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

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*

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

*Corresponding author
Email address: `fa.mardi@edu.umi.ac.ma` (Fatima Zahra Mardi)

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 ARIMA-UCB-MAB solution outperforms the two other solutions.

Keywords: Internet of Things, LoRaWAN network, network slicing, Multi-Armed Bandit, Resource allocation.

---

1. 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 network in order to meet various application requirements [3]. This innovation technology not only partitions the network, it also facilitates the allocation of specific resources and functionalities tailored to the particular needs of various IoT applications or services.

The integration of network slicing into LoRaWAN considerably enhances its 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, enabling them to meet the scalable requirements of diverse applications while efficiently managing the growing number of connected devices.

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 necessarily to design an efficient resource allocation scheme to dynamically allocate resources to each service according to changes in load. While Deep Reinforcement Learning (DRL) offers a promising avenue for achieving optimal resource allocation, it often require necessitate substantial training data and computational resources, posing potential constraints in resource-limited environments such as LoRaWAN networks.

In this context, the Multi-Armed Bandit (MAB) algorithm emerges as a promising approach due to their ability to make rapid decisions in dynamic environments, where con- ditions can change quickly [5]. Several methods utilizing Multi-Armed Bandit (MAB) algorithms have been used to improve the performance of various IoT networks. 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.

1.1. 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. Similarly, authors in [11] utilize multi-player MABs to address resource allocation in LoRa networks, focusing on optimizing transmission parameters to avoid interference, reduce packet loss, and minimize energy consumption. While these studies contribute to optimizing the overall network performance, they do not offer solutions for network slicing and may not be designed to effectively handle large-scale scenarios involving multiple network slices sharing the same available radio resources. This is particularly crucial in a network slicing environment, where context-based resource management for each service is necessary to ensure optimal performance.

One of the significant challenges in networking is the efficient support for diverse services with varying requirements and constraints. Network slicing has emerged as a key technology to address this challenge by enabling the creation of multiple virtual networks on top of a common physical infrastructure. In [12], the authors introduce a multi-agent DRL strategy within the realm of spectrum trading for Network Slicing in 5G RAN. The paper focuses on optimizing resource allocation using a Multi-Agent Deep Reinforcement Learning (MADRL) approach. In [13], the authors propose an approach based using reinforcement learning to optimize resource allocation in B5G networks by leveraging LWA technology, while incorporating Network Slicing to address diverse user service demands in the 6G era.

Moreover, recent works have explored network slicing implementation in LoRaWAN, showcasing a growing interest in optimizing resource allocation and meeting the diverse service requirements of IoT applications and services. In [14], 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 [15], 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. Moreover, in [16], the authors 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 for each network slice. However, these approaches often rely on static channel allocation, which can lead to under utilization or overloading of resources. As the number of services on the network increases, this static approach can compromise the overall efficiency of the network and limit its ability to provide optimal performance for diverse IoT applications.

Meanwhile, In [17], the authors propose a resource allocation strategy in LoRaWAN network slicing using the H-DQN approach to ensure service prioritization and enhance the network's Packet Delivery Rate. Authors in [18] propose an approach to differentiating Packet Delivery Ratios (PDRs) in LoRaWAN. They exploit the isolation provided by the available radio frequencies to group devices into clusters with different PDR targets and to optimize behavior configuration in order to reach the desired objectives. In [19], the authors propose a resource allocation model using deep federated reinforcement learning in LoRaWAN network slices. which consist to dynamically optimize the SF and TP parameters, in order to improve the quality of service of users in various LoRa network slices. However, these proposed approaches may not thoroughly address the need for dynamic adaptation to changing traffic demands and environmental conditions within LoRaWAN networks.

1.2. Contributions and outlines

Based on the literature review and analysis in LoRaWAN slicing, the above works on network slicing have the following limitations.

- In most of the existing works on LoRaWAN slicing, radio resource allocation is based on a channel-based system. This approach, marked by a static allocation paradigm, has its limitations. Essentially, in a system when we assign one or more channels to specific service, and this becomes a problem due to the finite number of available channels. As the number of services grows, it becomes increasingly challenging to meet the needs of all these services within this restricted channel framework.

- Most LoRaWAN network slicing schemes face challenges with the dynamic nature of real-world traffic patterns and the consequent challenges of managing evolving device traffic loads.
- Some works based on reinforcement learning (DRL) approaches require large amounts of data and computational power to learn and adapt. This high resource demand can make them unsuitable for resource-constrained servers, such as those used in IoT or embedded environments.

Addressing the limitations experienced by existing methodologies, in this study, we focus on a single LoRa GW, uplink scenario that contains several network slices sharing available radio resources, our paper emphasizes the network slicing approach within the new adaptive LoRaWAN framework. Specifically, 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.The main contributions of this paper can be summarized as follows:

- We propose an approach that focuses on optimizing resource allocation by taking advantage of two key dimensions: time and frequency. By dividing channels into smaller time intervals, we facilitate more efficient and adaptive resource allocation. This approach is extremely useful in addressing the challenges posed by frequent traffic fluctuations, which are known to have a negative impact on network performance. By integrating temporal channel partitioning, our strategy improves resource utilization, guaranteeing a dynamic network adaptable to changing traffic demands. By focusing on the crucial dimensions of time and frequency, our approach promotes highly efficient resource allocation tailored to the diverse needs of individual services within the network, achieved through optimizing the Packet Delivery Rate (PDR) and ensuring the Service Level Agreement (SLA) of each service.

- We propose three innovative solutions utilizing MAB algorithms to address resource allocation challenges across three different services in order to maximize network performance while ensuring the Service Level Agreements (SLAs) of supported services. MAB algorithms are particularly well-suited for this task due to their ability to make rapid decisions in dynamic environments, where conditions can change quickly. Firstly, we use the Upper Confidence Bound - Multi-Armed Bandit (UCB-MAB) algorithm, which 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. Secondly, 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. Finaly, 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) and ensuring the SLA. 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, enhancing its capability to meet varying SLA effectively.

The rest of this paper is organized as follows. Section 2 discusses the system model. Section 3 elaborates on the proposed resource allocation solutions. Section 4 shows and discusses the obtained results of our proposed solutions. Finally, the conclusion is given in Section 5.

2. System Model

2.1. 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.,

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

2.2. 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 $X_\sigma \sim N\,(0, \sigma^2)$, the normal distribution with zero mean and σ^2 variance to account for shadowing [20].

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 [21] [20]:

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

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.

2.3. 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.

2.4. 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 Agreement (SLA) 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 for a service, where $\mathcal{X}_{b_{ij},s,t} = 1$ indicates that resource block b_{ij} is allocated to service s 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.

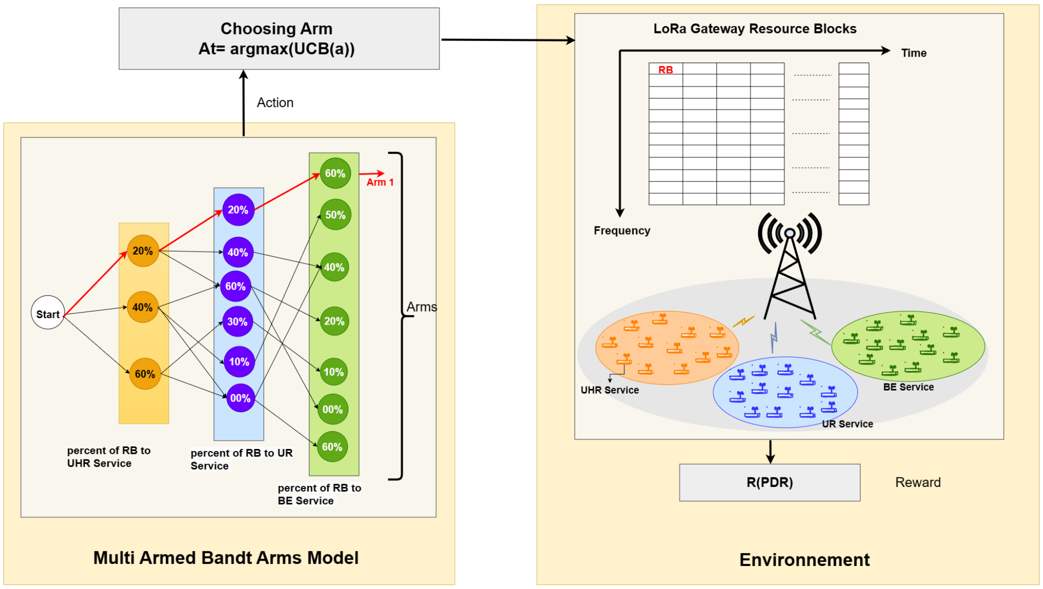


Figure 1: Multi-Armed Bandit LoRaWAN Network architecture

3. Multi-Armed Bandits based Resources allocation

3.1. 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 [22]. 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 [22].

3.2. MAB Resource Allocation

The Multi-Armed Bandit (MAB) framework [23] 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

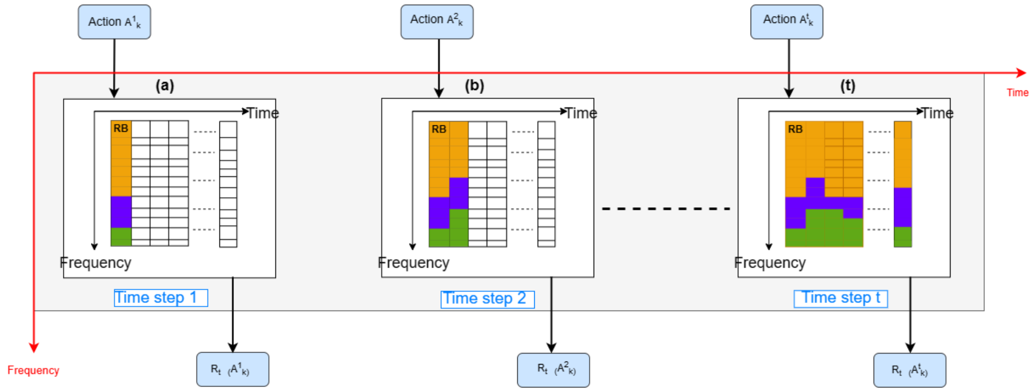


Figure 2: Multi-Armed Bandit architecture

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 $\mathbb{A} = \{A_1, A_2, ...A_k\}$. As shown in Figure 2, 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, where each arm is related to a resource allocation scenario. We achieve this by defining each scenario (arm) as a combination of resource allocation percentages between different services as presented in Figure 1. 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 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%

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

3.3. 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 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.

3.4. Q-UCB-MAB Solution

As a second solution, we propose a modified version of the UCB-MAB algorithm that incorporates Q-learning to estimate the action values [22]. 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

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 $A_1, A_2, \ldots, A_T$

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

environment [24]. 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:

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

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.

Furthermore, Q-UCB-MAB incorporates the UCB strategy for exploration. This is achieved by adding the exploration term denoted as $c\sqrt{\frac{2\log t}{N_t(a)}}$ to the Q-value. This term considers how often an option has been explored in the past and encourages exploration of less-tested options even if their current Q-values are lower.

By combining Q-learning's adaptation and UCB's exploration, Q-UCB-MAB effectively learns and adapts in dynamic network resource allocation scenarios. This integration enhances the algorithm's ability to balance exploration and exploitation strategies, enabling it to navigate the trade-off between acquiring new information and leveraging known rewards effectively.

3.5. 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

Algorithm 2 ARIMA-UCB-MAB Resource Allocation

Input: Total time steps T, number of arms N, exploration parameter $c > 0$, ARIMA parameters

Output: Optimal Allocations $A_1, A_2, \ldots, A_T$

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

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) [25]. 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 [26][25].

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:

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

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. Additionally, UCB incorporates this exploration term to encourages exploring new options even if they do not have the highest historical returns. By combining the reward prediction from ARIMA and the exploration term of UCB, the algorithm selects the option that offers the best balance between exploiting historically high-performing options and exploring potentially better new options. 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.

The ARIMA-UCB-MAB solution plays a vital role in enhancing network performance through its innovative approach to resource allocation. By leveraging the ARIMA model's capability to forecast future reward values, the solution guides the UCB algorithm towards options with the highest potential performance. This strategic allocation, informed by both exploration-exploitation term and predicted future demands, optimizes resource utilization. Consequently, the system requires less exploration of sub-optimal options, leading to faster learning and quicker achievement of optimal performance.

4. 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 [27], a discrete-event simulator developed by Bor et al. using the Simpy library to analyze scalability and collision issues in LoRa networks. The simulations are built in Python 3.9. 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 designed to dynamically allocate resource blocks to different services (RBs) based on the real-time demand and specific service requirements within the LoRa network. By continuously adjusting the allocation of resources based on current demand and service needs, this intelligent approach ensures efficient resource allocation, maximizing packet delivery ratio (PDR) and meeting the service level agreements (SLAs) of diverse services within the LoRa network.

Our scenario presents an original and distinct framework for optimizing resource block (RB) allocation in LoRa gateway. By incorporating a temporal aspect, the allocation of RBs for individual services, combined with the adaptive RB allocation strategy based on real-time demand and performance requirements, sets our model apart from traditional resource allocation methodologies in LoRaWAN networks slicing. Consequently, direct comparisons with existing works may be limited due to the novelty and tailored nature of our proposed scenario.

In this particular scenario, we consider an uplink scenario in which a LoRa gateway (GW) is installed in a 10 x 10 km^2 square area, serving as a central communication for the LoRa network. The LoRa nodes are evenly distributed across the coverage area of the gateway. The resource blocks of LoRa gateway are shared among three services. Each RB is reserved exclusively for a particular service, with the objective of optimizing resource utilization and enhance service performance by providing dedicated resources for each service.

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 (345600000 ms), divided into T = 2000 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 within the typical 868 MHz European sub-band. Additionally, each LoRa device transmits packets to the LoRa gateways using a random spreading factor between 7 and 12, with a payload size of 20 bytes per packet. To accurately model the traffic generation process, we adopt a Poisson process with a packet transmission rate (λ) of 1×10^{-5} milliseconds (ms). We perform simulations using 100 LoRa nodes, with the LoRa nodes adhering to a 1% duty cycle restriction as mandated for operation in the ISM bands. To ensure reliability and consistency in the results, the simulation is repeated 20 times.

To dynamically manage the allocation of RBs, in our model, we have implemented tree solutions UCB-MAB, Q-UCB-MAB, and ARIMA-UCB-MAB 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. To optimize the performance of these solutions within the LoRaWAN, we configured them with specific parameters influencing their exploration-exploitation trade-off and resource allocation decisions, in which, we set the confidence parameter (c) to 5 to balance exploration and exploitation. In the Q-UCB-MAB approach, we incorporated a learning rate (α) with $\alpha = 0.5$ to strike a balance between learning from new experiences and prioritizing long-term performance. Regarding the ARIMA-UCB-MAB solution, we integrated an ARIMA model with specific parameters (p, d, q) to predict future network traffic patterns. By considering these aspects, the ARIMA model builds a better overview of how the network traffic typically operates and can anticipate future patterns. This information is then used by the UCB algorithm to make informed decisions about allocating resources effectively. For a detailed summary of the parameters used in our LoRaWAN network. Table 1 provides a summary of the various parameters used in our LoRaWAN network.

Table 1: Simulation Parameters

Parameters name	**Value**
Simulation time	345600000 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	5.0
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 2 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 2: Resource Allocation Arms

Arms	Terms		
	Service 1	Service 2	Service3
1	20%	20%	60%
2	20%	40%	40%
3	20%	60%	20%
4	40%	60%	00%
5	60%	30%	10%
6	40%	10%	50%
7	60%	00%	40%
8	40%	00%	60%

4.1. 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 2, arm 5 is the best-performing arm, with 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

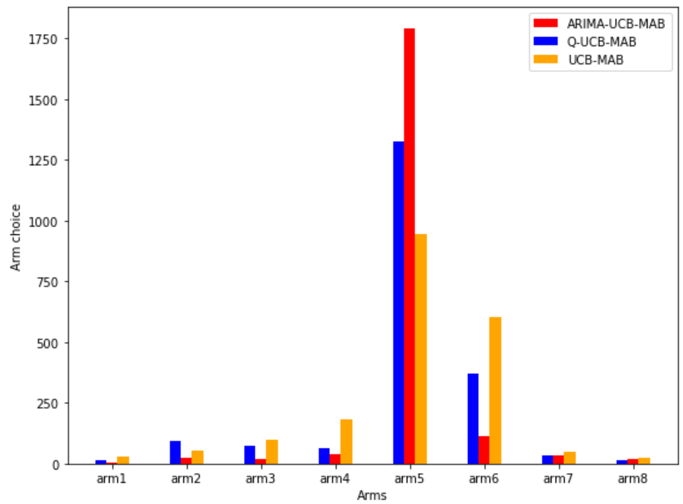


Figure 3: The Arm Selection Analysis

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.

4.2. 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. This can be attributed to its unique approach of using time-series analysis to make predictions. Leveraging an ARIMA model to forecast the rewards associated with each arm, the ARIMA-UCB-MAB solution optimized decision-making by intelligently balancing the trade-off between exploring lesser-known arms and exploiting known arms to achieve higher rewards. 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

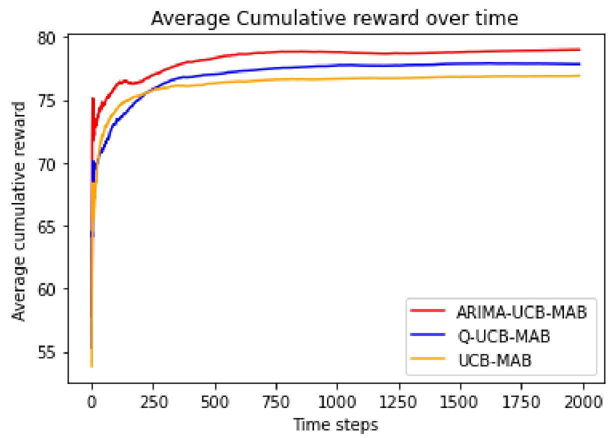
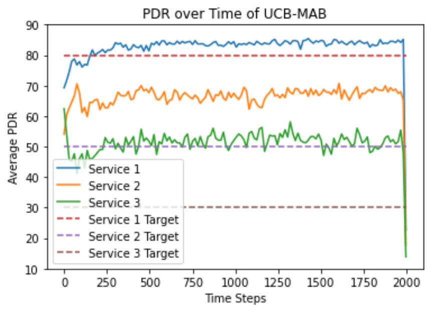


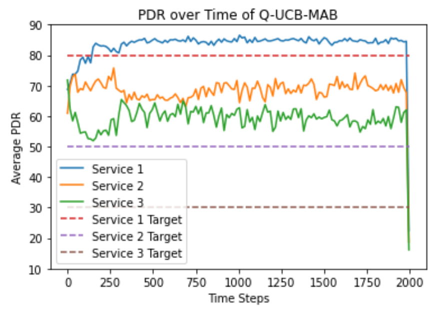
Figure 4: The average cumulative reward over time

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.

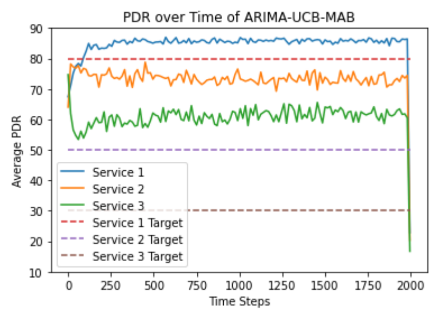
Figure 5 displays the average Packet Delivery Ratio (PDR) of three services over time for the three solutions. It is evident that Service 1 consistently maintains a PDR above its target of 80%, as illustrated in Figure 5 (Service 1 Target). Similarly, Service 2's PDR consistently surpasses its target of 50%, as depicted in Figure 5 (Service 2 Target), while Service 3's PDR consistently remains above its target of 30%, as shown in Figure 5 (Service 3 Target) for all 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 agree-



(a) The average PDR of UCB-MAB



(b) The average PDR of Q-UCB-MAB



(c) The average PDR of ARIMA-UCB-MAB

Figure 5: The average Packet Delivery Rate Analysis

ment (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. ARIMA-UCB-MAB demonstrates its efficiency by rapidly converging to optimal PDR levels compared to Q-UCB-MAB and UCB-MAB. It achieves the highest PDR1 of 88% for Service 1, which is 3% higher than Q-UCB-MAB and 5% higher than UCB-MAB. Similarly, for service 2 and 3, ARIMA-UCB-MAB obtains a PDR2 of 75%, exceeding Q-UCB-MAB by 5% and UCB-MAB by 7% and PDR3 of 63%, exceeding Q-UCB-MAB by 3% and UCB-MAB by 9%. These results indicate that ARIMA-UCB-MAB is more efficient for achieving the best superior performance. In addition, ARIMA-UCB offers better stability as demonstrated by its smoother PDR curve that is smoother and less sensitive to fluctuations than that of the Q-UCB-MAB. This improved stability leads to more reliable and consistent performance, reducing the risk of service interruptions. 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. These performance improvements directly benefit real-world Internet of Things (IoT) applications, providing a robust solution for scenarios requiring reliable service discovery, stable performance, and efficient resource allocation. For example, in healthcare, ARIMA-UCB-MAB can prioritize the transmission of critical health data during remote patient monitoring, enabling timely medical interventions. Similarly, in smart agriculture, ARIMA-UCB-MAB can prioritize real-time sensor data transmission for optimal irrigation, enhancing resource management efficiency. On the other hand, the Q-UCB-MAB solution, shown in Figure 5(b), 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. This solution is based on the Upper Confidence Bound algorithm, which efficiently balances between explore and exploit actions with high estimated rewards. Overall, these three solutions demonstrate an efficient resource allocation to network slices by optimizing the packet delivery rate (PDR) and guaranteeing the service level agreement (SLA) of each service.

4.3. Cost Function Analysis

Figure 6 analyzes the performance of MAB algorithms using a specific cost function defined in Equation 10, which evaluates the difference between the target PDR and the actual PDR for each service s. As shown in Figure 6, it is evident that all three solutions exhibit a decreasing in cost, converging to 0, indicating improved performance. This reduction signifies progress towards meeting the Service Level Agreement (SLA) requirements for each service.

The ARIMA-UCB-MAB solution, represented by the red curve, exhibits superior performance in cost reduction. It achieves a rapid decrease in cost during the initial phase, thanks to its ability to leverage historical data and make predictions about future PDR. This allows for optimal resource allocation, leading to the algorithm quickly reaching a stable state with a low cost, to 0. Furthermore, the blue curve depicts the performance of the Q-UCB-MAB solution. A decrease in cost function is also observed, but less performance-enhancing than for ARIMA-UCB-MAB during the initialization

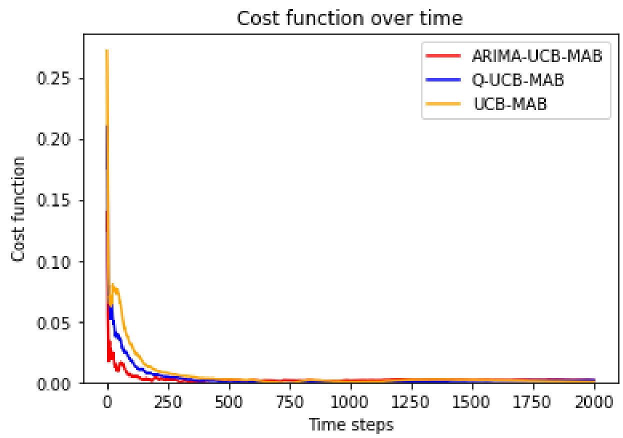


Figure 6: The cost function over time

phase. This suggests that Q-UCB-MAB achieves a lower rate of improvement compared to ARIMA-UCB-MAB in the beginning. Despite this, the paragraph clarifies that the blue curve still exhibits a decreasing trend, indicating the algorithm is learning and improving its cost function over time. On the other hand, the orange curve corresponds to the UCB-MAB solution, demonstrating a slower decrease in cost function compared to the other two solutions in the initialization phase. This is because UCB-MAB initially explores all options to gather information. However, this exploration phase makes it less susceptible to getting stuck in suboptimal solutions. The significant decrease in cost observed across all solutions (ARIMA-UCB-MAB, Q-UCB-MAB, and UCB-MAB) signifies their potential for achieving the SLA of each service, defined by the PDR target. A cost of 0 indicates successful SLA satisfaction.

5. 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. In contrast, the proposed ARIMA-UCB-MAB method provides a lightweight resource allocation solution for LoRaWAN networks, requiring less training data and offering faster convergence compared to Deep Reinforcement Learning (DRL). While it may not adapt as well as DRL, ARIMA-UCB-MAB stands out as a compelling choice for situations where interpretability, efficiency, and predictable traffic patterns are pivotal factors to consider. Additionally, its minimal computational requirements and easy implementation using existing software enhance its practicality for real-world applications. 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. While these MAB-based methods offer a promising result, future work can focus on making them more adaptable. This could involve dynamically adapting MAB hyperparameters and incorporating additional metrics (fairness and energy). Exploring complementary prediction models can also enhance resource allocation accuracy. By refining these approaches, we can unlock the full potential of MABs for network slicing resource allocation.

References

[1] K. Mekki, E. Bajic, F. Chaxel, F. Meyer, A comparative study of lpwan technologies for large-scale iot deployment, ICT express 5 (1) (2019) 1–7.

[2] J. Haxhibeqiri, E. De Poorter, I. Moerman, J. Hoebeke, A survey of lorawan for iot: From technology to application, Sensors 18 (11) (2018) 3995.

[3] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, R. Jain, Network slicing for 5g: Challenges and opportunities, IEEE Internet Computing 21 (5) (2017) 20–27.

[4] R. B. Sørensen, D. M. Kim, J. J. Nielsen, P. Popovski, Analysis of latency and mac-layer performance for class a lorawan, IEEE Wireless Communications Letters 6 (5) (2017) 566–569.

[5] S. Bubeck, N. Cesa-Bianchi, et al., Regret analysis of stochastic and nonstochastic multi-armed bandit problems, Foundations and Trends® in Machine Learning 5 (1) (2012) 1–122.

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

[7] F. Pase, M. Giordani, G. Cuozzo, S. Cavallero, J. Eichinger, R. Verdone, 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.

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

[9] S. A. Almarzoqi, A. Yahya, Z. Matar, I. Gomaa, Re-learning exp3 multi-armed bandit algorithm for enhancing the massive iot-lorawan network performance, Sensors 22 (4) (2022) 1603.

[10] S. Hasegawa, R. Kitagawa, A. Li, S.-J. Kim, Y. Watanabe, Y. Shoji, M. Hasegawa, Multi-armed-bandit based channel selection algorithm for massive heterogeneous internet of things networks, Applied Sciences 12 (15) (2022) 7424.

[11] H. Dakdouk, R. Féraud, N. Varsier, P. Maillé, R. Laroche, Massive multi-player multi-armed bandits for iot networks: An application on lora networks, Ad Hoc Networks 151 (2023) 103283.

[12] G. O. Boateng, G. Sun, D. A. Mensah, D. M. Doe, R. Ou, G. Liu, Consortium blockchain-based spectrum trading for network slicing in 5g ran: A multi-agent deep reinforcement learning approach, IEEE Transactions on Mobile Computing (2022).

[13] N. A. Elmosilhy, M. M. Elmesalawy, I. I. Ibrahim, A. M. Abd El-Haleem, Joint q-learning based resource allocation and multi-numerology b5g network slicing exploiting lwa technology, IEEE Access (2024).

[14] S. Dawaliby, A. Bradai, Y. Pousset, Adaptive dynamic network slicing in lora networks, Future generation computer systems 98 (2019) 697–707.

[15] S. Messaoud, A. Bradai, E. Moulay, Online gmm clustering and mini-batch gradient descent based optimization for industrial iot 4.0, IEEE Transactions on Industrial Informatics 16 (2) (2019) 1427–1435.

[16] F. Z. Mardi, M. Bagaa, Y. Hadjadj-Aoul, 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.

[17] F. Z. Mardi, M. Bagaa, Y. Hadjadj-Aoul, N. Benamar, Heuristic-deep q-network-based network slicing in lorawan, in: ICC 2023-IEEE International Conference on Communications, IEEE, 2023, pp. 4731–4736.

[18] A. Aimi, F. Guillemin, S. Rovedakis, S. Secci, Packet delivery ratio guarantees for differentiated lorawanservices, in: GLOBECOM 2022-2022 IEEE Global Communications Conference, IEEE, 2022, pp. 2014–2019.

[19] E. Ossongo, M. Esseghir, L. Merghem-Boulahia, A multi-agent federated reinforcement learning-based optimization of quality of service in various lora network slices, Computer Communications 213 (2024) 320–330.

[20] M. C. Bor, U. Roedig, T. Voigt, 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.

[21] C. Semtech, Lora modem design guide, Semtech Wireless and Sensing, Tech. Rep. (2013).

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

[23] A. Slivkins, et al., Introduction to multi-armed bandits, Foundations and Trends® in Machine Learning 12 (1-2) (2019) 1–286.

[24] C. J. Watkins, P. Dayan, Q-learning, Machine learning 8 (1992) 279–292.

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

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

[27] Lorasim.
URL https://www.lancaster.ac.uk/scc/sites/lora/lorasim.html