the wireless channel becomes more saturated, allowing for a degradation in the PDR, as illustrated in Fig. 9(a). However, the results show that the PDR for each service is maintained above its respective target: 90% for service 1, 70% for service 2, and 50% for service 3, indicating that the PSO–DQN effectively met the SLA for these services.

Moreover, as shown in Fig. 9(b), as λ increases, the overall delay of the three services also increases, as more packets are transmitted and require additional time for correct delivery. However, this can ultimately lead to a decrease in throughput due to the increased likelihood of collisions as shown in Fig. 9(c). This behavior also has a direct impact on energy consumption, as shown in Fig. 9(d), as the value of λ increases, LoRa nodes transmit more frequently, leading to higher energy consumption.

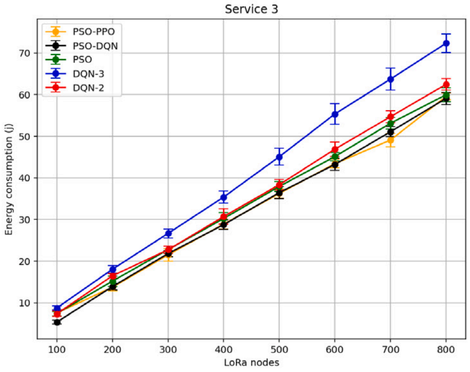
As traffic increases, the overall performance of the network decreases in terms of PDR, EC, delay, and throughput. Despite this, the PSO–DQN approach maintains strong performance even under heavy traffic, demonstrating the effectiveness of our hybrid solution in ensuring stable results. This highlights its robustness and ability to adapt to varying traffic conditions.

6. Conclusion

In this paper, we presented three approaches to optimize the allocation of transmission parameters in IoT LoRaWAN networks slicing. These solutions specifically address the critical challenges of balancing energy efficiency, reliability in resource-constrained environments. Specifically, we proposed three solutions in this work: DQN-3, PSO, and a hybrid approach PSO–DQN. Each solution is evaluated based on its ability to optimize transmission parameters such as SF, TP, and CR. The results for each solution are provided, demonstrating the performance improvements and trade-offs between the different approaches. In particular, the integration of PSO with DQN in the hybrid PSO–DQN approach represents a significant result, as it accelerates the convergence towards optimal configurations for transmission parameters. A key innovation of our work lies in the adaptive use of the CR, which provides a novel dimension to resource optimization. Unlike traditional methods, that use just the allocation of SF, and the TP, by incorporating CR alongside other transmission parameters, the proposed solutions enable a flexible resource allocation strategy. This not only improves



(a) UHRS Energy consumption (b) HRS Energy consumption

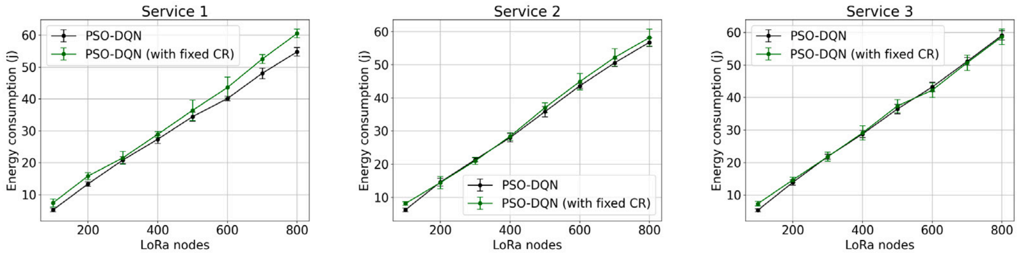


(c) BES Energy consumption

Fig. 6. Energy consumption evaluation.

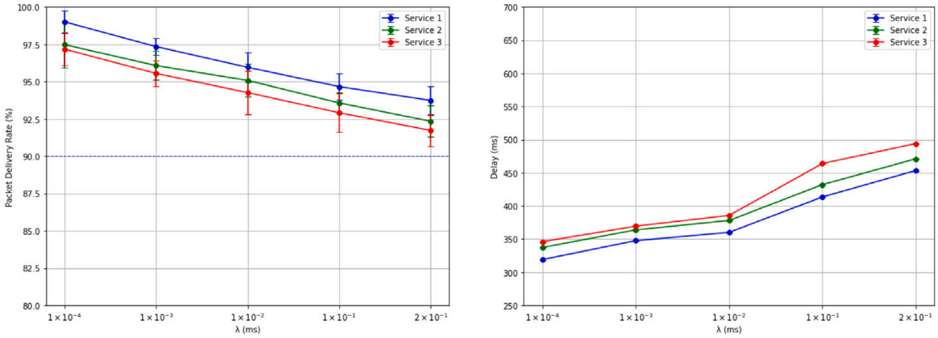
(a) UHRS Packet Delivery Rate (b) HRS Packet Delivery Rate (c) BES Packet Delivery Rate

Fig. 7. Packet delivery rate evaluation.

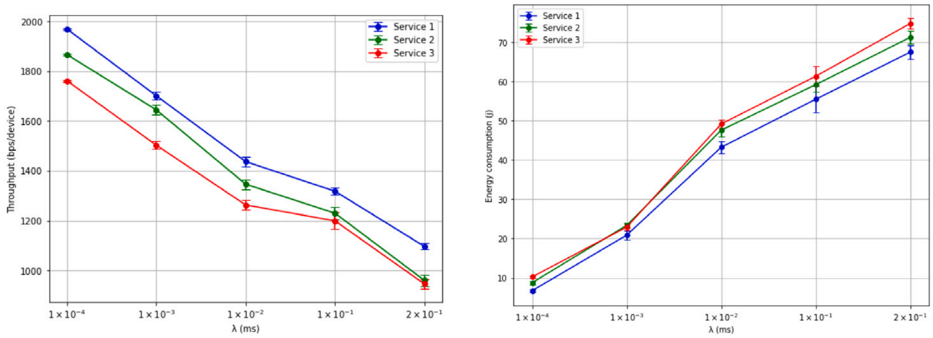


(a) UHRS Energy consumption (b) HRS Energy consumption (c) BES Energy consumption

Fig. 8. Energy consumption evaluation.



(a) Packet Delivery Rate (b) Delay



(c) Throughput (d) Energy Consumption

Fig. 9. Traffic variation evaluation.

the adaptability of the network to fluctuating demands but also ensures a robust quality of service for diverse IoT applications.

7. Limitations and future works

Despite the benefits of the proposed hybrid PSO–DQN resource allocation, and effective in ensuring convergence and SLAs in a high local density scenario (up to 800 LoRa nodes), several practical limitations must be acknowledged. In terms of scalability, like any model based DRL, our approach sees the size of state and action spaces grow with the number of slices and LoRa nodes. Although the results demonstrate efficiency for high local density, the use of a single centralized gateway can lead to saturation when traffic exceeds its capacity and limits robustness in the event of gateway failure. Moreover, the delay introduced by hybrid learning stems mainly from the PSO re-optimization and DQN re-training phases. In our model, the PSO is executed only at initialization to build a Top-K of solutions for each LoRa node. During normal network operation (routine traffic fluctuations, temporary peaks, or moderate load changes), PSO is not re-executed; the DQN simply selects in real time the best option from the existing Top-K, so no extra hybrid delay is introduced. However, a temporary delay only occurs in the event of major changes to the network (massive arrival of nodes, new slices, infrastructure changes like gateway changes or SLA modifications), requiring the PSO to be restarted and the DQN to be retrained.

Future work will investigate multi-gateway deployments aiming to improve network capacity and diversity. Additionally, we plan to examine multi-hop architectures to extend network coverage in shadowed areas, specifically by integrating relay nodes to bridge connectivity gaps in complex environments. Finally, a key focus will be on the implementation of federated learning to enable decentralized by distributing the computation across multiple gateways, each training the model locally on its own data, while the Network Server only aggregates the weights: this reduces saturation, improves fault tolerance, and decreases communication overhead while maintaining data privacy. By enabling parallel learning (multiple gateways), gateways can instantly adapt to local changes without waiting for global synchronization. Additionally, federated aggregation speeds up convergence, significantly reducing both the frequency of algorithm restarts and the temporary delay of the hybrid approach.

CRediT authorship contribution statement

Fatima Zahra Mardi: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Formal analysis, Conceptualization, Validation. **Yassine Hadjadj-Aoul:** Writing – review & editing, Supervision. **Miloud Bagaa:** Writing – review & editing, Supervision. **Nabil Benamar:** Writing – review & editing, Supervision.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work, the authors used ChatGPT in order to improve readability and correct grammatical errors. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

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.

Appendix A. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a swarm intelligence algorithm inspired by the social behavior of bird flocks and fish schools. It was originally proposed by Kennedy and Eberhart in 1995 (Eberhart and Shi, 2001). In PSO, a population known as a swarm consists of

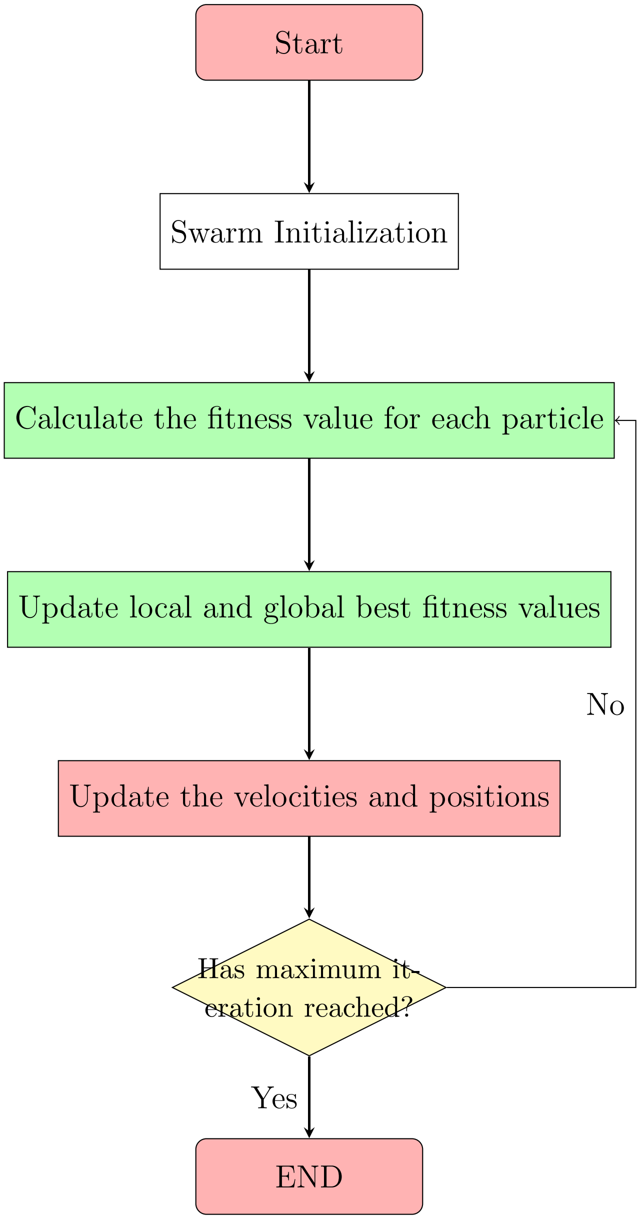


Fig. A.10. Flowchart of the particle swarm optimization.

individuals called particles, each representing a potential solution to the problem being addressed. As the search for optimal solutions progresses, each particle modifies its trajectory by considering both its own best-known position and the best position achieved by any particle in the swarm. This mechanism facilitates the exchange of information and experiences, allowing particles to leverage their individual discoveries (local best) as well as the collective experiences of their peers (global best) during the optimization process.

PSO is a powerful optimization algorithm that operates through a population of randomly initialized particles M, each representing a potential solution to a given problem. The algorithm initiates its search for the optimal solution by allowing these particles to traverse the search space, updating their positions iteratively until they converge on a relatively stable solution or reach a predetermined maximum number of iterations, denoted as T. In each iteration, the performance of the particles is evaluated on the basis of specific fitness criteria, which enables the algorithm to identify the best local position for each particle, as well as the best global position across the whole swarm. Each particle is envisioned as a point in N-dimensional space, where its state is characterized by two main features: position and velocity (Kennedy and Eberhart, 1995). The position of the ith particle at the tth iteration can be mathematically represented as, $X_i(t) = \left\{x_{i1}(t), x_{i2}(t), \dots, x_{iN}(t)\right\}$, indicating its coordinates in the N-dimensional space. Simultaneously, the velocity of the ith particle, which dictates how far and in what direction the particle will move in the next iteration, is expressed as $V_i(t) = \left\{v_{i1}(t), v_{i2}(t), \dots, v_{iN}(t)\right\}$.

The update mechanism for both position and velocity is governed by a set of equations that incorporate the particle's current velocity, its best-known position, and the best-known position of its neighbors. The velocity of each particle is defined by the equation (Shi and Eberhart, 1998):

$$V_i(t+1) = w \cdot V_i(t) + c_1 \cdot r_1 \cdot \left(P_{i,p_{best}} - X_i(t)\right) + c_2 \cdot r_2 \cdot \left(P_{g_{best}} - X_i(t)\right) \tag{A.1}$$

where $i = 1, 2, \dots, M$, $t = 1, 2, \dots, T$, w is the inertia weight, c_1 and c_2 are cognitive and social coefficients, r_1 and r_2 are random numbers in the range [0,1], $P_{i,p_{best}}$ is the local best position of the ith particle, $P_{g_{best}}$ is the global best position of the swarm, and $X_i(t)$ is the current position of the ith particle. Following the velocity update, the position of the ith particle is then updated using the equation (Shi and Eberhart, 1998):

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{A.2}$$

This iterative process not only facilitates exploration of the search space, but also enhances the convergence towards optimal solutions by balancing exploration and exploitation. As the algorithm progresses, the particles adjust their trajectories based on their own discoveries and the insights gained from the swarm, ultimately leading to improved performance in finding the best solution to complex optimization problems. Fig. A.10 illustrates the flowchart of the PSO process.

Appendix B. Deep Q-network (DQN)

The deep Q-learning algorithm typically consists of an agent utilizing a deep neural network (DNN) and an environment. The agent interacts with this environment and determines the actions to take. In our case, the LoRa Gateway acts as an agent, interacting with an environment to find the best combination of parameters that maximizes reliability and energy efficiency.

The proposed DNN is trained progressively with a set of training data per episode. Initially, the agent begins exploring the environment to gather information. At each time step t, the agent takes an action for its current state, either randomly or by using the action with the highest Q-value, as defined below:

$$a_t = \begin{cases} \text{argmax}\, Q(S_t, a_t; \theta) & \text{if } \epsilon < \epsilon_{th}, \text{ where } \epsilon_{th} \in (0,1) \\ \text{random action from } A & \text{otherwise.} \end{cases} \tag{B.1}$$

After taking the action according to the policy defined in (B.1), the agent receives an immediate reward R_t and observes the next state S_{t+1} from the environment.

To achieve more stable convergence towards the optimal policy, we incorporate a replay memory M into the deep Q-learning (DQL) framework. At each time step, when the DQN makes a decision, the agent's experience e_t is defined as a tuple:

$$e_t = (S_t, a_t, R_t, S_{t+1}),$$

where S_t and S_{t+1} represent the current and next states, a_t is the action taken, and R_t is the immediate reward. This experience is stored in replay memory M.

As the agent continues to interact with the environment and accumulates experiences in the replay memory M, it periodically samples mini-batches of experiences to update its deep Q-network.

To train the DNN, we calculate the loss function, which is typically defined as the mean squared error (MSE) between the predicted Q-values $Q(S_t, a_t; \theta)$ (where θ represents the parameters of the DNN) and the target Q-values y_t. The loss function can be expressed mathematically as:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} (y_t - Q(S_t, a_t; \theta))^2$$

Here, y_i is derived using the Bellman equation, defined as

$$y_t = R_t + \gamma \max_{a \in A} Q(S_{t+1}, a_{t+1}; \theta'),$$

where R_t is the immediate reward, γ is the discount factor, and S_{t+1} is the next state.

After calculating the loss function, the parameters of the DNN are optimized using back-propagation, which computes the gradients of the loss with respect to the network parameters. These gradients guide the weight updates, which is essential for learning.

The gradients are calculated as follows:

$$\nabla_\theta L(\theta) = \frac{\partial L(\theta)}{\partial \theta}.$$

Data availability

No data was used for the research described in the article.

References

Afolabi, I., Taleb, T., Samdanis, K., Ksentini, A., Flinck, H., 2018. Network slicing and softwarization: A survey on principles, enabling technologies, and solutions. IEEE Commun. Surv. Tutor. 20 (3), 2429–2453.

Alliance, L., 2015. A technical overview of LoRa and LoRaWAN. 20, White Paper, November.

Alwarafy, A., Abdallah, M., Ciftler, B.S., Al-Fuqaha, A., Hamdi, M., 2021. Deep reinforcement learning for radio resource allocation and management in next generation heterogeneous wireless networks: A survey. arXiv preprint arXiv:2106.00574.

Bor, M.C., Roedig, U., Voigt, T., Alonso, J.M., 2016. Do LoRa low-power wide-area networks scale? In: Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems. pp. 59–67.

Cheggour, S., Loscri, V., 2025. Frequency resource management in 6G user-centric CFmMIMO: A hybrid reinforcement learning and metaheuristic approach. arXiv preprint arXiv:2505.22443.

Dawaliby, S., Bradai, A., Pousset, Y., 2021. Joint slice-based spreading factor and transmission power optimization in LoRa smart city networks. Internet Things 14, 100121.

Eberhart, R.C., Shi, Y., 2001. Tracking and optimizing dynamic systems with particle swarms. In: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546), vol. 1, IEEE, pp. 94–100.

Fall, N.P., Marot, M., Diallo, C., Bernard, A., Roujanski, G., 2023. Optimizing mobility in LoRaWan: A resource reservation approach. In: GLOBECOM 2023-2023 IEEE Global Communications Conference. IEEE, pp. 7562–7567.

Farhan, L., Shukur, S.T., Alissa, A.E., Alrweg, M., Raza, U., Kharel, R., 2017. A survey on the challenges and opportunities of the internet of things (IoT). In: 2017 Eleventh International Conference on Sensing Technology. ICST, IEEE, pp. 1–5.

Gu, F., Niu, J., Jiang, L., Liu, X., Atiquzzaman, M., 2020. Survey of the low power wide area network technologies. J. Netw. Comput. Appl. 149, 102459.

Hamdi, R., Qaraqe, M., Althunibat, S., 2020. Dynamic spreading factor assignment in LoRa wireless networks. In: ICC 2020-2020 IEEE International Conference on Communications. ICC, IEEE, pp. 1–5.

Haxhibeqiri, J., De Poorter, E., Moerman, I., Hoebeke, J., 2018. A survey of LoRaWAN for IoT: From technology to application. Sensors 18 (11), 3995.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: Proceedings of ICNN'95-International Conference on Neural Networks, vol. 4, IEEE, pp. 1942–1948.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. ICLR, In: International Conference on Learning Representations, vol. 5, no. 6.

Lancaster University, 2025. LoRaSim: LoRa network simulator. https://www.lancaster.ac.uk/scc/sites/lora/lorasim.html. (Accessed 10 April 2025).

Li, Y., Yang, J., Wang, J., 2020. DyLoRa: Towards energy efficient dynamic LoRa transmission control. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications. IEEE, pp. 2312–2320.

Lima, E., Moraes, J., Oliveira, H., Cerqueira, E., Zeadally, S., Rosário, D., 2021. Adaptive priority-aware LoRaWAN resource allocation for internet of things applications. Ad Hoc Netw. 122, 102598.

Liu, Q., Ball, E.A., 2021. The meta distribution of the signal-to-interference ratio for long Range Wide Area networks with power control. IEEE Trans. Ind. Inform. 17 (4), 2579–2586. http://dx.doi.org/10.1109/TII.2020.3003474.

Mardi, F.Z., Bagaa, M., Hadjadj-Aoul, Y., Benamar, N., 2022. An efficient allocation system for centralized network slicing in lorawan. In: 2022 International Wireless Communications and Mobile Computing. IWCMC, IEEE, pp. 806–811.

Mardi, F.Z., Bagaa, M., Hadjadj-Aoul, Y., Benamar, N., 2023. Heuristic-deep Q-network-based network slicing in LoRaWAN. In: ICC 2023-IEEE International Conference on Communications. IEEE, pp. 4731–4736.

Mardi, F.Z., Hadjadj-Aoul, Y., Bagaa, M., Benamar, N., 2024. Resource allocation for LoRaWAN network slicing: Multi-armed bandit-based approaches. Internet Things 26, 101195.

Messaoud, S., Bradai, A., Ahmed, O.B., Quang, P.T.A., Atri, M., Hossain, M.S., 2020. Deep federated Q-learning-based network slicing for industrial IoT. IEEE Trans. Ind. Inform. 17 (8), 5572–5582.

Minhaj, S.U., Mahmood, A., Abedin, S.F., Hassan, S.A., Bhatti, M.T., Ali, S.H., Gidlund, M., 2023. Intelligent resource allocation in LoRaWAN using machine learning techniques. IEEE Access 11, 10092–10106.

Navas, R.E., Dandachi, G., Hadjadj-Aoul, Y., Maillé, P., 2023. Energy-aware spreading factor selection in LoRaWAN using delayed-feedback bandits. In: 2023 IFIP Networking Conference. IFIP Networking, IEEE, pp. 1–9.

Nguyen, T.T., Nguyen, H.H., Barton, R., Grossetete, P., 2019. Efficient design of chirp spread spectrum modulation for low-power wide-area networks. IEEE Internet Things J. 6 (6), 9503–9515.

Ossongo, E., Esseghir, M., Merghem-Boulahia, L., 2024. A multi-agent federated reinforcement learning-based optimization of quality of service in various LoRa network slices. Comput. Commun. 213, 320–330.

Semtech, 2025. SX1276 LoRa transceiver: Documentation (datasheet). https://www.semtech.com/products/wireless-rf/lora-connect/sx1276#documentation. (Accessed 17 June 2025).

Shi, Y., Eberhart, R.C., 1998. Parameter selection in particle swarm optimization. In: Evolutionary Programming VII: 7th International Conference, EP98 San Diego, California, USA, March 25–27, 1998 Proceedings 7. Springer, pp. 591–600.

Tahaei, H., Afifi, F., Asemi, A., Zaki, F., Anuar, N.B., 2020. The rise of traffic classification in IoT networks: A survey. J. Netw. Comput. Appl. 154, 102538.

Tang, J., 2025. Deep-reinforcement-learning–guided resource allocation and task offloading for 6G edge intelligence. Comput. Commun. 108364.

Tellache, A., Mekrache, A., Bradai, A., Boussaha, R., Pousset, Y., 2022. Deep reinforcement learning based resource allocation in dense sliced LoRaWAN networks. In: 2022 IEEE International Conference on Consumer Electronics. ICCE, IEEE, pp. 1–6.

Van den Abeele, F., Haxhibeqiri, J., Moerman, I., Hoebeke, J., 2017. Scalability analysis of large-scale LoRaWAN networks in ns-3. IEEE Internet Things J. 4 (6), 2186–2198.

Wang, Z., Shao, L., Yang, S., Wang, J., Li, D., 2023. CRLM: A cooperative model based on reinforcement learning and metaheuristic algorithms of routing protocols in wireless sensor networks. Comput. Netw. 236, 110019.

Zhang, S., Li, H., Wang, M., Liu, M., Chen, P.-Y., Lu, S., Liu, S., Murugesan, K., Chaudhury, S., 2023. On the convergence and sample complexity analysis of deep Q-networks with ϵ -greedy exploration. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23, Curran Associates Inc., Red Hook, NY, USA.