Journal of
Sensor and Actuator Networks

MDPI

Article

Multi-Armed Bandit Algorithm Policy for LoRa Network Performance Enhancement

Anjali R. Askhedkar and Bharat S. Chaudhari *

School of Electronics and Communication Engineering, Dr. Vishwanath Karad MIT World Peace University, Pune 411038, India; anjali.askhedkar@mitwpu.edu.in
* Correspondence: bsc@ieee.org

Abstract: Low-power wide-area networks (LPWANs) constitute a variety of modern-day Internet of Things (IoT) applications. Long range (LoRa) is a promising LPWAN technology with its long-range and low-power benefits. Performance enhancement of LoRa networks is one of the crucial challenges to meet application requirements, and it primarily depends on the optimal selection of transmission parameters. Reinforcement learning-based multi-armed bandit (MAB) is a prominent approach for optimizing the LoRa parameters and network performance. In this work, we propose a new discounted upper confidence bound (DUCB) MAB to maximize energy efficiency and improve the overall performance of the LoRa network. We designed novel discount and exploration bonus functions to maximize the policy rewards to increase the number of successful transmissions. The results show that the proposed discount and exploration functions give better mean rewards irrespective of the number of trials, which has significant importance for LoRa networks. The designed policy outperforms other policies reported in the literature and has a lesser time complexity, a comparable mean rewards, and improves the mean rewards by a minimum of 8%.

Keywords: LPWAN; LoRa; spreading factor; multi-armed bandits; discounted UCB; energy efficiency

check for updates

Citation: Askhedkar, A.R.; Chaudhari, B.S. Multi-Armed Bandit Algorithm Policy for LoRa Network Performance Enhancement. *J. Sens. Actuator Netw.* **2023**, *12*, 38. https://doi.org/10.3390/jsan12030038

Academic Editor: Mingjun Xiao

Received: 30 March 2023
Revised: 24 April 2023
Accepted: 27 April 2023
Published: 4 May 2023

1. Introduction

The low-power wide-area network (LPWAN) is a promising technology for the growing Internet of Things (IoT) and machine-to-machine (M2M) applications offering wide-area communication among a large number of devices with benefits such as low power and low cost. It can cater to a wide range of applications, such as agriculture, healthcare, home automation, smart city, smart grid, monitoring of industrial assets, critical infrastructure, environment, wildlife, and many others. Long range (LoRa) is an LPWAN technology that requires low power and offers long-range communication. It uses the unlicensed industrial, scientific, and medical (ISM) band with frequency ranges such as 433, 868, or 915 MHz, and can support data rates up to 50 Kbps. A LoRa network is built on the star of stars topology that comprises multiple nodes that communicate with a gateway using the LoRaWAN MAC layer protocol, and the Chirp spread spectrum (CSS) modulation method. The gateways relay the messages from the end devices to the network server [1]. LoRa modulation offers several parameters for customization, such as channels, spreading factors (*SFs*), transmission power, bandwidth, and data rate. The choice of these parameters affects the transmission energy, range, time-on-air, coverage, capacity, and overall performance of the network [2]. The end devices communicate with the gateway using an available subchannel and one of the six spreading factors. Collision may occur if different devices use the same channel and the same spreading factor simultaneously. Since there can be a large number of end devices in the network, the probability of collision increases, resulting in the degradation of network performance. Additionally, as the network grows, vulnerability and security issues arise wherein CSS used by LoRa seems to offer a robust approach [3]. In this context, machine learning algorithms can be utilized for optimal parameter selection to

minimize interference and maximize energy efficiency, and, in turn, maximize the network performance [4]. Efficient resource utilization and adaptive transmission are approaches to achieve better energy efficiency in LPWANs [5].

Resource allocation or parameter selection approaches in LoRa networks can be categorized as a centralized and distributed approach. A device can individually select its data rate and transmission power or let the network control these parameters. The adaptive data rate mechanism (ADR), recommended by the LoRa Alliance and implemented by the LoRa network, is an example of a centralized approach. The network server controls the transmission parameters of the end node. It reduces the transmit power of a node by adapting the data rate. In such a case, the network needs the knowledge of the transmitted power of the node for about the previous twenty transmissions, and then it estimates the transmit power for the next transmission by changing the data rate and communicates it to the node. The node then uses the information received from the server and adapts its parameters. However, the disadvantage of this approach is that it is suitable only in stable RF situations where the end nodes do not move [1]. In practical situations, the nodes can be mobile. Even for the simplest configuration with assumptions such as the Poisson point process distribution, i.e., nodes are uniformly distributed around the gateway, constant transmit power, single channel, and no interference from non-LoRa nodes, selecting the optimal parameters is still a complex problem. The ADR algorithm also has some limitations. ADR allocates *SF* to a node, depending on the uplink signal-to-noise ratio (SNR), so that nodes closer to the gateway select lower *SF* and nodes farther from the gateway use higher *SF*. If all the nodes are closer, then ADR may allot the same *SF* to them, leading to collisions due to the overuse of the same *SF* and underuse of other *SFs* [6]. Additionally, ADR tends to reduce energy consumption but suffers large packet losses [2].

The distributed learning algorithms seem useful in such scenarios where the end devices are considered as intelligent agents that choose a particular parameter from a given set of values at a given time. In this work, we propose to minimize interference and maximize the energy efficiency of the end devices in the LoRa network. The spreading factor is a factor whose selection impacts these two performance parameters of the LoRa network. Several other parameters, such as transmission power, coding rate, bandwidth, and channel frequency, also affect the network performance. The framework of a special class of learning algorithms, the reinforcement learning-based technique, and multi-armed bandit (MAB) algorithm adhere to such a scenario. This paper explores the discounted upper confidence bound (DUCB), a class of MAB algorithms, to address this problem and improvise successful transmissions with less time complexity.

The work focuses on decentralized decision making and optimizing the transmitter parameter selection. This paper makes two key contributions. The first is the use of discounted UCB for the optimal selection of LoRa transmission parameters. Previous studies indicate using MAB algorithms, such as TS and UCB, for LoRa applications. DUCB has been used in previous studies for cognitive radio, not specifically for LoRa as per the studies in the literature. The second contribution is the development of a discount function and exploration bonus for DUCB, and thus, a novel DUCB algorithm for LoRa requirements. The new algorithm aims to make an optimal selection of one of the LoRa parameters so as to maximize the successful transmissions leading to better energy efficiency, thus improving the network performance. The developed DUCB policy is compared with the existing UCB, DUCB, and other algorithms, and gives promising results with the advantage of lesser time complexity. The algorithm can also be simultaneously utilized for the selection of multiple LoRa parameters.

The paper is organized as follows. The related works are discussed in Section 2. Section 3 describes the LoRa technology in brief. The MAB and DUCB algorithms are described in Section 4. A novel policy with a new discount function and exploration bonus is proposed in Section 5. The impact of several discount functions and exploration bonuses on the performance of the LoRa network is evaluated using simulations, and an optimal algorithm is proposed in Section 6. Section 7 includes conclusions based on the results.

2. Related Work

LoRa technology is increasingly being used in IoT and wireless sensor network applications nowadays. A major factor that influences this network design is the network performance that is determined by the selection of transmission parameters. In [7], the design and implementation of a LoRa-based wireless sensor network for water quality monitoring is illustrated, and performance in terms of coverage is tested in a real environment. A modular LoRa-based IoT platform for smart farm monitoring, enabling the collection, exchange, processing, and visualization of relevant farm data, is proposed and evaluated in [8]. In [9], a framework for large-scale LoRa network deployment is designed using an open source LoRa emulator to estimate the network performance prior to real deployment.

LoRa transmissions and network performance have been widely studied in the literature. There are different approaches to selecting transmission parameters that improve performance, increase energy efficiency for LoRa networks, wireless sensor networks, IoT, cognitive radio, and others. A LoRa device can be configured to use a variety of transmission powers, coding rates, spreading factors, and bandwidth sets, and leading to about 6720 possible combinations. It is, thus, a huge task to determine the best possible option for maximizing the network performance. A thorough investigation of the effect of LoRa parameters on reliability and energy consumption is carried out, and a link probing method that efficiently determines an appropriate transmission parameter value is developed as a step towards an automated mechanism [10]. A new method to adjust the data rate and channel in dense LoRa networks is also proposed to improve the utilization of resources. Based on the data extraction rate, the method carries out channel estimation and alters the spreading factor to adjust to the varying channel. Experiments demonstrate that the proposed scheme improves capacity and reliability compared to other spreading factor provision techniques in dense networks [11]. A scheme to efficiently optimize the transmit power and spreading factor of the node by allocating distant nodes to different channels is proposed in the literature [12]. A genetic algorithm-based method is used to distribute the traffic over different channels, and the simulations carried out in a moderate contention scenario show reduction in the packet error rate for nodes. Two slightly complex approaches for the selection of *SF*, which have a better performance than the basic adaptive data rate technique, are proposed. The first approach uses a simple strategy to select *SF* depending on the number of devices in the network. The second approach employs an ordered water-filling method that allocates the spreading factors to equalize the transmitted packet's time-on-air (ToA) and appears robust to different operating conditions [13]. LoRa *SF* allocation using a K-means clustering algorithm allows more flexibility. Simulations show that the approach improves the coverage probability and also facilitates fair resource distribution [14]. An algorithm for network optimization based on the binary grey wolf optimizer is proposed by the authors in [15], and it minimizes the overall energy consumption in sensor networks. An innovative optimization agent algorithm based on particle swarm grey wolf optimization is proposed to achieve better energy efficiency in sensor networks [16].

One interesting approach is resource allocation using decentralized learning at the end node [17]. The end device can select different parameters, such as spreading factor, transmission power, sub-channel, etc., for each packet transmission to optimize performance in terms of energy efficiency, interference avoidance, and reliability. This approach is focused on the use of MAB algorithms. The first implementation of learning algorithms on devices in a LoRa network is proposed to reduce collisions with other devices in the ISM band. It is demonstrated on LoRa using the upper confidence bound (UCB) learning algorithm [17] for the selection of frequency channels. The experimentation conducted shows that the device's battery life can be extended by a factor of two with very low processing and memory overhead and better results than random selection. The algorithms are low-cost, and work can be extended to consider interference from other nearby gateways. Another MAB-based GNU radio implementation for IoT networks shows that intelligent objects can improve network access by using low complexity and stochastic

MAB algorithms such as UCB1 and Thompson Sampling (TS) [18]. It advocates that in stationary environments, both UCB1 and TS are efficient and converge-fast; TS gives a marginally better average performance, whereas UCB1 is faster to learn. It is also evident that the use of learning algorithms helps to accommodate more devices in a network when all the end devices are intelligent. The performance of UCB1 and TS algorithms, along with a time-frequency slotted ALOHA-based protocol is also analysed in recent works, validating a rise in successful transmissions and even in non-stationary settings [19]. A simulator is designed for resource allocation in LoRa and IoT networks using adversarial MAB, the EXP3 algorithm that considers inter-*SF* collision and capture effect [20]. This technique enhances the successful transmission rate, energy efficiency, and node lifetime. The EXP3 algorithm is limited by its long convergence time. The enhanced version, the EXP3.S algorithm, is computationally efficient and requires less convergence time than EXP3. It is a promising candidate for non-uniform device distribution in LoRa and IoT networks, although the convergence rate may worsen as the number of parameters to be selected increases [21]. Several researchers proposed stochastic and adversarial-based distributed learning algorithms, such as Updated UCB (UUCB), Updated UCB1 (UUCB1), and Updated EXP3 (UEXP3), to adapt the communication parameters of devices to the environment. Their simulation shows promising results for improving the energy efficiency and reliability of low-power IoT networks [4]. More recently, researchers have also explored the UCB algorithm for selecting the channel and different strategies based on UCB for retransmissions. The method improves the successful transmission rate in networks with a large number of devices and is equally efficient [22].

LoRa specifies the ADR algorithm for adaptive data rate, which is a centralized approach. Reinforcement learning is implemented to find the suitable parameters for reducing the power requirement in LoRa networks. The results show better throughput compared to the existing ADR method relative to energy transmission [23]. The node-based optimal selection of communication parameters is also extensively analysed, showing that different MAB algorithms outperform the standard ADR algorithm with respect to packet loss and energy requirement. The use of several such MAB learning algorithms is also studied for spectrum sensing and access in the perspective of cognitive radio. A discounted upper confidence bound-based selection algorithm is suggested for cooperative spectrum sensing that achieves better detection efficiency in a dynamic environment [24]. DUCB policy is also examined in the literature for the selection of frequency bands in a non-stationary cognitive radio scenario. A set of discount functions and exploration bonuses of the policy are considered as per the application requirements, and this policy provides lower cumulative regret, and hence, improved performance [25]. The literature shows the use of DUCB for cognitive radio applications. We have investigated the DUCB algorithm with various discount functions and exploration bonuses for LoRa networks, and a discount function specific to the requirements of LoRa is proposed and evaluated. The proposed algorithm performs better and has less complexity compared to the other known algorithms.

3. LoRa Technology

Long range (LoRa) is a proprietary technology by Semtech, which uses the chirp spread spectrum modulation method [10]. The default medium access control (MAC) protocol, LoRaWAN [1], is usually used with the LoRa networks. Chirp spread spectrum modulation has low-power characteristics similar to frequency-shift keying modulation, but provides a better communication range. The available LoRa transceivers work at a 137 MHz to 1020 MHz range of frequencies. Thus, they can work in licensed bands but are usually used in unlicensed ISM bands, such as 433 MHz, 868 MHz, and 915 MHz [26]. As illustrated in Figure 1, in a LoRaWAN network, the data transmitted by an end device can be received by multiple gateways in the neighbourhood. Each gateway forwards the packet from the end device to the cloud-based network server using some backhaul (either satellite, cellular, Ethernet or Wi-Fi). The network server manages the intelligent and

complex tasks, such as removing redundant packets, sending acknowledgements through the suitable gateway, and performs adaptive data rates. Handover is not required from one gateway to another, even for mobile nodes. Nodes in the LoRa network are asynchronous and communicate when they have data to send, using pure ALOHA, which saves energy. The LoRa network uses ADR and a multi-channel multi-modem transceiver in the gateway, thus ensuring good network capacity. The capacity depends on the number of channels, data rate, and how often the nodes transmit. Different spreading factors lead to orthogonal signals and changes in the data rate. Thus, the gateway can receive multiple different data rates on the same channel simultaneously [1,27]. Several alternatives to increase coverage and data rates and avoid interference in LoRa networks are also being explored [28].

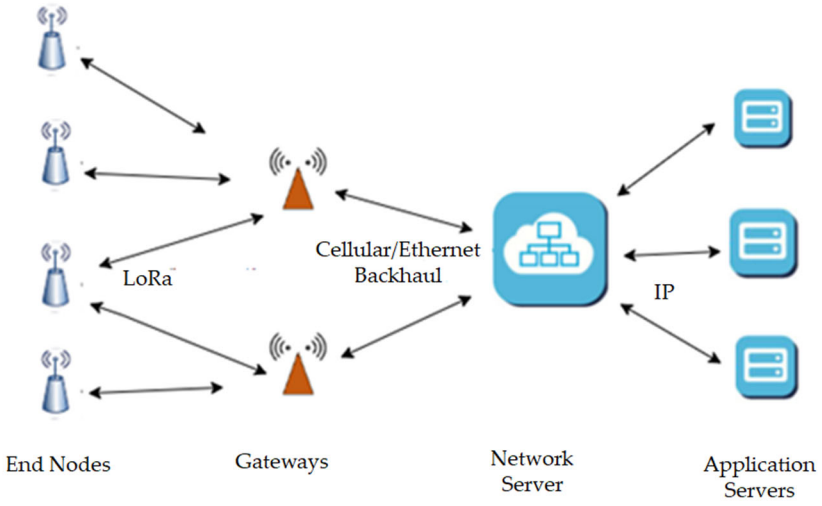

Figure 1. LoRa network architecture.

Any LoRa device can be configured for different parameters such as spreading factor, transmission power, carrier frequency, bandwidth, and coding rate [10]. As per the regulations, transmission power can be changed approximately in stages of 1 dB from 2 dBm to 17 dBm. The ratio of symbol rate to chip rate is termed as a spreading factor, and the higher the spreading factor, the more SNR, sensitivity, range, and also packet airtime. *SF* can be selected as any value from 7 to 12 [27]. A typical LoRa network works at a 125 kHz, 250 kHz, or 500 kHz bandwidth. A higher bandwidth gives a higher data rate, but the sensitivity reduces. The LoRa modem uses forward error correction (FEC) with a coding rate that can be set to $4/5$, $4/6$, $4/7$, or $4/8$. A higher coding rate means a better guard against errors but increases time-on-air.

The average energy needed to transmit a LoRa packet is given by

$$E_{avg} = P_t T_{pkt}\ N_p \tag{1}$$

where P_t is transmission power, T_{pkt} is the time required to transmit a packet, and N_p is the number of transmissions required to send a packet successfully.

The transmission power, the time for transmitting a packet, and the number of transmissions required for successful packet transmission are the important parameters for the improvement of network performance. The number of retransmissions is reduced if interference or collision are reduced. For LoRa, with an increase in the spreading factor (*SF*), the sensitivity improves, the need for retransmissions reduces, and then the average energy required for transmitting a packet also reduces. Based on the LoRa packet format, the time required to transmit a packet or ToA is given as [27]

$$T_{pkt} = (n_p + 4.25)\frac{2^{SF}}{BW} + \left(8 + max\left(ceil\left(\frac{8PL - 4SF + (28 + 16C) - 20H}{4(SF - 2DE)}\right)(CR + 4), 0\right)\right)\frac{2^{SF}}{BW} \tag{2}$$

where n_p is the number of programmed preamble symbols, PL is packet payload, H is 0 when the header is present and 1 when the header is absent, and DE is 1 when low data rate optimization is enabled and 0 when low data rate optimization is disabled. Cyclic redundancy checksum (CRC) is by default enabled (C = 1), so the term becomes 44, whereas if it is disabled (C = 0), the term becomes 28. The CRC field is present only in uplink transmissions.

An LPWAN network can be a single gateway or multi-gateway network comprising LoRa and non-LoRa nodes. LoRa uses a chirp spread spectrum with quasi-orthogonal spreading factors (SF). The typical SF values are 7, 8, 9, 10, 11, or 12. Interference can be from other LoRa nodes using the same SF (Co-SF), other LoRa nodes using different SF (Inter-SF), and other nodes using the same carrier frequency but not LoRa technology.

At the gateway, a signal is detected when the ratio of received signal to interference plus noise is greater than the receiver sensitivity for a particular SF at the desired LoRa node. For the high value of SINR, the interference power needs to be low, and the signal power should be high. For low-energy consumption, the signal power should be less. From the above equations, it is implicit that as SF increases, sensitivity improves, and the SINR required also decreases. For lower SF, ToA, average energy, and throughput are also low. As the network size N increases, the number of devices with the same SF would increase, and hence the probability of a successful transmission drops. There is always a trade-off between energy efficiency and interference avoidance. This paper studies and analyses the selection of SF using the MAB algorithm.

4. Multi-Armed Bandit Algorithms

Artificial intelligence (AI) is a rapidly progressing technology that enables intelligence in machines with the capability to inevitably learn from experiences without being explicitly programmed. It banks on the concept that machines can learn from previous data and make decisions using algorithms. Reinforcement learning (RL) is a form of machine learning in which an AI agent is trained by commands, and on each action that it takes, an agent receives a reward. The agent receives a positive reward for every positive action and a negative reward for a wrong action. Thus, the agent learns from its environment using this feedback, decides its next course of action, and in turn, improves its performance. The main aim of an agent in reinforcement learning is to obtain maximum positive rewards, thus improving performance.

Multi-armed bandit (MAB) is a machine learning structure based on reinforcement learning in which an agent has to select actions or arms to maximize its cumulative reward. In MAB, the player has a collection of k-arms. For every try, the player has to pick an arm, and a reward is received according to the selected arm, irrespective of the reward received if another arm has been chosen. An action is explored or performed multiple times, and based on the mean rewards obtained from the actions, further actions are exploited or performed to maximize the reward. Regret is the difference between the cumulative mean rewards and the reward that may have been obtained using an optimal policy, and the policy aims to lower regret.

Depending on the reward model, MAB problems can be further classified as stochastic and adversarial. In stochastic MAB problems, the rewards follow the stochastic distribution. Stochastic MAB can be further classified as stationary and non-stationary. Stochastic stationary indicates that the stochastic distribution of rewards is stationary. Stochastic stationary MAB assumes that every arm is associated with constant and unknown distribution, and rewards are independently generated. Thompson sampling (TS) and upper confidence bound (UCB) are examples of commonly used stochastic stationary algorithms [19,29]. Stochastic non-stationary implies that the rewards follow a non-stationary stochastic distribution. Non-stationary MAB algorithms consider a practical situation that rewards from the same arm at different times may not be the same. There are different algorithms or policies, such as ε−greedy, EXP3, DUCB, which can handle the stochastic non-stationary MAB problem [29,30].

Let us consider the selection of the spreading factor parameter in LoRa using a distributed learning algorithm using the MAB model for LoRa, as shown in Figure 2. This approach assumes the end device to be intelligent. From the given set of *SF*s, the end device has to select an *SF* or a strategy $s(t) = \{SFs\}$. The devices do not know their position or channel condition. Hence, the device may select any *SF* belonging to the set ϵ, $s \in S$. At every packet arrival time t, each end device chooses a strategy $s(t)$, depending on a certain distribution over S, which gives a reward of $r_{s(t)} \in \{0, 1\}$. Once the device selects a particular value of *SF* and transmits a packet, it can result in the transmission being successful or unsuccessful. If the LoRa gateway receives the packet successfully, it sends an acknowledgement to the device. This is analogous to a multi-armed bandit problem. The *SF* value can be modelled as the arm or action, and the state of *SF* (that means whether choosing that *SF* can result in successful transmission and receiving an acknowledgement or not) can be modelled as the reward. If the end device receives an acknowledgement, it can be said that it receives a reward = 1, otherwise the reward = 0. The end device utilizes only the locally available information that is the received acknowledgement, and selects an optimal value of *SF*, which encounters the least collisions. As the end nodes exhibit a dynamic nature, the *SF* selection problem can be modelled as a non-stationary MAB problem.

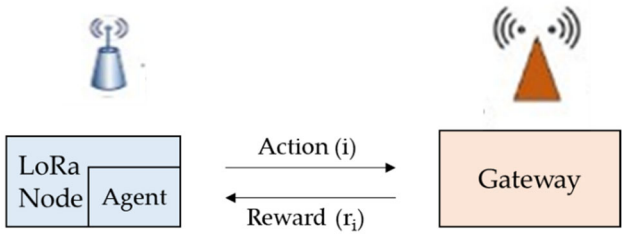

Figure 2. MAB model for LoRa.

4.1. Upper Confidence Bound Algorithm

The upper confidence bound (UCB) algorithm [22] first initializes by selecting each arm once, and builds the upper confidence bound or index for each arm. At every turn, it chooses the arm with the current maximum bound. It updates the confidence bound for that arm and devises a sequence for selecting the arm to maximize the rewards. The main idea is to augment the average reward value with a bias factor. UCB1 is a variation of UCB when the design parameter in the bias factor is chosen as 2 [22,29]. Upper confidence bound tuned (UCB-T) uses empirical variance in the bias factor, thus reducing the exploration to arms with small reward variance. An improved upper confidence bound algorithm also considers the effect of empirical variance [26].

4.2. Discounted Upper Confidence Bound Algorithm

TS and UCB are suitable for stationary MAB scenarios, and recent algorithms, such as DUCB, are applicable for solving a non-stationary problem. The UCB algorithm can be modified for the non-stationary problem by using a discounting factor [25]. In the discounted UCB policy, the most recent plays are given more weight using a discount factor, which averages past rewards. This policy seems suitable for time-varying wireless environments. Hence, the DUCB policy can also be optimized by utilizing the exploration bonus and the discount function to adapt to the complex and varying LoRa environment [25,31]. The core index of the DUCB policy selection arm is given as

$$U_k(t) = X_k(t) + B_k(t) \tag{3}$$

where $X_k(t)$ is used to predict the exploitation by discounted averages and $B_k(t)$ is the exploration bonus [20]. If the power function, which is described as $f(x) = \gamma^x$, is used as the discount function, then $X_k(t)$ can be written as

$$X_k(t) = \frac{\sum_{s=1}^{t} \gamma^{t-s} X_t^k 1_{i_s=i}}{\sum_{s=1}^{t} \gamma^{t-s} 1_{i_s=i}} \tag{4}$$

where $X_k(t)$ is the average reward of arm k at time step t, s is the sample, γ^{t-s} denotes the discount function, 1_{i_s} is the indicator function that has a value of 1 if i_s is true and of 0 if i_s is false.

The exploration bonus $B_k(t)$ is given as

$$B_k(t) = 2\mathrm{B}\sqrt{\frac{\xi, \left(\log \sum_{i=1}^{k} Ni(t)\right)}{N_k(t)}} \tag{5}$$

where N is the maximum number of trials, I is the index of the arms, X_k is the average reward for arm k, N_k is the number of times arm k is chosen, ξ is the bias parameter and $N_k(t) = \sum_{s=1}^{t} \gamma^{t-s} 1_{i_s=i}$.

Another exploration bonus based on statistical variance is also introduced and explored [25] given as

$$B_k(t) = \xi, \sqrt{\frac{X_k(t) - X_k(t)^2}{N_k(t)}} \tag{6}$$

where $X_k(t) - X_k(t)^2$ is the statistical variance of each arm reward and ξ is the bias parameter.

The recent plays give higher weight by using the appropriate discount function to weight data. A monotonically decreasing function can be designed for the given problem. The discount function needs to be chosen such that it is appropriate according to the application scenario. Some of the popular discount functions include the exponential function, power function, window function, and others. An arm that has not been explored for a long time is explored by using an appropriate exploration bonus. Exploration bonus can also be considered as a generally monotonic decreasing function and should be adjusted to ensure the exploration of the optimal arm when the rewards change.

5. Policy with New Discount Function and Exploration Bonus

LoRa devices are low-power and usually work at low data rates. LoRa uses the unlicensed band, and the devices have to follow the duty cycle constraints imposed as per the region of operation. The duty cycle may be 1% or 0.1%, limiting the number of transmissions per device per day. Hence, the adaptive parameter selection process in LoRa needs to be less complex and fast. We propose a modified DUCB algorithm, which aims to reduce the algorithm complexity and a new discount function for LoRa necessities [32].

In the DUCB policy, the discount function decides the weights assigned to samples, and hence, the discount function should vary as per the application scenario. The discount function $y = \left[\frac{\mathrm{N}-\mathrm{x}}{\mathrm{N}}\right]^a$ is appropriate for a scenario where changes are frequent, such as cognitive radio systems [21]. As LoRa nodes have a restricted duty cycle as well as low data rate capabilities, we have designed a new discount function, which is as follows

$$y = \left[\frac{\mathrm{N}-\mathrm{x}}{\mathrm{N}}\right]^{1/a} \tag{7}$$

This function is also characterized as a monotonically decreasing function; however, it is suitable for a scenario where changes are less frequent discount functions, such as UCB-E, UCB-P-3, UCB-L, UCB-P-1/3, and UCB-P-3/4, exploration bonuses, such as UCB-1 and UCB (with variance), were studied and a novel policy is proposed with a new discount function and exploration bonus. The policies with different existing [25] and proposed

discount functions are as mentioned in Table 1. Figure 3 depicts the varying rate of change of various discount functions. These different discount functions are monotonically decreasing and decide the weights assigned to recent plays indicating the exploitation of a particular arm. The exploration bonus decides the way the arms are explored by the policy. The policies with various existing [25,33] and the proposed exploration bonuses for DUCB are as mentioned in Table 2.

Table 1. Various discount functions for DUCB.

Policy	Discount Function
UCB-E	$y = \gamma^x = 0.9982^x; 0 < \gamma < 1$
UCB-P-3	$y = ((N-x)/N)^3$
UCB-L	$y = (N-x)/N$
UCB-P-1/3	$y = ((N-x)/N)^{1/3}$
UCB-P-1/2 (Proposed)	$y = ((N-x)/N)^{1/2}$
UCB-P-3/4	$y = ((N-x)/N)^{3/4}$

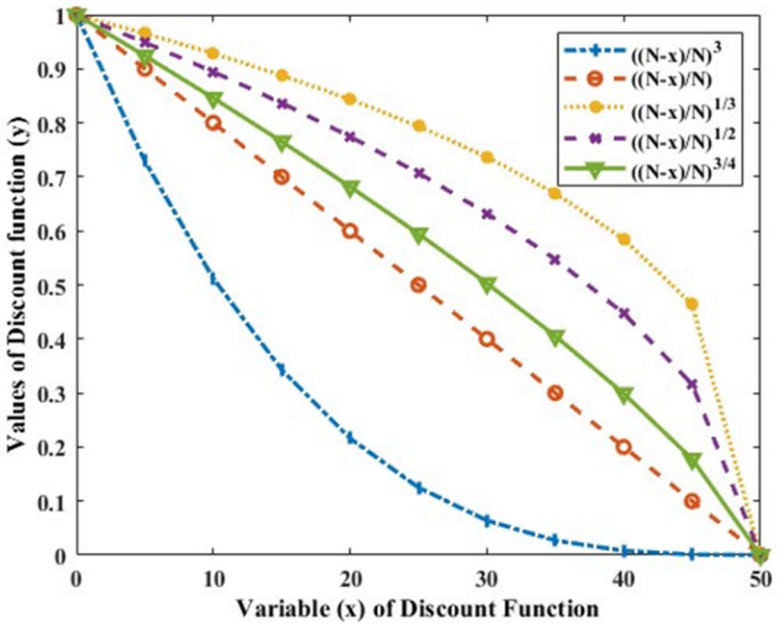

Figure 3. Various discount functions.

Table 2. Various exploration bonuses for DUCB.

Policy	Exploration Bonus
UCB-1	$\sqrt{\frac{2log(t)}{N_k(t)}}$
UCB (with variance)	$\sqrt{\frac{X_k(t)-X_k(t)^2}{N_k(t)}}$
UCB-O (Proposed)	$0.5\sqrt{\frac{(X_k(t)-X_k(t)^2)}{N_k(t)}}$

A modified DUCB policy for LoRa, UCB-P-1/2+O, is proposed in this paper (Algorithm 1). The core index of this policy is as given.

$$U_k(t) = X_k(t) + 0.5\sqrt{\frac{\left(X_k(t) - X_k(t)^2\right)}{N_k(t)}} \quad (8)$$

where $X_k(t)$ is the discounted average with the discount function as $\left[\frac{N-x}{N}\right]^{1/2}$

Algorithm 1 Proposed UCB-P-1/2+O

Inputs: Discount function *f(x)*, exploration bonus *B(x)* as per the policy.
Output: Received rewards.
1: Initially select each action once
2: For every trial $t = k + 1, k + 2, k + 3, \ldots$:
3: Set $U_k(t) = X_k(t) + B_k(t)$
4: Select action i_t = argmax$U_k(t)$
5: Receive the reward $X_k(t) \in [0,1]$
6: For all the arms, $k = 1, 2, 3, \ldots K$ set:
7: $X_k(t) = \frac{\sum_{s=1}^{t} f(s) X_t^k 1_{i_s=i}}{\sum_{s=1}^{t} f(s) 1_{i_s=i}}$
8: $B_k(t) = B(t)$

Figure 4 illustrates the flowchart for the simulation of the proposed DUCB-P-1/2+O policy that an intelligent node can utilize for the selection of transmission parameter *SF*. Initially, the node selects any *SF* value from the set {*SF*} and sends the packet to the gateway using this transmit parameter. If the packet is successfully received at the gateway, it sends an acknowledgement and the reward equals 1. In case the packet is not successfully received, there is no acknowledgement and the reward equals 0. Accordingly, the DUCB-P-1/2+O policy is updated and the next selection is performed as per the updated policy.

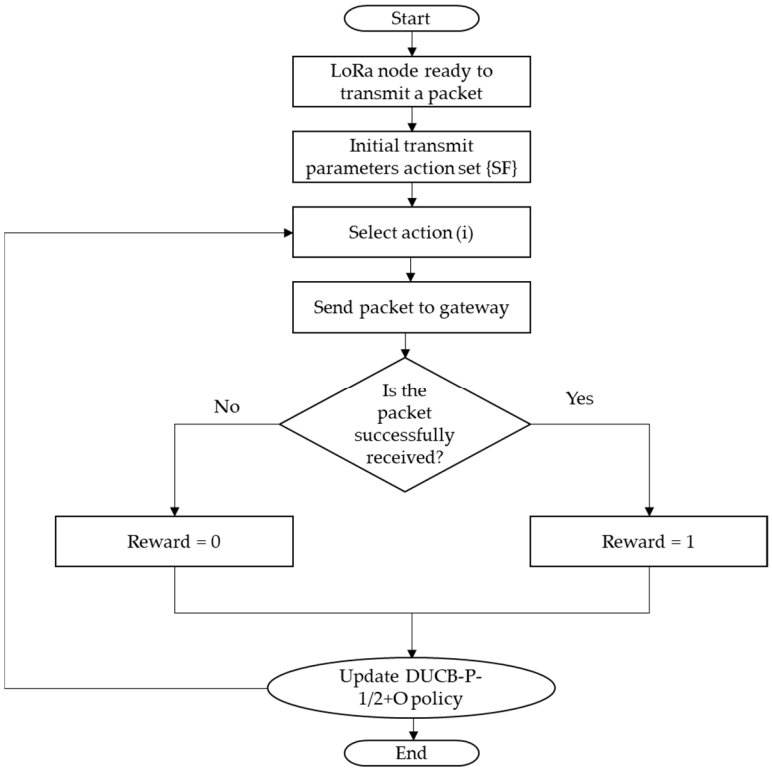

Figure 4. Proposed LoRa node transmit parameter selection using DUCB-P-1/2+O policy flowchart.

6. Results and Discussion

Spreading factor (*SF*) selection greatly impacts the performance of the LoRaWAN network, and hence, we have considered the *SF* selection problem. It is assumed that in a homogeneous LoRaWAN setting, there is a single gateway with multiple nodes operating on a single channel with constant power. Each node transmits packets with a certain *SF* value, unaware of the *SF*s used by other nodes. If a gateway receives the packet, it sends

an acknowledgement to the node (reward = 1). The node retransmits only if it does not receive an acknowledgement (reward = 0) from the gateway. A node can select one of the *SF*s from the available set of *SF*s; this is the selection of an action or arm as in an MAB algorithm. Simulations are carried out using a dataset of different *SF* values and mean rewards for a different number of trials. In every trial, an *SF* value is chosen as per the strategy in the proposed DUCB MAB algorithm. Mean rewards are computed for every policy with different combinations of discount functions and exploration bonuses, as given in Tables 1 and 2.

In this section, we evaluate the UCB-P-1/2+O policy and compare it with other policies that are used to handle the LoRa *SF* selection problem. Along with *SF*, other parameters, such as transmission power *Pt* and different channels, can also be chosen using the proposed policy to improve the network performance. As per the LoRa specifications, spreading factor *SF* = {7, 8, 9, 10, 11, 12} can be considered. For the simulations, the case considered is the selection of only *SF*, keeping the transmit power and channel the same, resulting in six actions or arms as per six *SF* values. This algorithm can very well be extended for the selection of multiple LoRa parameters simultaneously.

During the simulation, a few assumptions are made for better modelling and analysis of the LoRa network environment. The first assumption is that the policy action does not affect the reward change of any action. The second assumption is that the actions are independent of each other; the state or distribution of each action does not affect those of the other actions. The parameters of the simulation data are set as the number of actions k to be 6 and the maximum number of trials or time steps to be 50. Since LoRa follows the duty cycle limitations, the LoRa transmissions are less frequent [28], and hence the algorithms are evaluated over a lesser number of trials but with sufficient iterations to support the observations. Similar results are also obtained when the number of trials are further increased.

For analyzing the proposed strategies for different network scenarios, three different datasets representing three different scenarios are designed [25]. These datasets are designed considering the variation in the mean rewards of the different actions for a few practical situations. In Scenario A, the mean rewards of all the actions do not vary, and there is only one optimal action for the entire time. This is a representation similar to a stable or stationary situation. Although it may seem a simpler situation than most of the practical scenarios, the evaluation of the algorithm over this setup is relevant for the case of sparse and distant LoRa nodes in a network. This dataset with mean rewards for six actions is illustrated in Figure 5. In Scenario B, the mean rewards of multiple actions change at the same time, which represents a more practical LoRa situation. The dataset with mean rewards for six actions, where the mean rewards change four times and the optimal action changes five times, is shown in Figure 6. Scenario C represents the setting where the mean rewards of all the six actions are constant, except for one action for which it changes. This represents a situation where the mean rewards for an action decrease when they are already busy and increase when they become free. This setup is relevant from the point of view of a small network with a smaller number of nodes. Additionally, the optimal action changes over time, as depicted in Figure 7. The mean reward of Action 6 changes two times and the optimal action changes once, from Action 6 to Action 5.

To study the effect of the discount factor for a constant exploration bonus, the discount function is varied and mean rewards are calculated. The simulation results show that the mean rewards are consistently improved for the proposed UCB-P-1/2 with discount function compared to other discount functions mentioned in Table 1. To study the effect of the exploration bonus, keeping the discount factor constant, the exploration bonus is varied and the mean rewards are calculated. It is observed that the mean rewards are higher for the proposed UCB-O policy than other policies mentioned in Table 2. Furthermore, the combinations of several discount functions and exploration bonuses were evaluated, and the proposed UCB-P-1/2+O policy was formulated as it resulted in better mean rewards for the simulated scenarios.

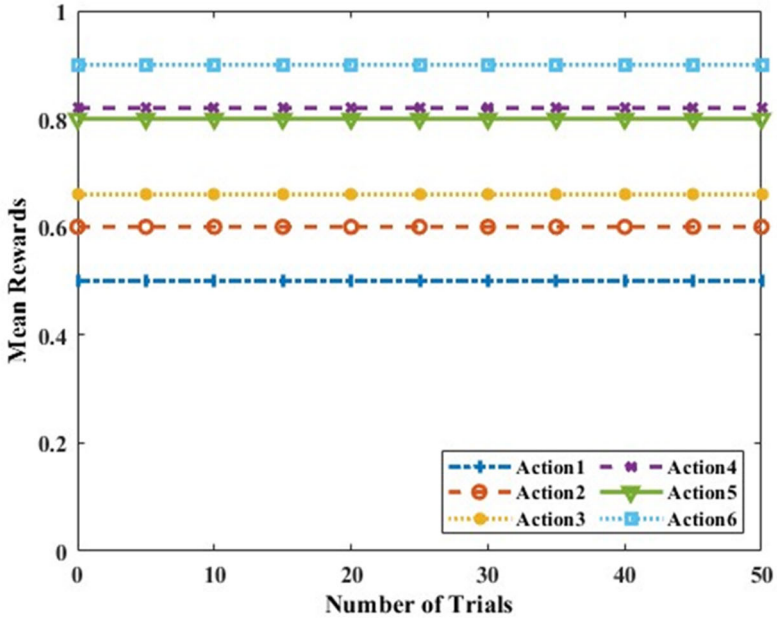

Figure 5. Dataset for Scenario A, where mean rewards of all the actions remain constant.

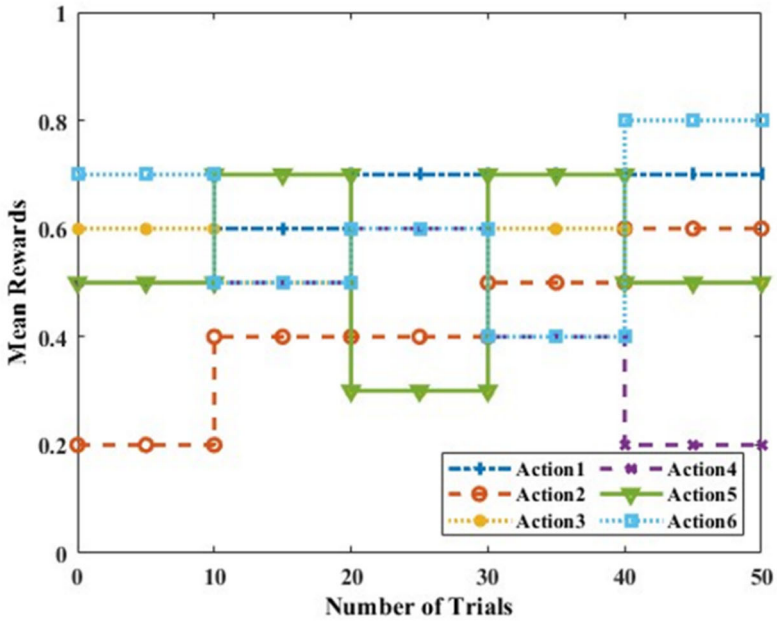

Figure 6. Dataset for Scenario B, where mean rewards of multiple actions change simultaneously.

The proposed algorithm is then evaluated and compared with other algorithms used in the literature for similar applications for scenarios, as discussed above, and for two different cases. The first case is a situation where just a single node uses the MAB algorithm, while the second case is when multiple nodes simultaneously use the MAB algorithm. The second case is a better representation of a practical situation in a network where multiple nodes transmit simultaneously in an intelligent manner. A few results are shown below for different policies, such as random sampling (RS), Thompson sampling (TS), UCB, DUCB, and the proposed DUCB –P1/2+O algorithm.

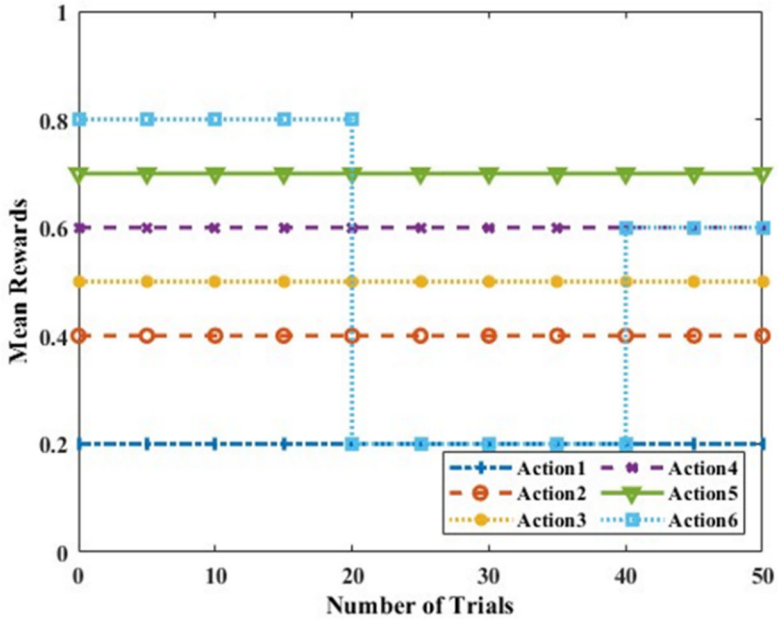

Figure 7. Dataset for Scenario C, where mean rewards of the actions are constant, except for one.

Case 1: Six actions for a different number of trials and Scenarios A, B, and C for a single intelligent node using the MAB algorithm.

Figure 8 shows the mean rewards for Scenario A with six actions and a single intelligent node, and it is seen that the proposed UCB-P-1/2+O algorithm gives an average better mean reward compared to other methods. Thompson sampling also gives better mean rewards, but the increase is not uniform. DUCB and UCB-P-1/2+O have similar trends when the number of trials is smaller, and UCB-P-1/2+O outperforms DUCB as the number of trials increase. It is also essential to observe the algorithm behaviour with respect to the execution time. Figure 9 shows the execution time of these algorithms. It is observed that the proposed UCB-P-1/2+O requires less time than DUCB and much less than TS. For Scenario A, it can be inferred from Figures 8 and 9 that the proposed UCB-P-1/2 +O algorithm is faster, yields more rewards, and is suitable if both time and reward criteria need to be satisfied.

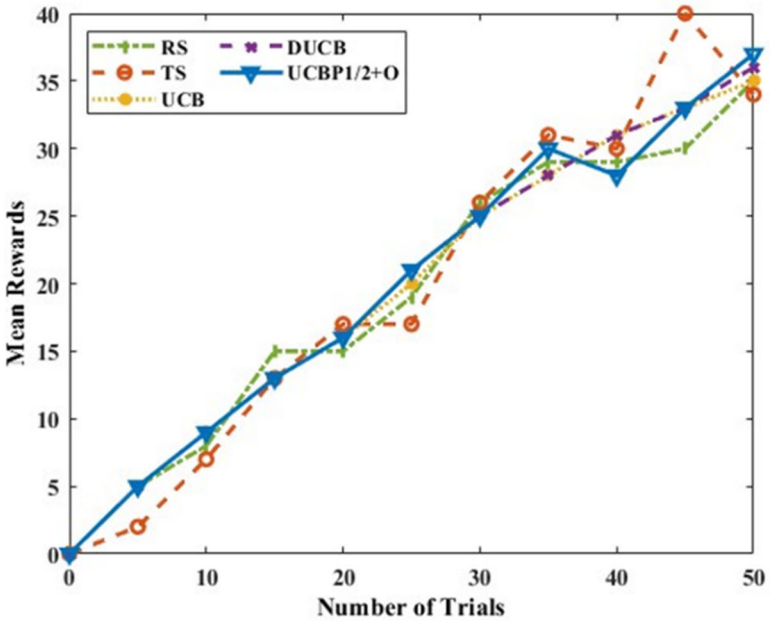

Figure 8. Mean rewards for Scenario A with six actions and a single intelligent node.

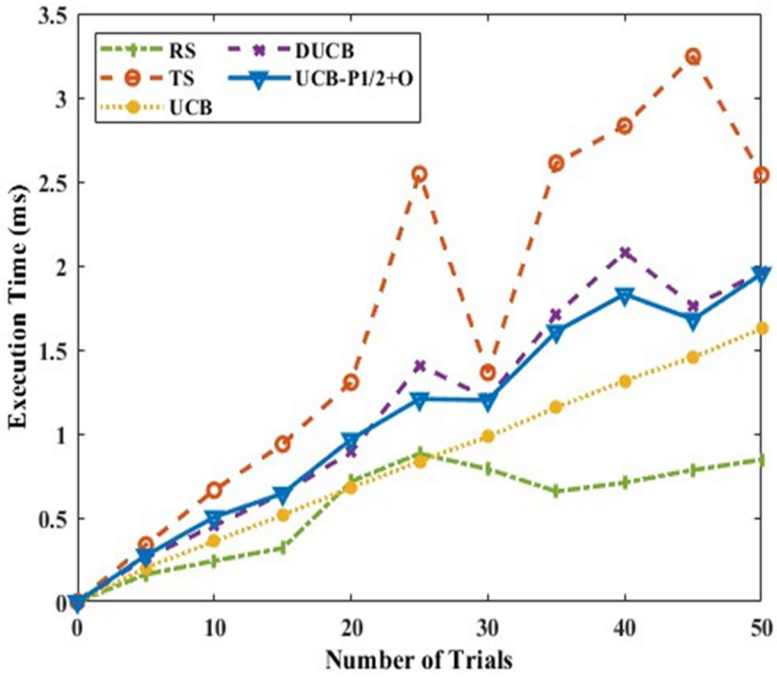

Figure 9. Execution time for Scenario A with six actions and a single intelligent node.

The mean rewards for Scenario B with six actions, as shown in Figure 10, indicate that UCB-P-1/2+O gives better mean rewards throughout the range of trials compared to other policies. In this situation, DUCB fairs less and TS does not perform satisfactorily. In terms of execution time, TS is the slowest and RS is the fastest, but RS does not ensure better rewards, as seen in Figure 11. UCB-P-1/2+O shows slow execution initially and becomes faster as the number of trials increases. Thus, for Scenario B, when both the criteria, less execution time and more rewards need to be met simultaneously, UCB-P-1/2+O outperforms the other policies.

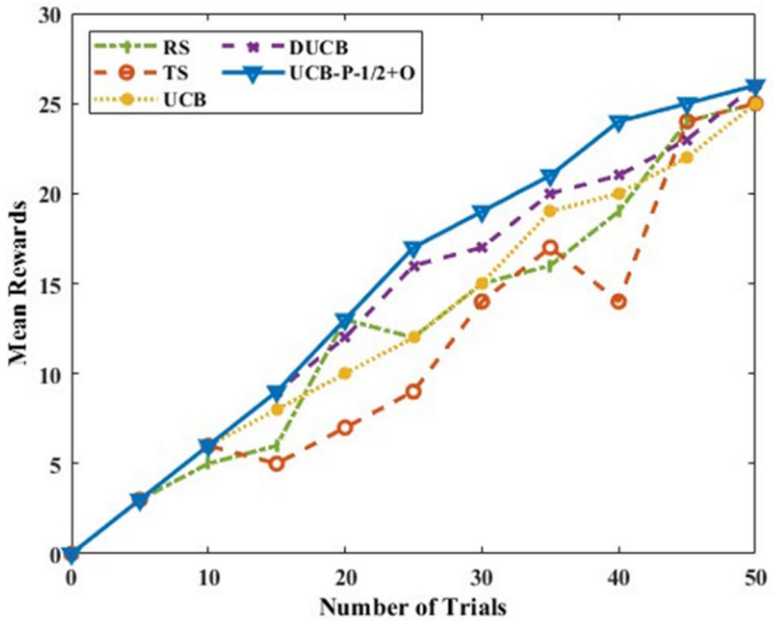

Figure 10. Mean rewards for Scenario B with six actions and single intelligent node.

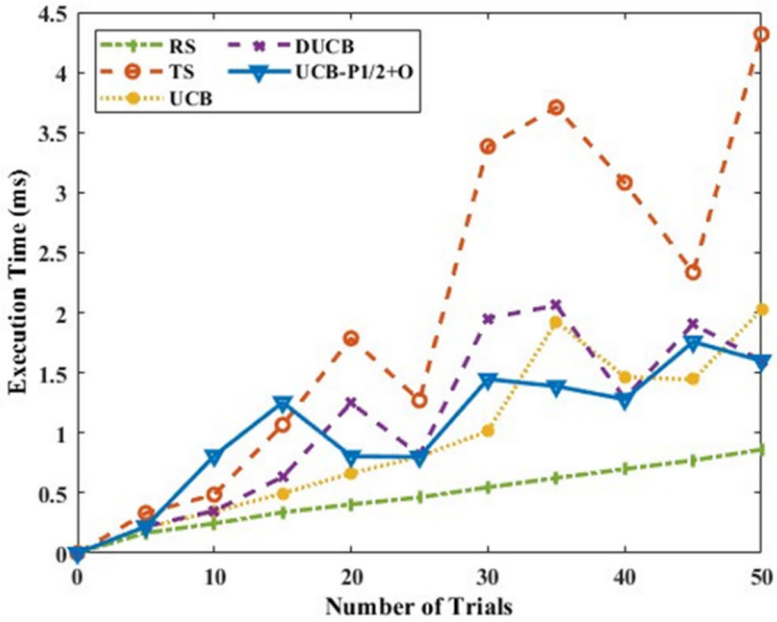

Figure 11. Execution time for Scenario B with six actions and single intelligent node.

Figure 12 shows the mean rewards for Scenario C and six actions; it is observed that UCB-P-1/2+O gives marginally better mean rewards than the remaining policies. TS performs better but is not consistent in this scenario as it was earlier. Figure 13 shows the execution time with respect to the number of trials for different algorithms. The execution time is slightly higher than UCB in some cases and comparatively less than TS, with the advantage of better mean rewards.

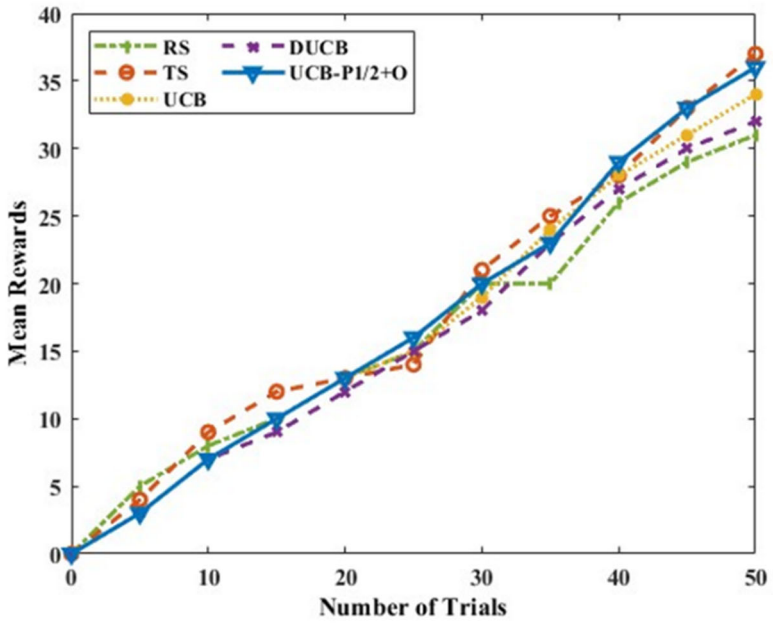

Figure 12. Mean rewards for Scenario C with six actions and a single intelligent node.

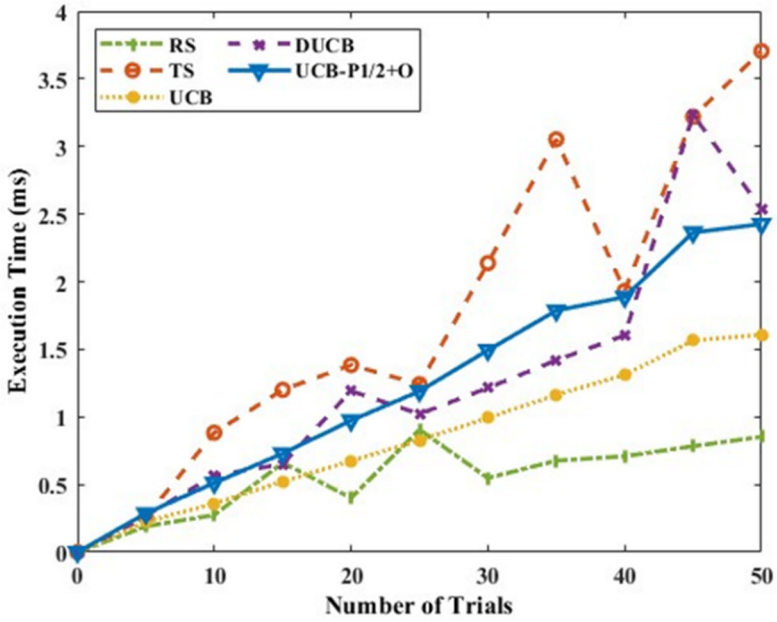

Figure 13. Execution time for Scenario C with six actions and a single intelligent node.

From the simulations, it is observed that for Scenarios A and B, the UCB-P-1/2 + O policy gives approximately 4% and 8% better mean rewards than the DUCB policy for the number of trials to be 50, and for the similar case of 40 trials in Scenario C, UCB-P-1/2 + O shows up to 8% increase in the number of mean rewards compared to DUCB. From the results obtained for various cases and scenarios, it can be concluded that for a single intelligent node, the proposed UCB-P-1/2+ O policy outperforms the other policies and gives better mean rewards compared to other policies, keeping the execution time fairly less.

Case 2: Six actions for a different number of trials and Scenarios A, B, and C for multiple intelligent nodes using the MAB algorithm. Here, five intelligent nodes are considered and mean rewards are computed.

Figure 14 displays the mean rewards for Scenario A with six actions and multiple intelligent nodes. It is observed that the mean rewards increase with the increase in the number of trials for all the different policies. It is seen that the proposed UCB-P-1/2+O gives better and consistent mean rewards compared to other methods. TS also performs similar to all the other algorithms, resulting in lesser rewards. Figure 15 shows the execution time of the algorithms, and it is observed that the time required for the proposed UCB-P-1/2+O policy is similar to the time required for DUCB for a few cases, and it is much less than the time required for TS, as the number of trials increases. This brings us to the similar conclusion that UCB-P-1/2+O gives a better reward and execution time outcome.

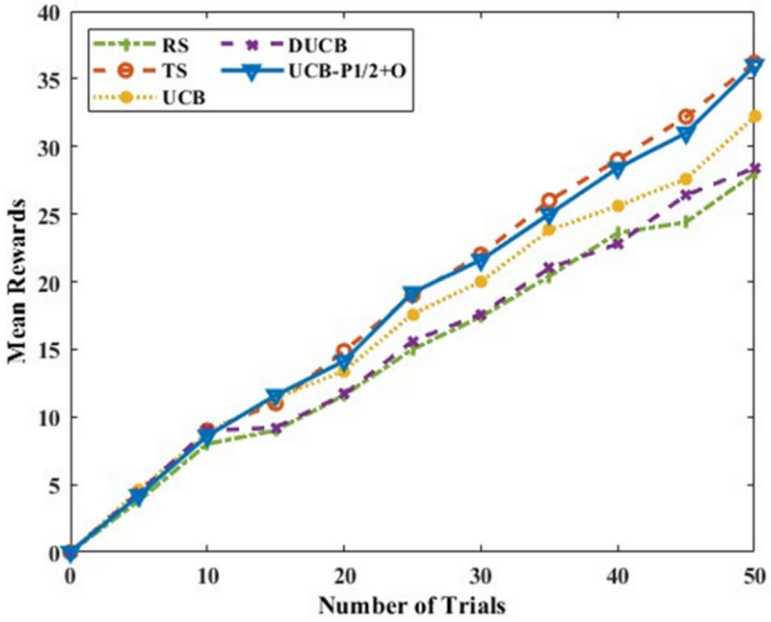

Figure 14. Mean rewards for Scenario A with six actions and multiple intelligent nodes.

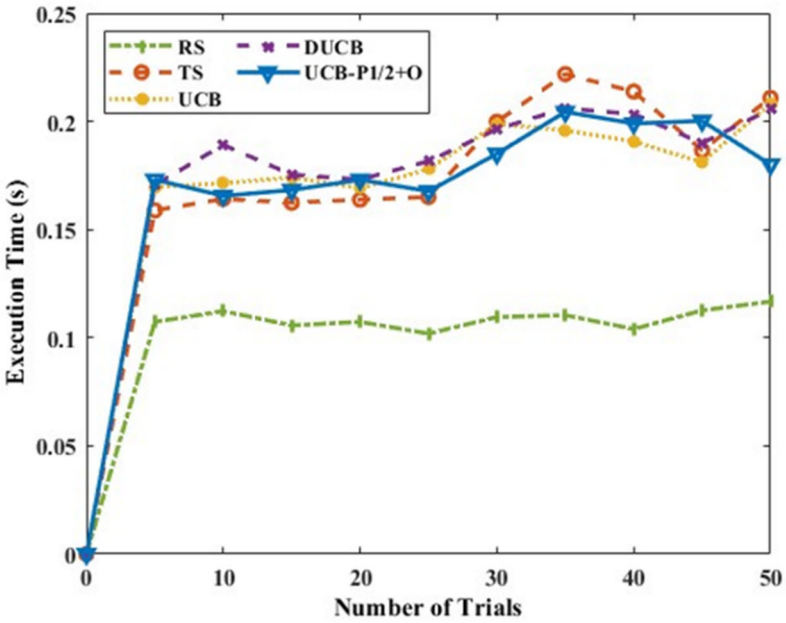

Figure 15. Execution time for Scenario A with six actions and multiple intelligent nodes.

The mean rewards for Scenario B with six actions and multiple intelligent nodes, as shown in Figure 16, indicate that UCB-P-1/2+O gives better mean rewards compared to other policies, although the difference is fairly small. DUCB also performs similar to the proposed algorithm. The execution time of UCB-P-1/2+O is even less than DUCB in a few cases, as in Figure 17, but UCB-P-1/2+O gives better mean rewards too. Although the execution time of RS appears to be less, the number of rewards obtained is not consistent and varies with trials. Based on Figures 16 and 17, similar conclusions can be drawn, namely that UCB-P-1/2+O is faster and better.

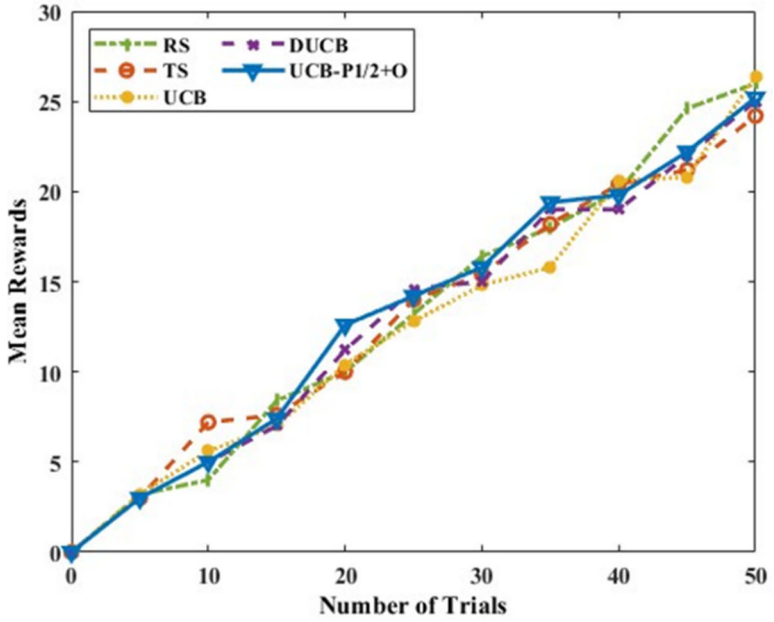

Figure 16. Mean rewards for Scenario B with six actions and multiple intelligent nodes.

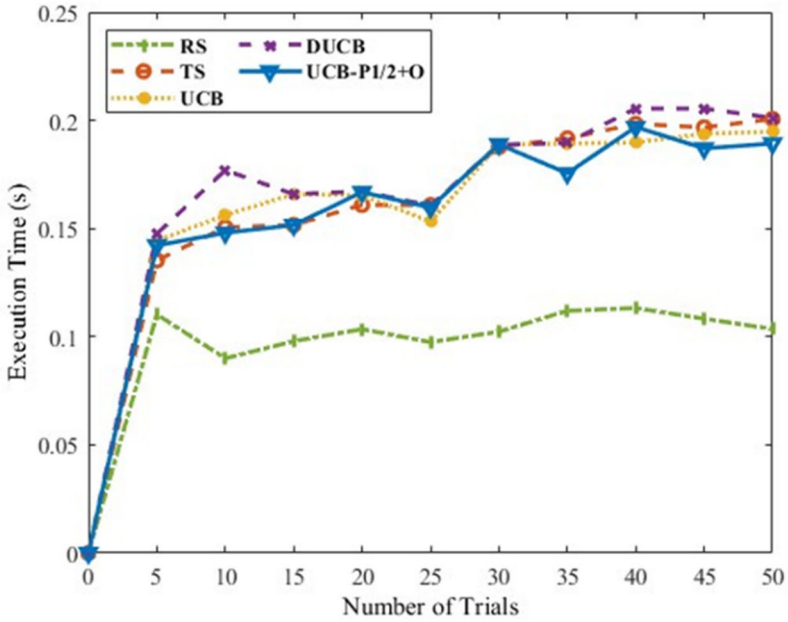

Figure 17. Execution time for Scenario B with six actions and multiple intelligent nodes.

Figure 18 displays the results of mean rewards for Scenario C with six actions and multiple intelligent nodes. It is observed that UCB-P-1/2+O gives better mean rewards than the remaining policies, with TS also performing better. The execution time is comparatively less than DUCB, UCB, and TS for most of the cases, as seen in Figure 19, with RS requiring the least time, as expected. Figures 18 and 19 lead to similar inferences that if both fast execution and better rewards are expected, UCB-P-1/2+O is suitable.

From the simulations, it is observed that for Scenarios A and B, the UCB-P-1/2+O policy gives approximately 15% better mean rewards than the DUCB policy for the number of trials to be 50, and similarly, for the lower values of the number of trials some increase in the number of mean rewards is observed. From the results obtained for various cases and scenarios, it can be concluded that for multiple intelligent nodes, the proposed UCB-P-1/2+O policy outperforms other policies and gives better mean rewards compared to other policies, keeping the execution time fairly less.

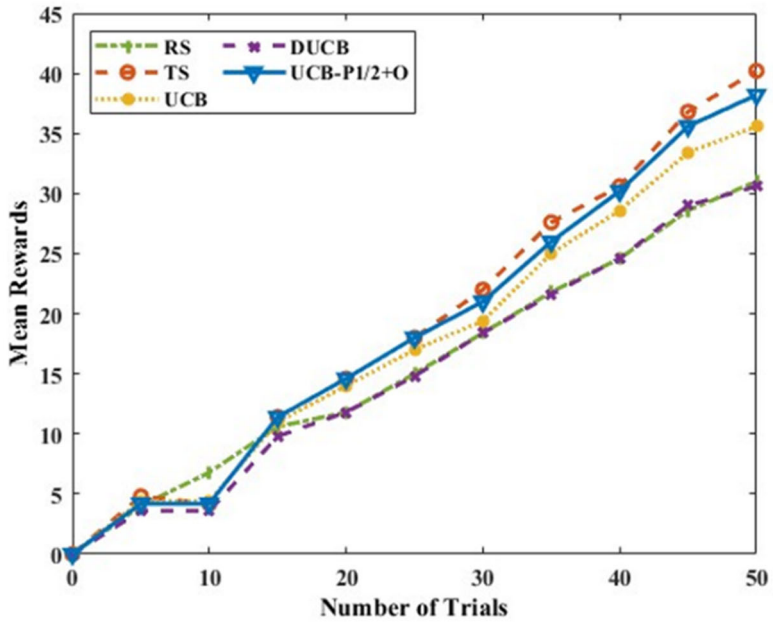

Figure 18. Mean rewards for Scenario C with six actions and multiple intelligent nodes.

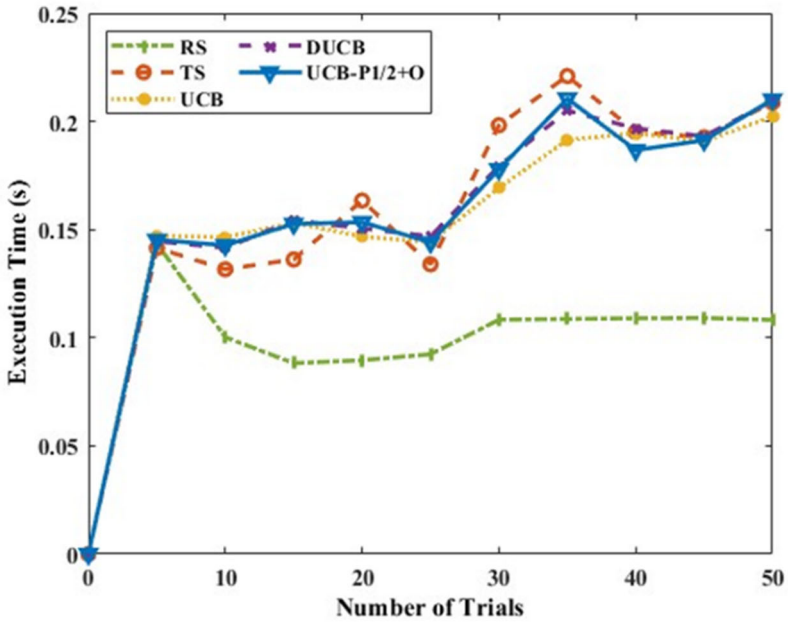

Figure 19. Execution time for Scenario C with six actions and multiple intelligent nodes.

The proposed discount function designed in this paper is as per the LoRa requirements and gives maximum mean rewards or, effectively, more successful transmissions. With LoRa being low-power, the algorithm used needs to be less complex and fast. So, we compare the proposed UCB-P-1/2+O policy with the traditional DUCB policy based on algorithm complexity. The traditional DUCB policy uses an exponential discount function increasing the time complexity. It can be inferred that since the proposed policy does not include an exponential discount function, it has an additional benefit of less time complexity than the traditional DUCB policy. The simulations show that the execution time required is also comparatively less, further proving the benefit of the use of this policy for LoRa transmissions.

7. Conclusions

LoRa parameter selection plays a significant role in determining the LoRaWAN network performance. Resource allocation in LoRa is a complex task as it involves choosing from a large set of transmitter parameter configurations. For the optimal parameter selection, we designed a modified DUCB policy, UCB-P-1/2+O, with a new discount function and exploration bonus for LoRa transmissions. To evaluate the proposed policy, simulations based on varied scenarios are carried out. The findings suggest that the proposed policy outperforms the other studied methods in the literature not only in terms of mean rewards but also in regard to execution time, making it a promising lightweight optimal solution. The selection of transmission parameters using this policy can contribute to interference avoidance, thus improving the energy efficiency and performance of the LoRa networks. Thus, the decentralized learning approach used by this policy is advantageous over the centralized one, or the traditional random access approaches. The proposed policy can also be applied for the selection of additional transmission parameters simultaneously and in different scenarios to further improve the network performance. An interesting future research direction may be to apply RL algorithms, more precisely MAB algorithms, to handle various other wireless communication challenges.

Author Contributions: Both authors have contributed equally in the research work and preparation of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not Applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. LoRaWAN for Developers. Available online: https://lora-alliance.org/resource_hub (accessed on 1 February 2023).
2. Kerkouche, R.; Alami, R.; Féraud, R.; Varsier, N.; Maillé, P. Node-Based Optimization of LoRa Transmissions with Multi-Armed Bandit Algorithms. In Proceedings of the 2018 25th International Conference on Telecommunications (ICT), Saint-Malo, France, 26–28 June 2018; pp. 521–526.
3. Gupta, S.; Chaudhari, B.S.; Chakrabarty, B. Vulnerable network analysis using war driving and security intelligence. In Proceedings of the International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–27 August 2016; pp. 1–5. [CrossRef]
4. Azari, A.; Cavdar, C. Self-organized low-power iot networks: A distributed learning approach. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–7.
5. Chaudhari, B.; Borkar, S. Design considerations and network architectures for low-power wide-area networks. In *LPWAN Technologies for IoT and M2M Applications*; Chaudhari, B., Zennaro, M., Eds.; Elsevier: Amsterdam, The Netherlands; Academic Press: Cambridge, MA, USA, 2020; pp. 15–35. [CrossRef]
6. Ochoa, M.N.; Guizar, A.; Maman, M.; Duda, A. Toward a Self-Deployment of LoRa Networks: Link and Topology Adaptation. In Proceedings of the 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Barcelona, Spain, 21–23 October 2019; pp. 1–7.
7. Sendra, S.; Parra, L.; Jimenez, J.M.; Garcia, L.; Lloret, J. LoRa-Based Network for Water Quality Monitoring in Coastal Areas. *Mob. Netw. Appl.* **2022**, 1–17. [CrossRef]
8. Codeluppi, G.; Cilfone, A.; Davoli, L.; Ferrari, G. LoRaFarM: A LoRaWAN-Based Smart Farming Modular IoT Architecture. *Sensors* **2020**, *20*, 2028. [CrossRef]
9. Al Homssi, B.; Dakic, K.; Maselli, S.; Wolf, H.; Kandeepan, S.; Al-Hourani, A. IoT Network Design Using Open-Source LoRa Coverage Emulator. *IEEE Access* **2021**, *9*, 53636–53646. [CrossRef]
10. Bor, M.; Roedig, U. LoRa Transmission Parameter Selection. In Proceedings of the 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS), Ottawa, ON, Canada, 5–7 June 2017; pp. 27–34.
11. Zhou, Q.; Xing, J.; Hou, L.; Xu, R.; Zheng, K. A Novel Rate and Channel Control Scheme Based on Data Extraction Rate for LoRa Networks. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019; pp. 1–6. [CrossRef]
12. Reynders, B.; Meert, W.; Pollin, S. Power and Spreading Factor Control in Low Power Wide Area Networks. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6.
13. Cuomo, F.; Campo, M.; Caponi, A.; Bianchi, G.; Rossini, G.; Pisani, P. EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations. In Proceedings of the IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Rome, Italy, 9–11 October 2017; pp. 1–8. [CrossRef]

14. Ullah, M.A.; Iqbal, J.; Hoeller, A.; Souza, R.D.; Alves, H. K-Means Spreading Factor Allocation for Large-Scale LoRa Networks. *Sensors* **2019**, *19*, 4723. [CrossRef] [PubMed]
15. Ghorpade, S.N.; Zennaro, M.; Chaudhari, B.S.; Saeed, R.A.; Alhumyani, H.; Abdel-Khalek, S. A Novel Enhanced Quantum PSO for Optimal Network Configuration in Heterogeneous Industrial IoT. *IEEE Access* **2021**, *9*, 134022–134036. [CrossRef]
16. Ghorpade, S.; Zennaro, M.; Chaudhari, B.S. Towards Green Computing: Intelligent Bio-Inspired Agent for IoT-Enabled Wireless Sensor Networks. *IJSNET* **2021**, *35*, 121. [CrossRef]
17. Moy, C. IoTligent: First World-Wide Implementation of Decentralized Spectrum Learning for IoT Wireless Networks. In Proceedings of the 2019 URSI Asia-Pacific Radio Science Conference (AP-RASC), New Delhi, India, 9–15 March 2019; pp. 1–4.
18. Besson, L.; Bonnefoi, R.; Moy, C. GNU Radio Implementation of MALIN: Multi-Armed bandits learning for Internet-of-Things Networks. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019.
19. Bonnefoi, R.; Besson, L.; Moy, C.; Kaufmann, E.; Palicot, J. Multi-Armed Bandit Learning in IoT Networks: Learning Helps Even in Non-Stationary Settings. In Proceedings of the Cognitive Radio Oriented Wireless Networks, Lisbon, Portugal, 20–21 September 2017; Marques, P., Radwan, A., Mumtaz, S., Noguet, D., Rodriguez, J., Gundlach, M., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 173–185.
20. Ta, D.-T.; Khawam, K.; Lahoud, S.; Adjih, C.; Martin, S. LoRa-MAB: A Flexible Simulator for Decentralized Learning Resource Allocation in IoT Networks. In Proceedings of the 2019 12th IFIP Wireless and Mobile Networking Conference (WMNC), Paris, France, 11–13 September 2019; pp. 55–62.
21. Chaudhari, B.S.; Zennaro, M. Introduction to Low-Power Wide-Area Networks. In *LPWAN Technologies for IoT and M2M Applications*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 1–13, ISBN 9780128188804.
22. Bonnefoi, R.; Besson, L.; Manco-Vasquez, J.; Moy, C. Upper-Confidence Bound for Channel Selection in LPWA Networks with Retransmissions. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW), Marrakech, Morocco, 15–18 April 2019. [CrossRef]
23. Park, G.; Lee, W.; Joe, I. Network resource optimization with reinforcement learning for low power wide area networks. *J. Wirel. Commun. Netw.* **2020**, *2020*, 176. [CrossRef]
24. Ning, W.; Huang, X.; Yang, K.; Wu, F.; Leng, S. Reinforcement Learning Enabled Cooperative Spectrum Sensing in Cognitive Radio Networks. *J. Commun. Netw.* **2020**, *22*, 12–22. [CrossRef]
25. Chen, Y.; Su, S.; Wei, J. A Policy for Optimizing Sub-Band Selection Sequences in Wideband Spectrum Sensing. *Sensors* **2019**, *19*, 4090. [CrossRef] [PubMed]
26. Askhedkar, A.; Chaudhari, B.; Zennaro, M. Hardware and software platforms for low-power wide-area networks. In *LPWAN Technologies for IoT and M2M Applications*; Chaudhari, B., Zennaro, M., Eds.; Elsevier: Amsterdam, The Netherlands; Academic Press: Cambridge, MA, USA, 2020; pp. 397–407. [CrossRef]
27. LoRa Modem Design Guide. Semtech Wireless & Sensing. July 2013. Available online: https://www.openhacks.com/uploadsproductos/loradesignguide_std.pdf (accessed on 15 February 2023).
28. Askhedkar, A.R.; Chaudhari, B.S.; Zennaro, M.; Pietrosemoli, E. TV white spaces for low-power wide-area net-works. In *LPWAN Technologies for IoT and M2M Applications*; Chaudhari, B., Zennaro, M., Eds.; Elsevier: Amsterdam, The Netherlands; Academic Press: Cambridge, MA, USA, 2020; pp. 167–179. [CrossRef]
29. Auer, P.; Cesa-Bianchi, N.; Fischer, P. Finite-time Analysis of the Multiarmed Bandit Problem. *Mach. Learn.* **2002**, *47*, 235–256. [CrossRef]
30. Auer, P.; Cesa-Bianchi, N.; Freund, Y.; Schapire, R.E. The Nonstochastic Multiarmed Bandit Problem. *SIAM J. Comput.* **2002**, *32*, 48–77. [CrossRef]
31. Garivier, A.; Moulines, E. On Upper-Confidence Bound Policies for Switching Bandit Problems. In *Proceedings of the Algorithmic Learning Theory*; Kivinen, J., Szepesvári, C., Ukkonen, E., Zeugmann, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 174–188.
32. Ghorpade, S.N.; Zennaro, M.; Chaudhari, B.S. IoT-Based Hybrid Optimized Fuzzy Threshold ELM Model for Localization of Elderly Persons. *Expert Syst. Appl.* **2021**, *184*, 115500. [CrossRef]
33. Audibert, J.-Y.; Munos, R.; Szepesvári, C. Tuning Bandit Algorithms in Stochastic Environments. In Proceedings of the Algorithmic Learning Theory; Hutter, M., Servedio, R.A., Takimoto, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 150–165.