Resource Allocation for LoRaWAN Network Slicing: Multi-Armed Bandit-based Approaches

Fatima Zahra Mardi, Yassine Hadjadj-Aoul, Miloud Bagaa, Nabil Benamar

Resource Allocation for LoRaWAN Network Slicing: Multi-Armed Bandit-based Approaches

Fatima Zahra Mardi[1], Yassine Hadjadj-Aoul[2], Miloud Bagaa[3], and Nabil Benamar[1, 4]
[1]Moulay Ismail University, Meknes, Morocco.
[2]University of Rennes, Inria, IRISA, France.
[3] Université du Québec à Trois-Rivières, Trois-Rivières, QC, Canada.
[4]Al Akhawayn University in Ifrane, Morocco.
fa.mardi@edu.umi.ac.ma, yassine.hadjadj-aoul@irisa.fr, miloud.bagaa@uqtr.ca, n.benamar@umi.ac.ma

***Abstract*—Wireless sensor networks have become increasingly popular in recent years due to the growing demand for Internet of Things (IoT) applications, including LoRaWAN networks. However, the effective allocation of resources remains a crucial challenge in LoRaWAN networks due to the limited bandwidth and the diverse demands for multiple services. This paper presents three novel resource allocation solutions for LoRaWAN network slicing to address this challenge. These solutions are based on the Multi-Armed Bandit (MAB) algorithm, which is known for balancing the exploration of available actions with the exploitation of optimal decisions. Our objective is to dynamically and efficiently allocate resources to network slices by treating the resource allocation as a MAB problem. This approach aims to maximize Packet Delivery Rate (PDR) performance while ensuring each service's Service Level Agreement (SLA). The first solution, UCB-MAB, uses the Upper Confidence Bound (UCB) strategy to balance exploration and exploitation to improve network performance. The second solution, Q-UCB-MAB, continuously updates Q-values using the Q-learning update equation and incorporates the UCB strategy for further optimization. Finally, the third solution, ARIMA-UCB-MAB, leverages the predicted reward value from the Autoregressive Integrated Moving Average (ARIMA) model within the UCB framework to enhance network performance. Our results demonstrate that all three solutions offer efficient resource allocation in terms of PDR and SLA satisfaction. Specifically, the ARMA-UCB-MAB solution outperforms the other two solutions.**

***Index Terms*—Internet of Things, LoRaWAN network, network slicing, Multi-armed Bandit, Resource allocation.**

I. Introduction

Today, the rapid growth of connected devices has considerably complicated the task of network management. The development of the Internet of Things (IoT) has given rise to a wide range of devices, including smartphones, tablets, smart home appliances, and industrial sensors, all interconnected within networks. The exponential growth in connected devices represents a considerable challenge for network providers, as they must ensure continuous connectivity while maintaining optimal network performance. As a result, network operators are deploying advanced technologies to adapt to dynamic scenarios and meet the diverse needs of different applications while optimizing resource utilization.

Low-Power Wide Area Network (LPWAN) is a wireless communication network considered as one of the most promising emerging technologies for IoT applications [1]. As part of this class of networks, the LoRaWAN network provides an optimal communication framework for IoT devices [2]. Its notable features include long-range wireless,low-power consumption, and low deployment cost, facilitating efficient and reliable connectivity across a wide range of IoT applications. To incerease the number of connected nodes in a LoRaWan network, "Network slicing" can be considered as a suitable alternative in this context. Indeed, Network slicing is a technology that divides a physical network into several virtual slices to meet various application requirements [3]. The integration of this technology further enhances LoRaWAN's capabilities by enabling the virtual partitioning of resources in a highly efficient and isolated manner. Through this integration, LoRaWAN can effectively allocate radio resources to meet the specific needs of diverse IoT applications or network slices. This powerful combination empowers IoT networks to accommodate the growing influx of devices while satisfying their varied service requirements.

In the context of LoRaWAN network slicing, as the number of devices connected to the LoRaWAN network increases, data collisions become significant due to simultaneous transmission attempts [4]. This, in turn, leads to a degradation of the overall network performance, which consequently poses a significant challenge in guaranteeing the need for various slices, each with its specific requirements. To overcome this problem, it's necessaraly to design an efficient resource allocation scheme to dynamically allocate resources to each service according to changes in load. In this context, the Multi-Armed Bandit (MAB) algorithm emerges as a promising approach for real-time decision-making in such a dynamic environment [5]. Several methods utilizing Multi-Armed Bandit (MAB) algorithms have been used to improve the performance of various IoTnetworks. In [6], Zhu et al. introduce a novel fair access scheme for cognitive radio-based wireless sensor networks (CR-WSNs) in the context of the IoT. The technique utilizes channel grouping based on an online learning method known as modified UCB-K, a multi-armed bandit algorithm. The scheme aims to solve the channel selection problem in CR-WSNs where the channels' statistical information is completely unknown to cognitive users. The proposed method adopts distributed learning with fairness to avoid collision between cognitive users and embody fairness between them.

In [7], Pase et al. propose a Multi-Armed Bandit (MAB) approach for resource allocation in Ultra-Reliable Low-Latency Communication (URLLC) scenarios in Industrial Internet of Things (IIoT) networks. The MAB approach is effective in allocating resources in a distributed manner in periodic and aperiodic traffic scenarios, even in dense networks with aggressive traffic. Also, the authors in [8] proposed a mean-field multi-armed bandit algorithm, which dynamically allocates users to different small cells based on their energy levels and channel conditions to minimize interference and maximize energy efficiency.

In this paper, we investigate the potential of the MAB algorithm to address the resource allocation challenges of LoRaWAN network slicing by proposing three solutions. Our primary goal is to guarantee the demand for each service while optimizing network resources. By implementing the MAB algorithm, we explore various resource allocation options to identify the best one that significantly improve the overall network performance. For resource allocation, our approach is based on two crucial dimensions: time and frequency. This enables us to achieve highly efficient resource management adapted to each service. The main contributions of this paper are as follows:

- We propose the use of the Upper Confidence Bound - Multi-Armed Bandit (UCB-MAB) algorithm, which can balance the trade-off between the exploration of the available scenarios of resource allocations and the exploitation of the best one to optimize the overall network performance effectively.
- We present the Q-learning Upper Confidence Bound - Multi-Armed Bandit (Q-UCB-MAB) solution that merges Q-learning and UCB strategies by dynamically updating Q-values using the Q-learning equation and achieves a balance between exploration and exploitation through UCB.
- We present the ARIMA Upper Confidence Bound - Multi-Armed Bandit (ARIMA-UCB-MAB) solution, which merges ARIMA prediction with UCB strategies. This innovative method leverages ARIMA predictions into the UCB, enabling more accurate and well-informed decision-making, resulting in enhanced performance and efficiency in resource allocation.
- The analyses and simulation results are provided to validate the effectiveness of our proposed solutions and demonstrate their efficiency in terms of Packet Delivery Rate (PDR).

The rest of this paper is organized as follows. In Section II, we present the related work. Section III discusses the system model. Section IV elaborates on the proposed resource allocation solutions. Section V shows and discusses the obtained results of our proposed solutions. Finally, the conclusion is given in Section VI.

II. Related work

The LoRaWAN network has been the subject of extensive studies and discussions within the research community in recent years. Authors in [9] proposed a decentralized decision-making solution for IoT-LoRaWAN networks using the re-learning EXP3 Multi-Armed Bandit (MAB) algorithm with expert distribution advice. The approach aims to improve network performance by optimizing transmission parameters for successful packet transmission with optimized power consumption. Authors in [10] focused on multi-arm bandit-based channel allocation method for massive IoT systems such as LoRaWAN networks, where devices dynamically select the best available channel to avoid congestion and improve communication efficiency. However, this approach may not be sufficient for more complex LoRaWAN networks that use network slicing, where context-based resource management of each service is required to ensure optimal performance.

One of the significant challenges in LoRaWAN is to provide efficient support for constrained services. To address this issue, recent works have explored network slicing implementationin LoRaWAN. In [11], Dawaliby et al. examine the potential of network slicing in LoRa networks to meet the diverse service requirements of IoT applications and services. They propose a dynamic inter-slicing algorithm that prioritizes slices according to their QoS requirements and reserves bandwidth on LoRa gateways using maximum likelihood estimation (MLE), as well as an intra-slicing strategy that maximizes the resource allocation efficiency of LoRa slices concerning their delay requirements. In [12], Messaoud et al. propose an approach to address the communication challenges of Industry 4.0 by integrating LoRaWAN and software-defined networking. The system leverages Online Gaussian Mixture Model Clustering (OGMMC) and Mini-Batch Gradient Descent (MBGD) algorithms to allocate network resources to devices based on their Quality of Service (QoS) requirements. Meanwhile, in [13], Mardi et al. propose a network slicing strategy based on a centralized coalition game to manage LoRa nodes in the LoRaWAN network efficiently. The initial coalitions were formed using the K-Means clustering algorithm to move LoRa nodes from one coalition to another in a way that maximizes reliability while ensuring the appropriate SLA for each network slice. Nevertheless, these proposed solutions face challenges in adapting to the dynamic nature of real-world traffic patterns, which can change frequently. These fluctuations may lead to difficulties in managing evolving device traffic loads over time, particularly when faced with unexpected traffic increases, potentially resulting in decreased network performance.

This paper focuses on the network slicing approach for the LoRaWAN framework. Unlike previous solutions, this approach involves dividing channels into smaller time intervals, which allows for more efficient and effective allocation of resources. We propose three innovative solutions utilizing MAB algorithms to address resource allocation challenges across three different services. Our main objective is to maximize network performance while ensuring the Service Level Agreements (SLAs) of supported services. By integrating MAB algorithms into network slicing, we enable a dynamic and adaptive resource allocation, which efficiently adjusts to

the changing network conditions and varying demands of the services. This approach ensures a smart resource distribution, allowing the LoRaWAN network to optimize its overall network performance.

III. System Model

A. Network Model

Figure 1 shows the LoRaWAN network slicing architecture, which is composed of a set of slices, denoted by $\mathbb{S} = \{1, \cdots, s\}$, where each slice $s \in \mathbb{S}$ corresponds to a specific service or application. Network slicing optimizes each service individually according to its unique requirements. In this context, and without loss of generality, the considered network slicing architecture consists of three virtual slices. The first slice, known as the Ultra High Reliability Slice (UHRS), is the highest priority. This slice includes mission-critical services, such as IoT applications for security, data protection, and e-health. By allocating resources to UHRS, the network ensures the efficient performance of these essential services and improves their reliability and efficiency. The second slice is known as the High-Reliability Slice (HRS) and holds a medium priority among the slices. It plays a critical role in sensing applications. Lastly, the third slice is called the Best Effort Slice (BES), which is assigned the lowest priority. This slice is primarily used for scale-reading applications, among others. By dividing the network into different slices with varying priorities, it becomes possible to allocate resources efficiently according to the specific needs and importance of each slice.

The slices are virtually integrated on top of LoRa Gateways. For the sake of simplicity, we will focus on a single LoRa gateway, denoted as GW. As shown in Figure 1, the physical resource blocks (RBs) of the LoRa gateway GW are structured in a 2-dimensional grid consisting of a time dimension and a frequency dimension. The time dimension is defined by a sequence of time slots, denoted as $\mathbb{T} = \{1, \cdots, t\}$, while the frequency dimension is characterized by a set of resource channels, denoted as $\mathbb{C} = \{1, \cdots, c\}$.

We represent the RBs of the LoRa gateway GW as $\mathbb{B}$. While, each RB in the grid is denoted as b_{ij}, where $i \in \mathbb{T}$ and $j \in \mathbb{C}$. This organization enables efficient allocation and utilization of resources within the LoRa gateway.

$$\mathbb{B} = \bigcup_{i \in \mathbb{T}, j \in \mathbb{C}} \{b_{ij}\} \tag{1}$$

Also, the notion of a resource block is of significant importance in the proposed framework. The definition of a resource block in this context differs from its use in cellular networks. While the latter systems allocate a single resource block to each device, in our model, several devices can share a resource block. Moreover, the size of a resource block is also large enough to accommodate many transmissions, eliminating the need to allocate strict time slots or ensure exclusive access, as it is the case in Slotted-Aloha-based models. This feature makes it more suitable for LoRaWAN networks, which have very wide coverage and where devices might have synchronization problems.

We consider an uplink scenario, in which a set of LoRa nodes denoted as $\mathbb{N} = \{1, \cdots, n\}$ are randomly located around the LoRa gateway GW. To ensure efficient resource allocation and isolation, each service $s \in \mathbb{S}$ has its set of LoRa nodes denoted as $\mathbb{N}_s$, noting that each LoRa node $n \in \mathbb{N}$ belongs to one specific network slice, as follows:

$$\forall k, l \in \mathbb{S}, \mathbb{N}_k \cap \mathbb{N}_l = \emptyset. \tag{2}$$

Additionally, each slice $s \in \mathbb{S}$ has a certain amount of resource blocks designated by $\mathbb{B}_{s_{ij}}$, which represents the available radio resources for that specific service.

$$\forall i \in \mathbb{T}, j \in \mathbb{C} : \mathbb{B} \ = \ \cup_{s \in \mathbb{S}} \mathbb{B}_{s_{ij}} \tag{3}$$

We consider that a given block $b \in \mathbb{B}$ belongs to one and only one service, i.e.,

$$\forall m, \ n \in \mathbb{S}, \quad \mathbb{B}_m \cap \mathbb{B}_n = \emptyset. \tag{4}$$

B. Path Loss and Channel Model

In our analysis, we adopt the log-distance path model with shadowing to define the channel model as

$$P_L(d) = \overline{P_L}(d_0) + 10n \log\left(\frac{d}{d_0}\right) + X_\sigma \tag{5}$$

where $P_L(d)$ is the path loss at distance d in [dB], $\overline{P_L}(d_0)$ is the mean path loss at the reference distance d_0, n is the path-loss factor and $X_\sigma \sim N\left(0, \sigma^2\right)$, the normal distribution with zero mean and σ^2 variance to account for shadowing [14].

To achieve a successful transmission, the received signal power P_{rx} must exceed the sensitivity threshold S_{rx} of the receiver. The received signal power P_{rx} can be expressed as follows:

$$P_{rx} = P_{tx} + GL - P_L \tag{6}$$

where P_{tx} is the LoRa node's transmit power, GL combines all general gains and losses and P_L is the path loss. In the following, the sensitivity of the LoRa receiver (S) is described as follows [15] [14]:

$$S = -174 + 10\log_{10}(BW) + NF + SNR \tag{7}$$

where –174 dBm is the thermal noise computed for 1Hz of BW, *NF* is the noise figure of receiver and *SNR* is the signal to noise ratio.

C. Performance Metrics

The evaluation of network performance in our paper focuses on a critical factor: the Packet Delivery Rate (PDR).

The PDR, as formulated in equation (8), is a fundamental metric of network efficiency. It represents the ratio of successfully received packets to the total transmitted packets. A high PDR indicates reliable data transmission and effective communication within the network.

$$PDR_{b_{ij},s}(t) = (\frac{N_s - N_c}{N_s}) \times 100 \quad \forall s \in \mathbb{S} \; \forall t \in \mathbb{T} \tag{8}$$

whereby N_s denotes the packets transmitted by the LoRa nodes associated with a particular service $s \in \mathbb{S}$ within the designated block b_{ij}, while N_c is the number of packets that experience collision at the gateway.

D. Problem Formulation

Our paper aims to address the resource block allocation problem, specifically in allocating resource blocks (RBs) to various services at each time step. Our main objective is to optimize the network's PDR, which reflects the efficiency of packet delivery. However, we recognize that achieving a high PDR is not enough. To ensure overall performance and service quality, we take into account the Service Level Agreements (SLAs) for each individual service. In our case, the SLA is defined by the PDR targets for each service, representing the desired percentage of successfully delivered packets.

Let define a binary variable, $\mathcal{X}_{b_{ij},s,t}$, which represents the resource allocation status of the service, where $\mathcal{X}_{b_{ij},s,t} = 1$ indicates that service s is allocated to resource block b_{ij} at time step t, and $\mathcal{X}_{b_{ij},s,t} = 0$ otherwise.

Our goal is to optimize the network's utility function, which is related to the packet delivery rate (PDR) by effectively allocating resource blocks to various services while ensuring the SLA of each service. Then, the utility function of the network is defined as

$$\mathbf{U} = \sum_{s \in \mathcal{S}} w_s PDR_s(t) - Cost(t) \quad \forall t \in \mathbb{T} \tag{9}$$

Here, w_s represents the importance of each service. The cost function is defined as follows:

$$Cost(t) = \sum_{s \in \mathcal{S}} \max\left(0, \frac{PDR_s^{Target} - PDR_s(t)}{PDR_s^{Target}}\right) \tag{10}$$

In simpler terms, this equation calculates the cost by comparing the target Packet Delivery Ratio (PDR) PDR_s^{Target} for a service s with the actual PDR achieved $PDR_s(t)$. If the achieved PDR is lower than the target, a cost is incurred.

The PDR of each service $s \in \mathcal{S}$ can be calculated as follow:

$$PDR_s(t) = \sum_{b_{ij} \in \mathbb{B}} \mathcal{X}_{b_{ij},s,t} \times PDR_{b_{ij},s}(t), \forall s \in \mathbb{S} \; \forall t \in \mathbb{T} \tag{11}$$

In this context, the term $PDR_{b_{ij},s}(t)$ characterizes the PDR associated with a specific service within a designated block b_{ij} at time step t, and its definition can be found in equation 8. Therefore, the optimization problem can be formulated as follows:

$$\max \mathbf{U}$$

subject to

$$\forall b_{ij} \in \mathbb{B}, \forall t \in \mathbb{T} : \sum_{s \in \mathbb{S}} \mathcal{X}_{b_{ij},s,t} = 1 \tag{12a}$$

$$\forall s \in \mathcal{S}, \forall t \in \mathbb{T} : \sum_{b_{ij} \in \mathbb{B}} \mathcal{X}_{bij,s,t} \leq 1 \tag{12b}$$

$$\forall s \in \mathbb{S}, \forall t \in \mathbb{T} : PDR_s(t) \geq PDR_s^{target} \tag{12c}$$

Constraint (12a) ensures that each block must be assigned to one and only one service. In constraint (12b), the allocation of a resource block b_{ij} at time step t is limited to at most one service. The constraint (12c) guarantees that the PDR for each service is equal to or greater than its target PDR denoted by PDR_s^{target}, thus ensuring the SLA.

IV. Multi-Armed Bandits based Resources allocation

A. MAB Basics

Multi-armed bandits (MABs) are a type of algorithm in the field of reinforcement learning (RL) and sequential decision-making. In MABs, an agent must repeatedly choose one of several actions, known as arms, at each time step [16]. Each arm has an associated reward distribution, and the goal is to maximize the agent's cumulative reward over time by learning which arms are most likely to yield the highest reward. However, the agent must balance two competing strategies: exploration and exploitation. Exploration involves trying out different arms to learn more about their reward distributions, while exploitation involves choosing the arms that are currently estimated to have the highest expected rewards based on the agent's past experience [16].

B. MAB Resource Allocation

The Multi-Armed Bandit (MAB) framework [17] is a powerful paradigm used for resource allocation in various decision-making problems. In the context of our paper, we leverage the MAB algorithm to efficiently allocate resource blocks (RBs) to different services in a dynamic network environment. As the state of the network may change over time due to factors such as varying traffic levels, our use of the MAB framework provides an efficient approach to adapt to these fluctuations and ensure optimal resource allocation.

In the MAB model, we define a set of possible arms or scenarios, denoted as $\mathbb{A} = \{A_1, A_2, ...A_k\}$. At each time step t, an action, $A_k^t \in \mathbb{A}$, is chosen from the set $\mathbb{A}$, leading to a specific reward, $R_t(A_k^t)$. The reward function is associated with the Packet Delivery Rate (PDR) for each service, as detailed in equation (9).

In our resource allocation framework, we have developed a novel approach to defining each arm, where each arm is related to a scenario of resource allocation, We achieve this by defining each scenario (arm) as a combination of resource allocation percentages between different services. To increase flexibility, we divide each arm into n parts, each corresponding to a different percent of resource allocation for a specific service. Specifically, let $p_{k,i}$ denote the percentage of resources

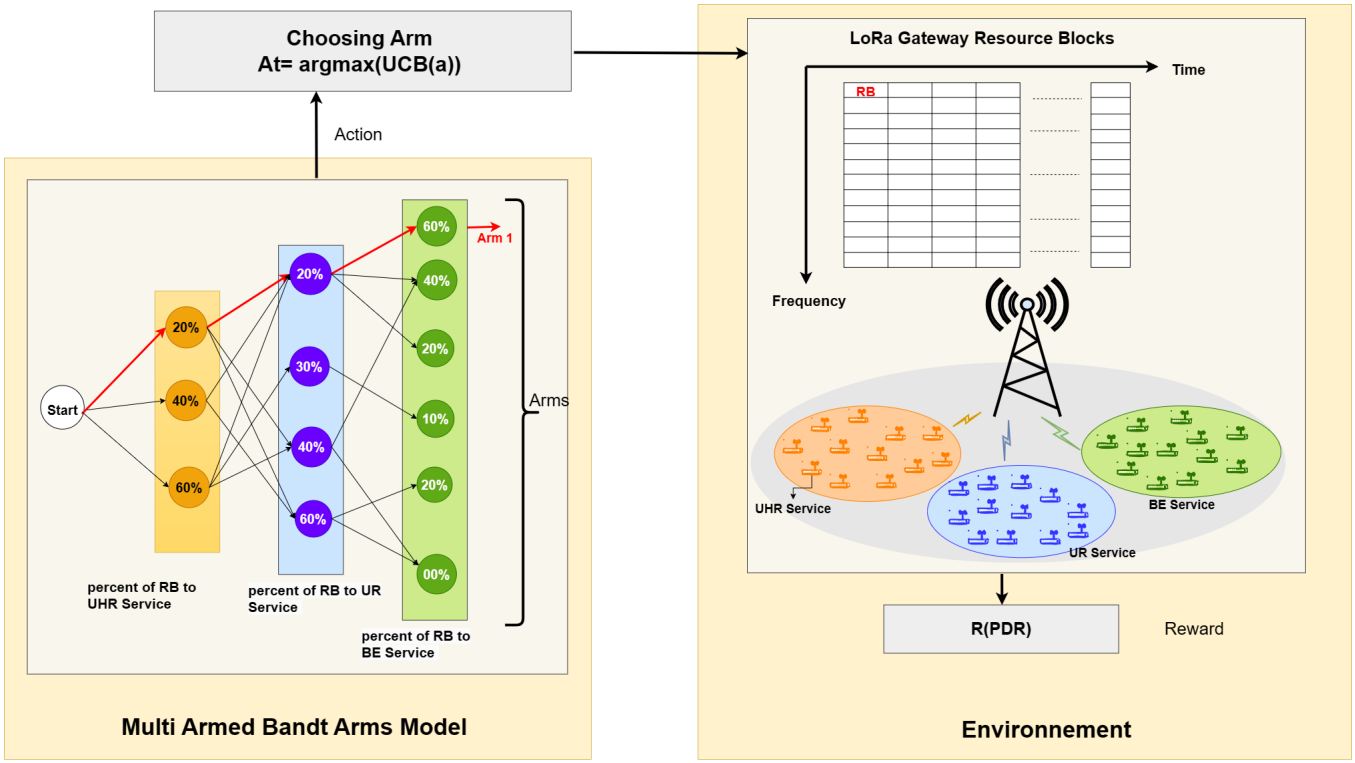


Fig. 1. Multi-Armed Bandit LoRaWAN Network architecture

allocated to service i in arm A_k. Then, we can represent each arm A_k as a vector of length n. Thus, the vector can be represented as:

$$A(k) = (p_{k,1}, p_{k,2}, \ldots, p_{k,n}) \tag{13}$$

where $A_k \in \mathbb{A}$ is the resource allocation vector for arm (scenario) k, $p_{k,i}$ corresponds to the percentage of resources allocated to service i in the k-th arm.

By using the equation (14), we ensure that the total percentage of resources allocated to all services is 100%

$$\sum_{i=1}^{n} p_{k,i} = 1 \tag{14}$$

C. UCB Solution

As the first solution, we propose a resource allocation based on the UCB-MAB (Upper Confidence Bound - Multi-Armed Bandit) algorithm. UCB is a well-known technique employed in the MAB framework. It effectively manages the trade-off between exploration (trying out different arms to gather information) and exploitation (allocating resources to the most promising arms based on available information). During each time step t, the UCB algorithm calculates the value of the UCB for each arm a. The UCB equation is given as:

$$UCB_t(a) = Q_t(a) + c\sqrt{\frac{2\log t}{n_t(a)}} \tag{15}$$

In this equation, $UCB_t(a)$ represents the upper confidence bound for action a at time step t. $Q_t(a)$ represents the estimated value of action a at time t, $n_t(a)$ denotes the number of times action a has been selected up to time t. The term $c\sqrt{\frac{2\log t}{n_t(a)}}$ represents the exploration term in the UCB algorithm. In which c is a parameter that balances exploration and exploitation. It controls the degree of exploration in the decision-making process. A higher value of c encourages more exploration, while a lower value promotes more exploitation of the currently best-performing options. $\sqrt{\frac{2\log t}{n_t(a)}}$ is the confidence interval term. It quantifies the uncertainty or confidence in the estimated reward for action a at time step t. It reflects the trade-off between exploring new actions and exploiting the actions with higher expected rewards.

To make a decision at each time step t, the agent selects the action with the highest UCB value, which is represented by the equation:

$$A_t = \underset{a}{\operatorname{argmax}}\ UCB_t(a) \tag{16}$$

In this equation, A_t represents the action selected at time step t. By choosing the action with the highest UCB value, the UCB algorithm balances between exploiting the actions that have shown high expected rewards and exploring the actions that have not been selected frequently or have uncertain reward estimates.

The learning agent aims to estimate the action values, denoted as $Q(a)$, which determine the potential rewards

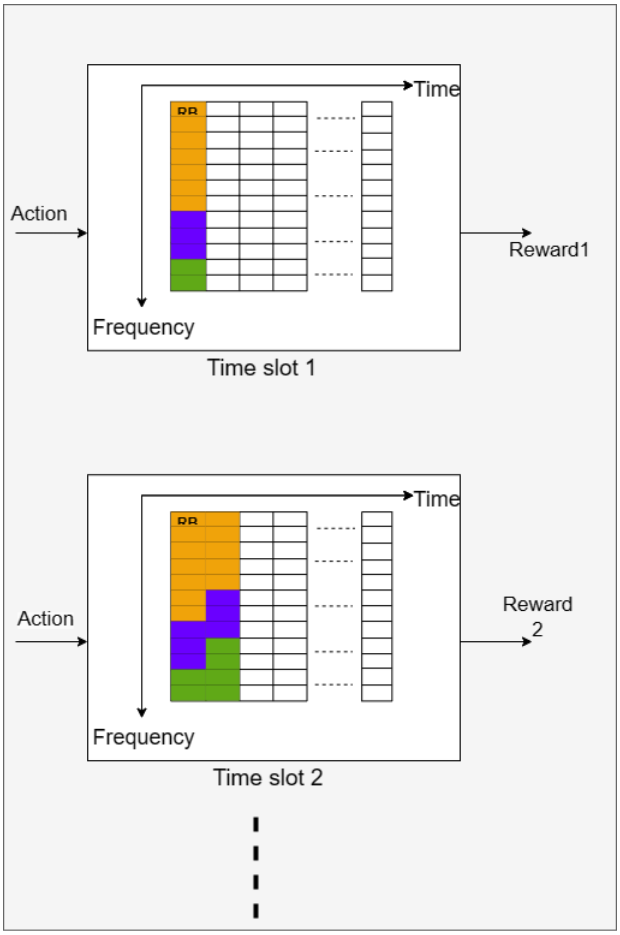


Fig. 2. Multi-Armed Bandit architecture

associated with each arm. Through continuously updating these estimates based on observed rewards received from the environment and selections, the UCB algorithm strives to identify the arm with the highest potential for long-term cumulative reward. This estimation process guides the agent's decision-making by striking a balance between exploiting the best-performing arm and exploring other arms that may offer higher rewards.

The Q-values are iteratively updated based on the observed rewards and the number of times each arm has been selected. This iterative update enables the algorithm to refine its estimates and make informed decisions that maximize the cumulative reward over time. The equation used for updating the action values is as follows:

$$Q_t(a) = Q_{t-1}(a) + \frac{R_t - Q_{t-1}(a)}{N_t(a)} \tag{17}$$

where R_t represents the reward obtained at time t, and $N_t(a)$ denotes the number of times arm a has been selected up to time step t. The update equation calculates the new Q-value $Q_t(a)$ by averaging the previous Q-value $Q_{t-1}(a)$ with the difference between the current reward and the previous Q-value, divided by the number of times the action has been selected. This iterative update process allows the Q-values to converge towards the true expected reward over time.

By iteratively updating the estimated action values and selecting actions based on their UCB values, the UCB algorithm gradually learns and improves its decision-making over time, effectively exploring the available options while maximizing the total reward obtained.

D. Q-UCB-MAB Solution

Algorithm 1 Q-UCB-MAB Resource Allocation

Input: Total time steps T, number of arms N, exploration parameter $c > 0$, learning rate α
Output: Optimal Allocations $A_1, A_2, \ldots, A_T$

1: Initialize Q-values: $Q(a) = 0$ for all arms a
2: Initialize action counts: $N_t(a) = 0$ for all arms a
3: **for** $t = 1$ to T **do**
4: **for** each arm a **do**
5: Calculate Upper Confidence Bound: $UCB_t(a) = Q(a) + c\sqrt{\frac{2 \log t}{N_t(a)}}$
6: **end for**
7: Choose the arm with the highest $UCB_t(a)$: $a_t = \arg\max_a UCB_t(a)$
8: Observe actual reward: $R_t(a_t)$
9: Update Q-value for chosen action: $Q(a_t) = (1 - \alpha) \cdot Q(a_t) + \alpha \cdot R_t(a_t)$
10: Increment action count for chosen action: $N_t(a_t) = N_t(a_t) + 1$
11: **end for**

As a second solution, we propose a modified version of the UCB-MAB algorithm that incorporates Q-learning to estimate the action values [16]. This approach enhances the algorithm's ability to make optimal decisions for resource allocation, even in complex and dynamic network environments like our LoRaWAN networks, where rewards change over time. Q-learning is an established reinforcement learning technique that allows the agent to learn and develop optimal action-selection policies by interacting with the environment [18]. It is commonly used in situations where the agent has limited knowledge about the environment and learns through trial and error.

By integrating Q-learning into the UCB-MAB algorithm, our modified Q-UCB-MAB algorithm incorporates the exploration-exploitation trade-off while also allowing the agent to learn and update action values based on observed rewards and expected future rewards. As proposed in Algorithm 1, during each time step t, the model calculates the UCB value for each action a (see line 5), and then, the agent selects the action with the highest UCB value using the equation given in line 7.

As presented in algorithm 1, the estimation of action values, denoted as $Q(a)$, is updated using the Q-learning equation. The equation for updating the action values is as follows:

$$Q_t(a) = (1 - \alpha) * Q_{t-1}(a) + \alpha^* R_t \tag{18}$$

where $Q_t(a)$ represents the updated value of action a at time step t, R_t denotes the immediate reward obtained at time t, α ($0 \leq \alpha \leq 1$) is the learning rate, and $Q_{t-1}(a)$ is the previous Q-value for action a. The Q-value update equation combines the previous Q-value with the newly obtained reward, allowing the agent to gradually learn and update its estimates of the Q-values over time.

Integrating the Q-learning equation into the UCB-MAB framework provides a more comprehensive and adaptive approach to resource allocation. This modified algorithm offers the potential for improved performance by incorporating both exploration and exploitation strategies, effectively navigating the trade-off between gathering information and exploiting known rewards.

E. ARIMA-UCB-MAB Solution

In our proposed resource allocation scheme, we propose a third solution that leverages the ARIMA (AutoRegressive Integrated Moving Average) model to forecast a predicted value of the next transmission cycle that assists the algorithm in selecting the optimal arm. This predictive value serves as a guiding factor, enabling the algorithm to make well-informed choices to improve resource allocation over time.

The ARIMA model is a powerful tool used in time series analysis to capture and forecast patterns in sequential data. It combines three essential terms: autoregressive (AR), differencing (I), and moving average (MA) [19]. The autoregressive (AR) component, denoted as AR(p), models the relationship between the current observation and its previous observations, and it is characterized by the parameter 'p' denoting the number of lagged series used to forecast periods ahead. The differencing (I) component, denoted as $I(d)$, is used to transform non-stationary data into stationary data by removing trends or seasonality, and it is represented by the parameter 'd', indicating the number of differencing orders applied. Lastly, the moving average (MA) component, denoted as $MA(q)$, captures the relationship between the current observation and the residual errors from previous observations. It is defined by the parameter 'q', which signifies the number of lagged forecast error terms used in the prediction equation [20] [19].

As mentioned in Algorithm 2, in our resource allocation scheme, we train the ARIMA model using historical data to predict the rewards for each arm in the next transmission cycle, denoted as $\hat{R}_t(a)$. By incorporating the ARIMA model into our resource allocation framework, we utilize the forecasted rewards as an essential component in calculating the UCB algorithm for each arm as follows:

$$\mathit{UCB}_t(a) = \hat{R}_t(a) + c\sqrt{\frac{2\log t}{n_t(a)}} \tag{19}$$

where $\hat{R}_t(a)$ represents the predicted ARIMA reward for arm a at time t, and the term $c\sqrt{\frac{2\log t}{n_t(a)}}$ represents the exploration term in the UCB algorithm. The UCB represents a measure of uncertainty in the PDR predictions and helps balance exploration and exploitation in your decision-making process. By selecting the arm with the highest UCB value (see line 7), the algorithm make an informed choice that optimizes the trade-off between exploring different arms and exploiting the arm that seems most promising according to both historical data and the level of uncertainty associated with its predictions.

Algorithm 2 ARIMA-UCB-MAB Resource Allocation

Input: Total time steps T, number of arms N, exploration parameter $c > 0$, α, ARIMA parameters
Output: Optimal Allocations $A_1, A_2, \ldots, A_T$

1: Train ARIMA model using historical data to forecast $\hat{R}_t(a)$ for each arm a
2: **for** $t = 1$ to T **do**
3: **for** each arm a **do**
4: Calculate forecasted reward: $\hat{R}_t(a)$ using ARIMA model
5: Calculate Upper Confidence Bound: $\mathit{UCB}_t(a) = \hat{R}_t(a) + c\sqrt{\frac{2\log t}{n_t(a)}}$
6: **end for**
7: Choose arm with highest $\mathit{UCB}_t(a)$: $a_t = \arg\max_a \ \mathit{UCB}_t(a)$
8: Observe actual reward: $R_t(a_t)$
9: Update ARIMA model with new data
10: **end for**

By leveraging the ARIMA model's capabilities in forecasting future rewards, our resource allocation scheme optimizes the allocation of resources by considering both the historical performance and the predicted future rewards. The integration of the ARIMA model with the UCB framework provides a robust approach for efficient resource allocation, allowing the system to adapt and make informed decisions while maximizing the cumulative rewards over time.

V. Simulation Results and Discussions

In this section, we demonstrate the effectiveness of our proposed resource allocation solutions based on the multi-armed bandit (MAB) algorithm by evaluating their performance using a model based on LoRaSim [21]. Our proposed LoRaWAN Slicing architecture includes three dynamic resource allocation mechanisms that optimize the utilization of resource blocks in the LoRa gateway. These mechanisms are based on the MAB algorithm, which enables to assignment of RBs to services in an adaptive manner, taking into account their current demand and performance requirements.

In this particular scenario, a LoRa gateway (GW) is installed in a 10 x 10 km² square area, and its resource blocks are shared among three services. Each RB is reserved exclusively for a particular service, and all LoRa nodes associated with that service use the assigned RBs to transmit their packets. The LoRa nodes are uniformly assigned to these services, with each LoRa node is assigned to one of these services. The

model operates over a period of 96 hours, divided into T = 3000 time steps in the time domain. To ensure that the LoRa nodes can transmit their packets with frequency diversity, 16 channels are allocated in the frequency domain.

To dynamically manage the allocation of RBs, in our model, we have implemented tree solutions based on the Multi-Armed Bandit (MAB) algorithm that balances the exploration and exploitation to continuously adapt the RB allocation strategy to maximize overall service performance while ensuring the SLA of each individual service. Table I provides a summary of the various parameters used in our LoRaWAN network.

TABLE I
SIMULATION PARAMETERS

Parameters name	**Value**
Simulation time	172800000 ms
Number of LoRa Gateway	1
Carrier Frequency (CF) / Channels number	868 MHz / 16
Transmission Power (TP)	14 dBm
Spreading Factor (SF)	{7,8,9,10,11,12}
Bandwidth (BW)	125 KHz
Coding Rate (CR)	$4/5$
Exploration-exploitation parameter C	0.5
Learning rate α in Q-learning	0.5
ARIMA parameters (p, d, q)	2,1,2

In our MAB-based LoRaWAN slicing model, we employ 8 arms (scenarios) to represent the different resource allocation options. Each arm is composed of three terms, with each term representing the proportion of allocated resource blocks assigned to individual services. Therefore, it's worth noting that the total of all resource allocations across services within each arm always adds up to 100%. Table II provides a summary of the different arms used in the Slicing Model, which helps to visualize the different resource allocation options available. By using this strategy, our solutions dynamically allocate resources to different services based on their needs, contributing to enhanced network performance.

TABLE II
RESOURCE ALLOCATION ARMS

Arms	Terms		
	Service 1	Service 2	Service3
1	20%	20%	60%
2	20%	40%	40%
3	20%	60%	20%
4	40%	20%	40%
5	60%	30%	10%
6	40%	60%	00%
7	60%	20%	20%
8	60%	40%	00%

A. The Arm Selection Analysis

To further analyze the performance of the Multi-Armed Bandit algorithm, Figure 3 illustrates the arm selection frequency for each of the proposed solutions. The results showed that all three solutions selected arm 5 more often than the other arms, indicating a clear preference for this arm. As presented in Table II, arm 5 is the best-performing arm, with

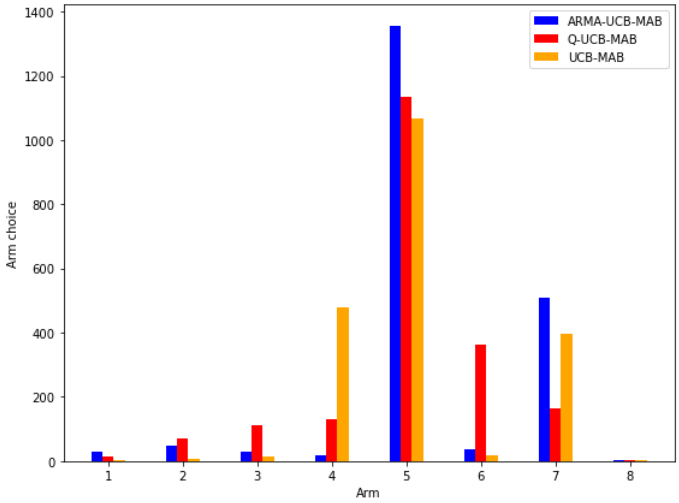


Fig. 3. The Arm Selection Analysis

60% of the resources allocated to the first service (UHRS) as the high priority service, 30% to the second service (HRS) as the medium priority service, and 10% to the third service (BE) as the best effort service. This allocation allowed the first service to receive the largest share of resources compared to the other services, as it is the most prioritized service. This scenario of resource allocation contributed to obtaining the best results compared to the other allocation resource scenarios (arms). However, we observed that the ARIMA-UCB-MAB solution selected arm 5 more often than the Q-UCB-MAB solution, and the Q-UCB-MAB selected arm 5 more often than the UCB-MAB solution. This discrepancy variation in arm selection frequency is due to differences in the exploration and exploitation strategies used by each solution. The ARIMA-UCB-MAB solution, which integrates the ARIMA predictive system and UCB decision-making framework, enables us to embrace arm 5 more widely. In contrast, the Q-UCB-MAB solution, which uses the integration of the Q-values-based Q-learning helps to control its exploration strategy, leading to a relatively greater selection of arm 5. On the other hand, the UCB-MAB algorithm follows classic UCB principles and assigns less frequent selections to arm 5, which means that it explores other options.

B. Packet Delivery Rate Analysis

Our results, as shown in Figure 4, indicate that the ARIMA-UCB-MAB solution outperformed both the Q-UCB-MAB and UCB-MAB solutions in terms of average cumulative reward. Specifically, the ARIMA-UCB-MAB solution consistently achieved higher rewards than the other two solutions throughout all iterations, despite not initially outperforming them in the first 300 iterations. It is possible that the ARIMA model required more time to learn and adjust to the data before making accurate predictions and achieving higher rewards. This could be due to the fact that the ARIMA solution is a

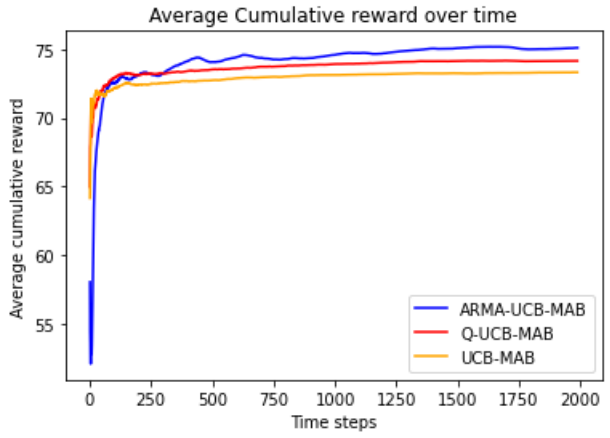


Fig. 4. The average cumulative reward over time

time-series model that needs sufficient data to make accurate predictions.

While the Q-UCB-MAB solution performed better than the UCB-MAB solution, as indicated in the graph, this is due to its ability that the Q-UCB-MAB strategy leverages Q-learning to update its Q-values based on past experiences and prioritize arms with higher expected rewards. This tailored exploration approach can lead to quicker identification and exploitation of the best-performing arm, resulting in higher overall rewards. In contrast, the UCB-MAB solution evenly explores all arms, which may not be the most efficient way to identify the best-performing arm. However, the UCB solution may not be the most efficient way to identify the best-performing arm, especially in scenarios where some arms have higher expected rewards than others.

Figure 5 displays the average Packet Delivery Ratio (PDR) of three services over time for the three solutions. It is evident that service 1 consistently maintains a PDR above its target threshold (80%). Similarly, service 2's PDR stays higher than its goal (50%), and service 3's PDR remains above its target (30%) for three solutions. This indicates that all three solutions have effectively met the service level agreement (SLA) for these services. Notably, the impact of prioritization based on the weight factor is reflected in the PDR performance. Service 1, which has the highest weight, consistently achieves the highest PDR among the services for all three solutions. Service 2, with a lower weight, demonstrates a slightly lower but still acceptable PDR for all three solutions. Similarly, Service 3, assigned the lowest weight, exhibits a lower but still satisfactory PDR for all three solutions. This highlights the importance of considering the weight factor when prioritizing services in LoRaWAN networks and demonstrates the ability of all three solutions to effectively meet the SLA requirements for each service. In summary, figure 5 showcases the algorithm's effectiveness in meeting the service level agreement (SLA) for each service, while also emphasizing the relative prioritization among these services.

Regarding the performance of the three different algorithms: UCB-MAB, Q-UCB-MAB, and ARIMA-UCB-MAB. While all three solutions demonstrate effectiveness in meeting the SLA for each service, the ARIMA-UCB-MAB solution, shown in Figure 5(c), stands out as the best performer. This is due to its unique approach of using time-series analysis to make predictions and adjust parameters accordingly, resulting in optimal performance from the start. The Q-UCB-MAB solution, shown in Figure 5(b), shows a gradual improvement in PDRs as learning occurs. This solution is based on the Q-learning algorithm, which requires a learning process to adjust the Q-values and optimize performance. The UCB-MAB solution, shown in Figure 5(a), achieves satisfactory PDR results for each service from the beginning. This solution is based on the Upper Confidence Bound algorithm, which efficiently balances between explore and exploit actions with high estimated rewards.

VI. Conclusion

This paper presents three approaches for resource allocation schemes for LoRaWAN network slicing based on a multi-armed bandit (MAB) algorithm. In this context, we explored the performance of these approaches and their efficacy in addressing the resource allocation challenge. The simulation results show that the ARIMA-UCB-MAB solution delivers the best results due to its adeptness in capturing intricate time series patterns, enabling accurate predictions based on historical data. Subsequently, the Q-UCB-MAB solution effectively demonstrates the advantages of continuous learning and adaptability, and finally the UCB solution due to its ability to effectively balance actions by taking into account the trade-off between exploration and exploitation. Overall, these three solutions demonstrate that MAB algorithms hold great promise for efficiently handling the complexities of allocating resources to network slices dynamically while maximizing the packet delivery rate (PDR) and guaranteeing the service level agreement (SLA) of each service.

References

[1] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," *ICT express*, vol. 5, no. 1, pp. 1–7, 2019.

[2] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A survey of lorawan for iot: From technology to application," *Sensors*, vol. 18, no. 11, p. 3995, 2018.

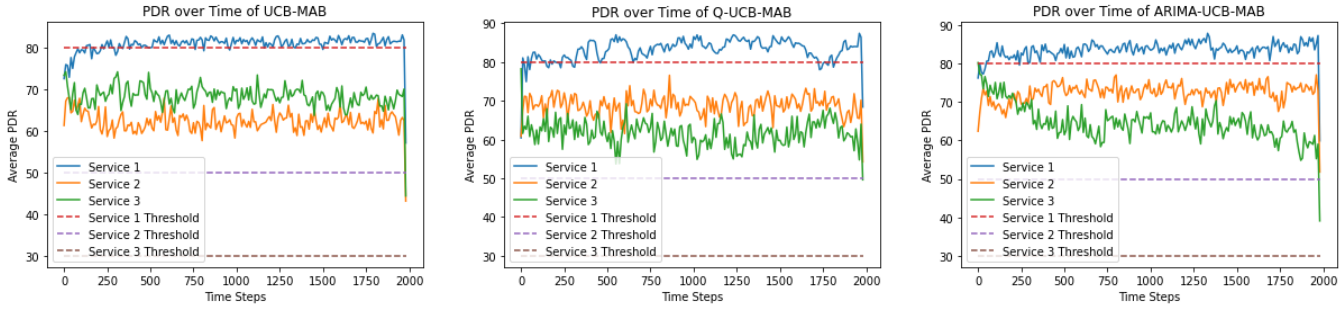
[3] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, "Network slicing for 5g: Challenges and opportunities," *IEEE Internet Computing*, vol. 21, no. 5, pp. 20–27, 2017.

[4] R. B. Sørensen, D. M. Kim, J. J. Nielsen, and P. Popovski, "Analysis of latency and mac-layer performance for class a lorawan," *IEEE Wireless Communications Letters*, vol. 6, no. 5, pp. 566–569, 2017.

[5] S. Bubeck, N. Cesa-Bianchi *et al.*, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends® in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[6] J. Zhu, Y. Song, D. Jiang, and H. Song, "Multi-armed bandit channel access scheme with cognitive radio technology in wireless sensor networks for the internet of things," *IEEE access*, vol. 4, pp. 4609–4617, 2016.

[7] F. Pase, M. Giordani, G. Cuozzo, S. Cavallero, J. Eichinger, R. Verdone, and M. Zorzi, "Distributed resource allocation for urllc in iiot scenarios: A multi-armed bandit approach," in *2022 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2022, pp. 383–388.



(a) The average PDR of UCB-MAB (b) The average PDR of Q-UCB-MAB (c) The average PDR of ARIMA-UCB-MAB

Fig. 5. The average Packet Delivery Rate Analysis

[8] S. Maghsudi and E. Hossain, "Distributed user association in energy harvesting dense small cell networks: A mean-field multi-armed bandit approach," *IEEE Access*, vol. 5, pp. 3513–3523, 2017.

[9] S. A. Almarzoqi, A. Yahya, Z. Matar, and I. Gomaa, "Re-learning exp3 multi-armed bandit algorithm for enhancing the massive iot-lorawan network performance," *Sensors*, vol. 22, no. 4, p. 1603, 2022.

[10] S. Hasegawa, R. Kitagawa, A. Li, S.-J. Kim, Y. Watanabe, Y. Shoji, and M. Hasegawa, "Multi-armed-bandit based channel selection algorithm for massive heterogeneous internet of things networks," *Applied Sciences*, vol. 12, no. 15, p. 7424, 2022.

[11] S. Dawaliby, A. Bradai, and Y. Pousset, "Adaptive dynamic network slicing in lora networks," *Future generation computer systems*, vol. 98, pp. 697–707, 2019.

[12] S. Messaoud, A. Bradai, and E. Moulay, "Online gmm clustering and mini-batch gradient descent based optimization for industrial iot 4.0," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1427–1435, 2019.

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

[14] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "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*, 2016, pp. 59–67.

[15] C. Semtech, "Lora modem design guide," *Semtech Wireless and Sensing, Tech. Rep.*, 2013.

[16] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[17] A. Slivkins *et al.*, "Introduction to multi-armed bandits," *Foundations and Trends® in Machine Learning*, vol. 12, no. 1-2, pp. 1–286, 2019.

[18] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[19] P. J. Brockwell and R. A. Davis, *Introduction to time series and forecasting*. Springer, 2002.

[20] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[21] "Lorasim." [Online]. Available: https://www.lancaster.ac.uk/scc/sites/lora/lorasim.html