Optimizing Energy Efficiency of LoRaWAN-based Wireless Underground Sensor Networks

Zhao, Guozheng; Lin, Kaiqiang; Chapman, David; Metje, Nicole; Hao, Tong

DOI:
10.1016/j.iot.2023.100776

Citation for published version (Harvard):
Zhao, G, Lin, K, Chapman, D, Metje, N & Hao, T 2023, 'Optimizing Energy Efficiency of LoRaWAN-based Wireless Underground Sensor Networks: A Multi-Agent Reinforcement Learning Approach', *Internet of Things*, vol. 22, 100776. https://doi.org/10.1016/j.iot.2023.100776

Optimizing Energy Efficiency of LoRaWAN-based Wireless Underground Sensor Networks: A Multi-Agent Reinforcement Learning Approach

ARTICLE INFO

Keywords:
Energy efficiency
multi-agent reinforcement learning
LoRaWAN-based WUSNs
dynamic underground environment
adaptive data rate

ABSTRACT

Extended battery lifetime is always desirable for wireless underground sensor networks (WUSNs). The recent integration of LoRaWAN-grade massive machine-type communications (MTC) technology and WUSNs, namely LoRaWAN-based WUSNs, provides a promisingly energy-efficient solution for underground monitoring. However, due to the limited battery energy, massive data collisions and dynamic underground environment, the energy efficiency of LoRaWAN-based WUSNs still possesses a significant challenge. To propose a liable solution, we advocate reinforcement learning (RL) for managing the transmission configuration of underground sensors. In this paper, we firstly develop the multi-agent RL (MARL) algorithm to improve the network energy efficiency, which considers the link quality, energy consumption and collisions between packets. Secondly, a reward mechanism is proposed, which is used to define independent state and action for every node to improve the adaptability of the proposed algorithm to dynamic underground environment. Furthermore, through the simulations in different underground environment and at different network scales, our results highlight that the proposed MARL algorithm can quickly optimize the network energy efficiency and far exceed the traditional adaptive data rate (ADR) mechanism. Finally, our proposed algorithm is successfully demonstrated to be able to efficiently adapt to the dynamically changing underground environment. This work provides insights into the energy efficiency optimization, and will lay the foundation for future realistic deployments of LoRaWAN-based WUSNs.

1. Introduction

The monitoring and assessment of underground infrastructure and environment by wireless sensor networks (WUSNs) have recently become a research hot spot due to the accelerating development and use of underground space [1–3]. Considering the practical requirement of long-range and long-duration communications, as well as the energy consumption of battery-powered nodes deployed underground [4, 5], the Low-Power Wide-Area Networks (LPWANs) communication technologies have been recently developed and applied to WUSNs for sustainable underground monitoring [6–8].

Among various LPWANs technologies such as the Long Range (LoRa), the Sigfox, and the Narrowband-IoT (NB-IoT), LoRa demonstrates the advantages in better security, higher maximum packet length and the support for deployment of a private Long-Range Wide Area Network (LoRaWAN), all of which making it a promising candidate for pairing with WUSNs for large-scale underground monitoring [6]. There have therefore been some works carried out on the LoRaWAN-based WUSNs [4, 9].

It is essential to reduce the energy consumption of the network and thus increase its lifetime, especially for LoRaWAN-based WUSNs where sensors are buried underground and batteries are difficult to replace or recharge. Besides the recent investigation of backscatter-assisted wireless-powered underground sensor networks approach to increase the energy efficiency and throughput [10], for the classical LoRaWAN, researchers have made dedicated efforts to improve the network lifetime by adjusting the physical layer parameters or updating the strategy for transmission configurations. For example, researchers have developed the adaptive data rate (ADR) mechanism to optimize data rates and to further reduce the energy consumption by adjusting the spreading factor (SF) and transmit power (TP) [11, 12]. More recently, the authors in [13] built on the ADR to further optimize the trade-off between the packet delivery ratio and energy consumption. Furthermore, given the adjusted parameters, some works adopted machine learning to update the transmission configurations for each underground sensor [14, 15]. However, these mechanisms cannot adapt to the dynamically changing underground environment and therefore cannot be directly applied to LoRaWAN-based WUSNs.

ORCID(s):

In recent years, the reinforcement learning (RL) approach has become increasingly popular for systems with complex and dynamic problem spaces. Different from the other machine learning methods such as supervised learning and unsupervised learning, the RL approach uses trial-and-error search method to discover the network environment and to learn the resource management strategies without labelling the dataset at each time step. That can produce the most effective action policy which adapts to environmental changes over time, which makes RL as a powerful tool to adapt to dynamically changing underground environment. Hence, we propose to adopt RL for the energy efficiency optimization in LoRaWAN-based WUSNs in this study.

RL [16], one of the machine learning paradigms, recently becomes popular as the employed intelligent agent is able to achieve a goal under uncertain and complex environment, through maximizing the notion of cumulative rewards. It therefore can assist to adapt to the changing underground environment that traditional ADR normally fails. In this paper, we set goals to leverage RL to achieve the awareness of the dynamic underground environment and to derive the best transmission parameters to optimize the energy efficiency. There are two classic models in RL, the single-agent RL (SARL) and the multi-agent RL (MARL). Although SARL can observe the global environmental states, each agent learns independently, often resulting in unsatisfactory results in dynamic environment [17]. Furthermore, it is unsuitable for the massive networks that would cause unrealistically large action space. In contrast, MARL allows each agent to sense the local environment and to learn its own policy with a fixed action space, which would facilitate the efficient search for the best actions, as well as effectively reduce collisions between packets in the large-scale LoRaWAN-based WUSNs.

In this paper, we present a MARL-based algorithm that can effectively optimize the transmission parameters to improve the energy efficiency of large-scale LoRaWAN-based WUSNs, with the adaptability to the dynamic underground environment. In this work, we focus on the energy efficiency of nodes buried underground and assume that the gateway has stable energy supplies such as a grid system. Notice that we use the uplink transmission to upload the sensor data and the downlink transmission to assign optimized uplink transmission parameters, i.e., SF and TP. Furthermore, we use the maximum transmit power to allocate the optimized parameters to underground sensors to ensure successful delivery in aboveground-to-underground (AG2UG) as the refractive loss in AG2UG is higher than that in underground-to-aboveground (UG2AG). Therefore, this work focuses on optimizing the transmission parameters of nodes in the uplink (UG2AG). In order to implement the optimization of transmission configurations, we firstly establish the network architecture and system model. Aided by MARL, we define the reward function to represent the link quality, energy consumption and collision, and then match each node with the appropriate agent to learn its own policy and to identify the best configuration for all nodes in a cooperative manner. Compared with the standard ADR algorithm, we analyze and discuss the improvement in energy efficiency of LoRaWAN-based WUSNs for different network capacities, scalability and underground conditions. We also verify the adaptability of the algorithm to the dynamically changing underground environment based on the daily variation of volumetric water content (VWC) in the field. The main contributions of this paper can be summarized as follows:

1) We construct the network architecture of LoRaWAN-based WUSNs and utilize the MARL optimization algorithm to improve the energy efficiency. We also develop and successfully execute the evaluation algorithm for two key metrics, i.e., energy per packet (EPP) and data extraction rate (DER).

2) Considering the dynamic underground environment, we adopt the method called ‘central evaluation, marginal decision-making’ to optimize the network energy efficiency. In this method, each agent has a low-dimension, fixed-size state space and action space, which relaxes the training of RL algorithms and maintains the rapid rate of convergence. Furthermore, we consider the variability of each agent’s contribution to the overall performance by dissimilarly assigning each agent a reward function based on its individual performance to shorten the convergence time of the algorithm.

3) We verify the superiority (e.g., the improvement in energy efficiency is in the order of one thousand) of the proposed algorithm through the comparison with the standard ADR mechanism for different network configurations and underground environment. Its verified adaptability to the dynamically changing underground environment provides useful insights into the practical implementation of large-scale LoRaWAN-based WUSNs.

The remaining part of this paper is organized as follows. Section 2 reviews the related works. In Section 3, we present the system model of the LoRaWAN-based WUSNs. We design the MARL optimization algorithm and evaluation algorithm to improve the energy efficiency of LoRaWAN-based WUSNs in Section 4. In Section 5, we evaluate the proposed optimization algorithm for various network conditions as well as its adaptability to the dynamic environmental changes. Finally, conclusions are drawn in Section 6.

2. Related Work

2.1. LoRaWAN

For any wireless sensor network, it is essential to increase its network capacity and reduce the network energy consumption. To achieve this goal, LoRa officially introduced the ADR mechanism, which is to reduce the data transmission rate by optimally adjusting SF and TP [12]. However, the ADR mechanism neither effectively maximizes the network performance [18, 19], nor avoids packet collisions [20].

There have been various efforts to further improve the performance of LoRaWAN. In terms of collisions, the authors in [18] proposed two SF allocation schemes by combining the SF orthogonality and radio range visibility, to reduce the collision occurrence with consequently increased DER and throughput. Simulation results showed that both methods consistently guarantee high bit rates under high traffic loads. In [21], they further introduced the EXPLoRa-KM and EXPLoRa-TS to improve the network performance in regions with a high probability of collisions. The author of [20] proposed CA-ADR, taking into account the collision probability at the MAC layer to assign data rates to nodes, which outperforms the standard ADR that only considers the link-level performance. In addition to improving DER and network throughput, reducing the network energy consumption (NEC) is another essential objective. In [22], the authors studied the data rate fairness and adjusted different TP to maintain the nodes' lifetime. The optimization of energy efficiency was illustrated in [19], where the authors innovatively introduced a low-complexity user scheduling scheme combined with SF and TP to achieve near-optimal energy efficiency. In [13], the authors focused on the impact of the coding rate (CR), and proposed an enhanced greedy ADR mechanism with CR adaptation to the combined SF, TP, and successfully demonstrated the optimized trade-off between DER and EPP. Considering the ADR mechanism is not suitable for a dynamically changing network [23, 24], the authors in [25] proposed the enhanced ADR to improve the quality of service (QoS) of the LoRaWAN with mobility. More recently, the authors in [26] provided an energy efficiency model based on the symbol error model to adjust SF and TP, thus improving the energy efficiency of LoRaWAN for a dynamic environment.

Although the collision, data rate allocation, energy efficiency have been intensively studied by various researchers, the adaptation of LoRaWAN to the tempo-spatially dynamic underground environment has not yet been explored. The temporally changing VWC of soils and spatially varying path loss between nodes and gateways impedes the improvement of the network performance of LoRaWAN-based WUSNs. However, the intrinsic advantage of RL is that it can utilize the agent-reward scheme to efficiently learn and adapt to the complex environment with optimal parameters. It, therefore, inspires us to adopt RL to adjust the physical layer parameters thus improving the network energy efficiency in the LoRaWAN-based WUSNs.

2.2. Reinforcement Learning in LoRaWAN

Reinforcement learning has brought to the attention in the field of wireless communications for its exploration of optimal solutions to decision problems. The authors in [27] developed a flexible LoRaWAN simulator where the resource was flexibly allocated by treating the resource allocation problem as the multi-armed bandit problem. However, it cannot be applied to the network where nodes have high mobility. To deal with this issue, the authors in [28] proposed LoRaDRL to configure SF and TP of nodes at run using deep RL (DRL). Although this work reduces collisions and ensures better network performance, it may be unsuitable for the massive machine-type communications (MTC) network due to the large amount of action spaces associated with signal agents. Furthermore, its convergence time will inevitably be prolonged extensively in a massive MTC network under the dynamic underground environment. In [29], the authors applied the Q-learning-based MARL to manage the LoRaWAN. Simulation results successfully demonstrated the improvement in the data transmission reliability and network power consumption. However, this study mainly focused on the reduction in power consumption without improving the network energy efficiency. Moreover, Q-learning, an off-policy RL method, uses the greed policy for decision making and, in general, limits the scale of state and action spaces [30]. The authors in [31] used the MARL to automatically allocate network resources in LoRaWAN. In this method, the Deep Q-Network (DQN) adopted by each agent uses the states of all nodes as an input layer parameter to process the training. However, when the number of nodes is large, such as thousands of nodes for massive MTC scenarios, the ultra-high dimensional input layers make the training of RL algorithms impractical if not totally infeasible, by significantly slowing down the rate of convergence. Besides, when the number of nodes changes in practice, the size of the input layer of DQN will also need to change, which means that the structure of DQN must be adaptable to external changes, raising the difficulty in network flexibility. To partically addess this issue, in [32], the authors adopted the network slicing technique to propose a deep deterministic policy gradient based slice optimization algorithm to improve the performance of LoRaWAN. This algorithm divided the physical resources of the gateway into multiple virtual networks to provide different QoS guarantee. The authors in [33] provided an approach to the network management under multiple co-existing applications using MARL. Their results revealed that MARL can significantly improve ADR and the responsiveness compared with the single-agent model. However, this approach did not put focus on the total network energy efficiency, which is a key metric for massive MTC networks. Furthermore, RL operations are all performed at the node level, which inevitably increases the overall energy consumption of the nodes. This, therefore, is not suitable for WUSNs, where the energy conservation is vital due to the difficulties of battery replacement and recharging.

Different from the these existing works, we mainly focus on the large-scale LoRaWAN-based WUSNs in the dynamic underground environment, such as a massive MTC network with thousands of nodes. We propose to implement MARL to adaptively balance the trade-off between DER and EPP and thereby to improve the network energy efficiency in the dynamically changing underground environment. Specifically, we design the architecture of the MARL algorithm based on the characteristics of WUSNs, so that it could quickly adapt to the changes in both the network configurations and the underground environmental conditions. We also take into account the collisions between packets and embody it in the reward function.

In our algorithm, we draw on the central idea of *Counterfactual Multi-Agent Policy Gradients* [34] to make judgement and decisions. In this idea, each agent can make decisions based on their current observations and experienced history, and obtain the 'critics' on the decision at the concentration, which is called 'central evaluation, marginal decision-making'. In essence, we acquire information on the actions and observations taken by all agents and determine the overall reward for a specific action under a specific agent. In other words, our algorithm doesn't ignore the inter-agent dependencies and is significantly different from the SARL and traditional MARL methods, which are inadequate for collaborative tasks. Furthermore, this method allows each agent to have a partial but distinct view of the environment, which means a low-dimension, fixed-size and independent state space for every agent. As a result, the low dimensional input layers can relax the training of RL algorithms and maintain the rapid rate of convergence. Besides, the independent environment for each agent benefits the adaptability to the change of the number of nodes. Notably, in order to reduce the energy consumption of each node, we set the corresponding agent for each node at a gateway and only allow the RL operations to be implemented at the gateway.

3. System Design

3.1. Network Architecture

A typical LoRaWAN-based WUSN consists of sensor nodes, gateways, network servers and application servers (Fig. 1). Sensor nodes are deployed underground to gather various environmental information (PH, moisture, and temperature, etc.), and a gateway is placed above the ground surface to receive the data packets from sensor nodes. The transmission of such data is subject to the path loss and collisions with other packets. The network servers receive the data packets from the gateway, and are used to manage the whole network, including the acknowledgement of packets and packet routing, etc. The application servers take the responsibility for processing the application-specific data packets received from sensor nodes [32].

3.2. System Model

3.2.1. Channel Model

To describe the electromagnetic waves propagation between air and soil, the proper channel models serve as the basis of WUSNs. In the paper, we use the channel models in [4] and [35] for the UG2AG and AG2UG communications, as illustrated in Fig. 2. The burial depth of a sensor node is h_u, the height of a gateway deployed above the ground surface is h_a, $d_{ug}$is the distance of the underground communication, d_{ag} is the distance of the aboveground communication, $d_{internode}$ is the horizontal distance between a gateway and a sensor node, and $d_{surface}$ is the distance accounting for the soil-air interface propagation.

The received signal strength, P_r, at the receiver is given by

$$P_r = P_t + G_t + G_r - \begin{bmatrix} L_{ug}\left(d_{ug}\right) + L_R + aL_{ag}\left(d_{ag}\right) \\ +bL_{surface}\left(d_{surface}\right) - 10\log\chi^2 \end{bmatrix}, \tag{1}$$

where P_t is the transmit power, G_t is antenna gain of the transmitter, G_r is the receiver antenna gain, $L_{ug}\left(d_{ug}\right)$ and $L_{ag}\left(d_{ag}\right)$ are the path losses in soils and air, respectively. $L_{surface}\left(d_{surface}\right)$ is the attenuation caused by lateral

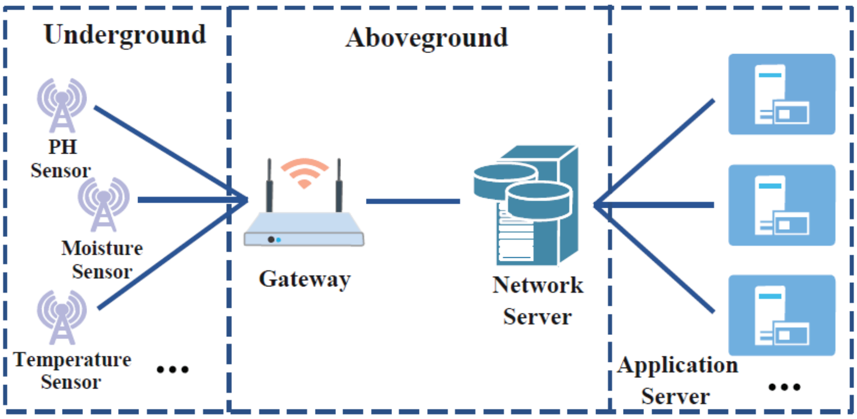

Figure 1: System architecture of LoRaWAN-based WUSNs.

waves, L_R is the refraction loss, which is classified as the refraction loss for UG2AG/AG2UG communications [4], $-10\log\chi^2$ is the attenuation caused by the multi-path fading. a and b in Eq. (1) are the coefficients of L_{ag} and $L_{surface}$, respectively. In this study, due to the gateway is placed 3 m above the ground surface, the attenuation of lateral waves can be ignored [36], hence a equals 1 and b equals 0. χ is a random variable of the Rayleigh distribution, and the probability density function of χ is:

$$f(\chi) = \frac{\chi}{\sigma_R^2} e^{-\chi^2/2\sigma_R^2}, \tag{2}$$

where σ_R is Rayleigh distribution parameter.

In Eq. (1), $L_{ug}\left(d_{ug}\right)$ and $L_{ag}\left(d_{ag}\right)$ can be calculated by

$$L_{ug}\left(d_{ug}\right) = 6.4 + 20\log d_{ug} + 20\log\beta + 8.69\alpha d_{ug}, \tag{3}$$

$$L_{ag}\left(d_{ag}\right) = -147.6 + 20\log d_{ag} + 20\log f, \tag{4}$$

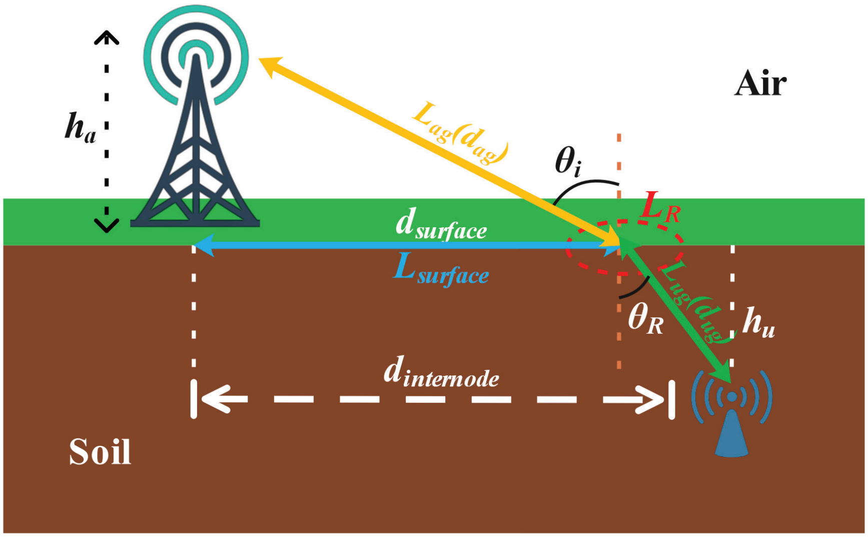

Figure 2: The channel model of UG2AG and AG2UG communications.

Table 1
Sensitivity of LoRa (in dB)

BW\SF	7	8	9	10	11	12
125 kHz	-126.50	-127.25	-131.25	-132.75	-134.50	-137.25
250 kHz	-124.25	-126.75	-128.25	-130.25	-132.75	-134.00
500 kHz	-120.75	-124.00	-127.50	-128.75	-128.75	-132.25

where f is the operation wave frequency, α and β are the attenuation constant and the phase shifting constant, respectively, that are given as

$$\alpha = 2\pi f \sqrt{\left(\mu_r \mu_0 \epsilon' \epsilon_0 / 2\right) \left[\sqrt{1 + (\epsilon''/\epsilon')^2} - 1\right]}, \tag{5}$$

$$\beta = 2\pi f \sqrt{\left(\mu_r \mu_0 \epsilon' \epsilon_0 / 2\right) \left[\sqrt{1 + (\epsilon''/\epsilon')^2} + 1\right]}, \tag{6}$$

where ϵ' and ϵ'' are the real part and imaginary part of the soil's dielectric constant, respectively.

In the UG2AG communication, the refraction losses, L_R in (1), can be given by

$$L_R = L_{ug-ag} \simeq 10 \log \left[\left(\sqrt{\epsilon'} + 1 \right)^2 / 4\sqrt{\epsilon'} \right], \tag{7}$$

Correspondingly, L_R for the AG2UG communication is written as

$$L_R = L_{ag-ug} \simeq 10 \log \frac{\left(\cos\theta_i + \sqrt{\epsilon' - \sin^2\theta_i} \right)^2}{4\cos\theta_i \sqrt{\epsilon' - \sin^2\theta_i}}, \tag{8}$$

where θ_i is the incident angle [32].

3.2.2. Adjustment Parameters

There are four physical layer parameters, i.e., SF, TP, code rate (CR) and bandwidth (BW) in LoRaWAN. Considering that link quality, data collision and network energy consumption are all related to SF and TP, we use these two parameters as the components of the active space in our optimization algorithm. Their brief description is given below.

Spreading Factor (SF) SF is the ratio between the symbol rate and chip rate, which indicates the number of symbols sent per packet [37]. Each symbol exists 2^{SF}chips. For LoRa, there are six SFs ranging from 7 to 12. A higher SF implies a stronger demodulated signal-to-noise ratio and thus a longer propagation distance. However, this leads to a lower transmission rate, prolonged time-on-air and ultimately more energy consumption. In addition, a higher SF provides a higher receiver sensitivity. Combined with BW, the receiving sensitivities of LoRa with different SF and BW are shown in Table 1. The orthogonality between different SF can reduce the collisions between the packets equipped with different SF, thus increasing the probability of successful packet transmission.

Transmission Power (TP) According to Eq. (1), TP directly affects the success of packets reaching a gateway since P_r must be greater than the receiver's sensitivity for the packets received by a gateway. TP can be adjusted from -4 dBm to 20 dBm in 1 dB steps; however, the range is usually limited to 2-20 dBm due to the hardware implementation limitations [38]. Although higher TP improves the received signal quality and expands the communication range, it also increases the energy consumption at sensor nodes. Furthermore, it can prevent other nodes from successfully delivering packets using the same transmission parameters. That leads to the competition between nodes and thus results in inefficient use of network capacity [39]. When two packets reach the gateway at the same time, the target packet can be successfully demodulated only if the difference between their received power is greater than a certain threshold (e.g., 6 dB) [38].

3.2.3. Evaluation Metrics

The link quality and energy consumption of data transmission are vital for WUSNs, because they decide the transmission quality and life of the network. Hence, in this study we utilize DER and EPP as the evaluation metrics, that need to be optimally balanced.

DER can be calculated by

$$DER = N_{received} / N_{transmitted}, \tag{9}$$

where $N_{received}$ is the number of packets received by a gateway. $N_{transmitted}$ is the number of packets transmitted by the sensor node.

The network energy consumption (NEC) represents the energy consumed during the packet transmission phase, which does not include the energy received and sent by nodes and gateway. It can be calculated as

$$NEC = N_{transmitted} \times \left(E_{transmission} + E_{algorithm}\right), \tag{10}$$

where $E_{transmission}$ is the energy consumption of transmissions, $E_{algorithm}$ is the energy consumption of the algorithm itself, which can be given by

$$E_{algorithm} = V_{supply} \times I_{work} \times T_{run}, \tag{11}$$

$$E_{transmission} = V_{supply} \times I_{transmitted} \times T_{transmitted}, \tag{12}$$

where V_{supply} is the supply voltage and it is set at 3V in this study, I_{work} is the working current, T_{run} is the program runtime, $I_{transmitted}$ is the transmission current and $T_{transmitted}$ is the duration of the transmission:

$$T_{transmitted} = T_{preamble} + T_{payload}, \tag{13}$$

The preamble duration $T_{preamble}$ is given by

$$T_{preamble} = \left(n_{preamble} + 4.25\right) \times 2^{SF} / BW, \tag{14}$$

where $T_{preamble}$ is the number of preamble symbols, and $T_{payload}$ is the payload duration, which can be calculated as

$$T_{payload} = \left(8 + \max\left(\begin{array}{c} \mathrm{ceil}\left(\frac{8PL-4SF+28+16-20H}{4(SF-2DE)}\right) \\ \times(CR+4) \end{array}\right)\right) \times 2^{SF} / BW, \tag{15}$$

where H is the identifier to illustrate if the implicit header is disabled, which usually sets to 0, and only set to 1 when SF=6 is employed. DE is the data rate optimization option that is set to 1 when SF equals 11 or 12, and 0 when the rest of SF are employed.

Therefore, the EPP can be directly calculated by

$$EPP = NEC / DER. \tag{16}$$

4. MARL-Based Optimization Algorithm

4.1. MARL Model for LoRaWAN-based WUSNs

To improve the transmission quality and prolong the network life of large-scale LoRaWAN-based WUSNs in the dynamic underground environment, we use MARL in this study. Traditional SARL struggles to cope large networks due to the massive action spaces. For example, if the number of nodes is 10000, the action space need to be

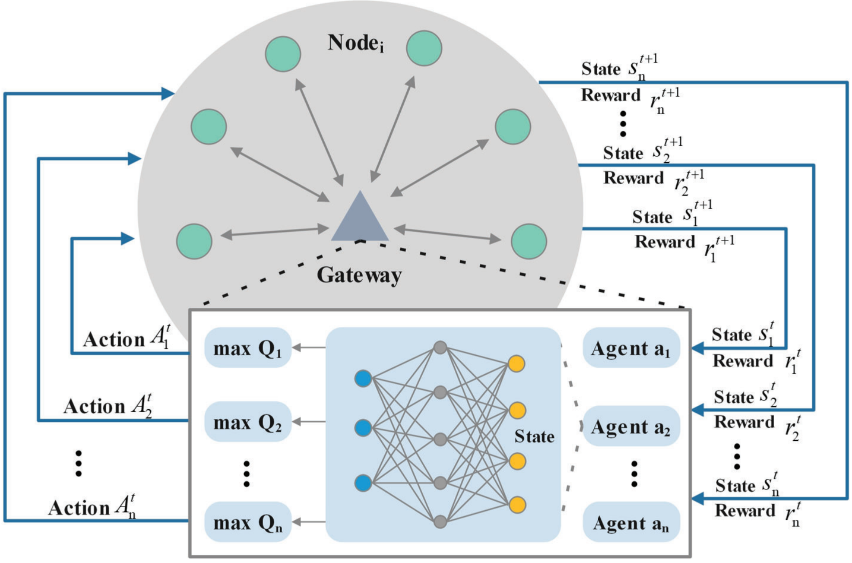

Figure 3: Optimization architecture of the proposed MARL algorithm.

$(N_{SF}\times N_{TP})^{10000}$, where N_{SF} and N_{TP} are the number of operational SF and TP, which can be 6 and 19, respectively. The resulting action space has a size too big to be practically solvable. In addition, the SARL takes a lot of time to complete the network optimization [28], which is inefficient to adapt to a dynamic underground environment. On the contrary, MARL allows each node to have a corresponding agent that learns its own policy and has an action space with a fixed size, which facilitates a significant reduction in the action space and consequently can efficiently find the best action.

It consumes energy to run optimization algorithms. Since the total energy of a buried node is limited, we avoid the optimization at the node, and instead conduct it at the gateway in this study. Furthermore, considering the massive action spaces used in the centralized learning paradigm, we adopt 'centralized training with decentralized execution (CTDE)'-type MARL to optimize the network energy efficiency. With CTDE, the gateway has a centralized location to receive the states of all nodes and to allocate the computing space to each agent to run DQN in parallel for training. After the training, the selected action will be transmitted to the corresponding node and be executed by the node. Because the action is executed at the node, $R1$ and $R2$ cannot be immediately acquired by the gateway. Hence, only when the node conducts the action selected by the gateway and initiates the next transmission does the gateway calculate the rewards ($R1$ and $R2$) of this action by Eqs. (20-21).

In Fig. 3, we present the architecture of the proposed system consisting of multiple LoRa nodes buried underground and one gateway placed 3 m above the ground surface. The gateway sets multiple agents (i.e., a_1, a_2,…, a_n) to the corresponding nodes (i.e., node$_1$, node$_2$,…, node$_n$) to select the optimal actions (SF and TP) and then transmit them to the relevant nodes. As we know that the Markov decision process (MDP) model is usually used on the assumption that the whole environment can be recognized. However, in this study when the environment can only partially presented by the multi-agent system, we must utilize the Partially Observable Markov Decision Process (POMDP) [40] and implement with MARL.

Specifically, we set their own state of the node as the environmental state for rapid adaptation to any dynamic changes in the underground environment. As shown in Fig. 3, each node$_i$ has a state s_i^t of its own at time t. Moreover, SF and TP are configured by the node when transmitting data at time t, defining as the action A_i^t. After a node takes A_i^t to send the packets, the gateway receives the reward r_i^t which represents the transmission quality and energy consumption of node$_i$, as well as the collision with other nodes when node$_i$ sends data.

Each agent of the network model uses a DQN that combines deep learning with the Q learning algorithm in Deep Mind [41]. In Q-learning, the action with the highest Q value is selected from all the available ones, and the classic formula for the updated Q value is shown as

$$Q(s, A) \leftarrow Q(s, A) + \alpha \left[r + \gamma \max_{A'} Q\left(s', A'\right) - Q(s, A) \right], \tag{17}$$

where α is the learning rate and γ is a discount factor. Furthermore, r represents the reward at time $t + 1$ when an agent transits from the current state s to the next state $s^{'}$.

The DQN consists of two Q-networks which are the current Q-network and the target Q-network. The current Q-network is used to select the action and update the model parameters, while the target Q-network can obtain the target Q value that can be calculated as

$$y_j = \begin{cases} r_j, & \text{if } s^{'} \text{ is terminal } \phi_{j+1}, \\ r_j + \gamma \max_{A'} Q\left(\phi_{j+1}, A'; \theta\right), & \text{otherwise } \phi_{j+1}, \end{cases} \tag{18}$$

where $s^{'}$ is the next state, $A^{'}$ is the selected action, ϕ_{j+1} is the eigenvector of the state $s^{'}$ and θ is the set of parameters of the Q function.

4.2. The Components of Agents

4.2.1. State

A state-space needs to contain enough information to represent the current environment. In the WUSNs, a state-space should hold key information including each node and the sent and received packets, that directly or indirectly affects the network performance. However, it does not mean that there can be as many parameters in the state-space as possible. Every new parameter increases the dimension of the state-space, which influences the optimized performance of the DQN. In Eq. (19), we present the state s_i corresponding to each node$_i$. Furthermore, every agent takes as input values only the parameters contained in the state space s_i of the corresponding node$_i$.

$$\begin{pmatrix} s_1 \\ s_2 \\ \cdot \\ \cdot \\ s_n \end{pmatrix} = \begin{pmatrix} \text{node}_1 & SF_1 & TP_1 & RSSI_1 & Energy_1 \\ \text{node}_2 & SF_2 & TP_2 & RSSI_2 & Energy_2 \\ & & \cdot & & \\ & & \cdot & & \\ & & \cdot & & \\ \text{node}_n & SF_n & TP_n & RSSI_n & Energy_n \end{pmatrix}, \tag{19}$$

where SF_i and TP_i are the SF and TP used by the node$_i$ for each data transmission. $RSSI_i$ is the received signal strength indicator of the signal sent by node$_i$ arriving at the gateway. $Energy_i$ denotes the energy consumed by node$_i$ for the packet transmission.

4.2.2. Action

In this study, we depend on the adjustment of SF and TP to improve the transmission quality and reduce the network energy consumption of LoRaWAN-based WUSNs. Hence, SF and TP are designed as the action of RL. The action space a_i includes 114 fixed actions, which are $SF\{7, 8, 9, 10, 11, 12\} \otimes TP\{2, 3, 4, \ldots, 18, 19, 20\}$. Every agent a_i selects the optimal action A_i from the action space based on DQN to send it to the relevant node.

4.2.3. Reward

The aim of this work is to optimize the energy efficiency of WUSNs in the dynamic underground environment. In the traditional multi-agent system, all agents share a common reward function. It means that the gateway can only calculate the reward function after it has received packets from all nodes, which significantly extends the convergence time of the algorithm. Furthermore, it does not consider the variability of each agent's contribution to the overall performance. Hence, we dissimilarly assign each agent a reward function based on its contribution to global performance.

According to Eq. (16), the network energy efficiency is dependent upon DER and NEC, which is decided by the transmission quality, collision and consumed energy during data transmission per node. Hence, in order to enhance the network energy efficiency, each node needs to firstly ensure that packets are successfully received by the gateway with as little energy consumption as possible. This is represented as $R1$, and can be calculated by Eq. (20), where $R_{pathloss}$ is used to optimize the link quality. If the link quality is good and the packets can reach the node, $R_{pathloss}$ has a positive value and $R1$ is a positive reward. Otherwise, $R_{pathloss}$ has a negative value and $R1$ is a negative reward.

Within Eq. (20), $Energy$ indicates the amount of energy consumed by the node for the packet transmission, and is used to leverage with the link quality. For instance, only the case combining a reasonable link quality with a relatively small $Energy$ leads to a large $R1$ that the MARL algorithm shall optimize for.

$$R1 = \frac{\beta R_{pathloss}}{R_{energy}} = \frac{\beta(RSSI - Sensitivity)}{Energy}, \tag{20}$$

where $RSSI$ is the received signal strength index of packets arriving at the gateway. $Sensitivity$ is the receiving sensitivity of LoRa, which changes with different SF and BW. The specific values of the sensitivities are listed in Table 1. β is an expansion factor, which is used to ensure that the energy consumption can be reduced while achieving high transmission quality.

Note $R1$ is based on the assumption that packets successfully sent by each node to the gateway can only be received if they are collision-free. To account for the collision that inevitably happens in practice, $R2$ is set to represent the collision checking of packets in Eq. (21).

$$R2 = \begin{cases} 1, & \text{if packet is not collided with other packets,} \\ -1, & \text{otherwise.} \end{cases} \tag{21}$$

If the $R_{pathloss}$ has a positive value but the packet is not received by the gateway, it means that the packet has collided with other packets. If the gateway successfully receives the packets, no collision has occurred.

In summary, we need to firstly determine whether a packet is lost ($R1$) and then determine whether a collision has occurred ($R2$). Such logical sequence is used to determine whether the reception is successful. Hence, the total reward function must be multiplication and can be calculated as $R = R1 \times R2$.

4.3. The topology of the agent in MARL

In our algorithm, we use two Q-networks to calculate the Q value and update the parameters of the Q-network. The parameters of the target Q-network do not need to be updated iteratively but are copied from the current Q-network at regular intervals. These two Q-networks have the same topology in which there are three layers, i.e., input, hidden and output layers. Besides, the numbers of nodes in these layers of two Q-networks are the same.

The input layer includes five parameters($node_i$, SF_i, TP_i, $RSSI_i$, $Energy_i$) that belong to the state of each node i corresponding to an agent. The output layer of each agent includes 114 Q values of actions which consist of $SF\{7, 8, 9, 10, 11, 12\} \otimes TP\{2, 3, 4, \ldots, 18, 19, 20\}$. In addition, we use a hidden layer to link the input layer and the output layer. The number of hidden nodes is determined by the empirical formula of hidden nodes [42] displayed in Eq. (22). In this algorithm, the number of hidden nodes is set to 24.

$$\mathrm{H} = \sqrt{M + N} + \alpha \tag{22}$$

where M is the number of nodes in the input layer, N is the number of nodes in the output layer and H is the number of nodes in the hidden layer. α is a constant between 1 and 10.

4.4. Optimization and Evaluation Algorithms

Optimization Algorithm The optimization principle is depicted in Algorithm 1. This algorithm is run throughout the data transmission process where the node periodically sends the packets to a gateway. After node_i transmits the packets to a gateway, the gateway obtains the environment state s_{i}^{t-1} and the initial action A_{i}^{t-1} that node_i adopts. Then the agent corresponding to node_i inputs the state s_{i}^{t-1} and outputs the action A_{i}^{t-1}.

However, the next state s_{i}^{t} and reward r_{i}^{t} are not immediately available, but after node_i has finished transmitting the packets and taking action A_{i}^{t}. Therefore, the node uses these action parameters from the gateway to transmit the next packets. Up until this point, the agent can only calculate the reward r_{i}^{t} dependent upon the new state s_{i}^{t}. At the same time, the tuple (s_{i}^{t-1}, A_{i}^{t-1}, r_{i}^{t}, s_{i}^{t}) is added to the memory.

Evaluation Algorithm To assess the ability of the optimization algorithm in balancing DER and NEC thereby improving the energy efficiency of the LoRaWAN-based WUSNs, we also set EPP as the evaluation index, besides DER. Algorithm 2 illustrates the principle of the evaluation process, in which the EPP of the network is calculated every 60 minutes. In this way, we can verify the adaptability of this algorithm to the dynamically changing underground environment.

Algorithm 1 MARL Optimization Algorithm for LoRaWAN-based WUSNs

```
Initialize the target and online Q-networks of all agents
Initialize the state spaces $s_i^0$, action space $A_i^0$ and $t_i = 0$
**Simulation Start**
Initialize the LoRaWAN
**while** True **do**
    Node$_i$ sent packets $p_i^t$
    Get state of the node at every agent $s_i^t$
    Calculate the reward value $r_i^t$
    $R = \frac{\beta * R_{pathloss_i} * R_{others_i}}{R_{Energy_i}} = \frac{\beta\eta_i(RSSI_i - Sen_i)}{Energy_i}$
    $\eta_i = \begin{cases} 1 & \text{Received} \\ -1 & \text{Not received (packets collision)} \end{cases}$
    **if** $t_i > 0$ **then**
        Collect $(s_{\mathrm{i}}^{t-1}, A_i^{t-1}, r_{\mathrm{i}}^t, s_{\mathrm{i}}^t)$ and add it to the memory
    **end if**
    Feed the state to DQN to get action $A_i^t$
    Take the action $A_i^t$ at the state $s_i^t$ and store in $A_i^{t-1}$
    Send the next packet with action $A_i^t$
    Collect the $s_i^t$ and add it to the $s_i^{t-1}$
    Compute the change in Q value using target Q-network
    Update the online Q-network
    $\phi \leftarrow \phi - \alpha \sum_j \frac{dQ_\phi(s_j,A_j)}{d\phi}\left(Q_\phi\left(s_j, A_j\right) - y_j\right)$
    **if** $steps > target\_replace$ **then**
        Update the target Q-network $\phi'$
    **end if**
    $t_i = t_i + 1$
**end while**
**Simulation end**
```

Algorithm 2 Evaluation Algorithm

```
**Simulation Start**
Initialize the LoRaWAN and $N = 0$
**while** True **do**
    Calculate $DER$ using Eq. (9)
    Calculate $NEC$ using Eq. (10)
    **if** $SimTime > 60\ min \times N$ **then**
        Calculate $EPP$ using Eq. (16)
        The consumed energy of network $Energy = 0$
        The number of packets sent by nodes $N_{transmitted} = 0$
        The number of packets received by a gateway $N_{recived} = 0$
        $N = N + 1$
    **end if**
**end while**
**Simulation end**
```

Complexity Analysis of the Algorithm We mainly analyze the complexity of the optimization algorithm which is based on DQN, by its computational complexity. If there are K agents corresponding to nodes and N times in our simulation, the time complexity of this algorithm is $O(KN)$. Besides, the space complexity of this algorithm is $S(KN)$.

4.5. Simulation Flow of the Optimization Algorithm

Fig. 4 displays the simulation process of the optimization. Before the transmission is initiated, the network parameters are set and nodes are deployed randomly. At first, all nodes set with the same parameters to transmit the packets to the gateway. When all packets arrive at the gateway, the gateway obtains the state of each node$_i$ and allocates the corresponding agent to each node. Then, the agent uses Algorithm 1 to calculate the rewards, and to update Q-value and Q-network, before the optimal action based on the $\epsilon - greedy$ algorithm is determined. Finally, the node$_i$ adopts the new parameters from the gateway to transmit the next batch of packets.

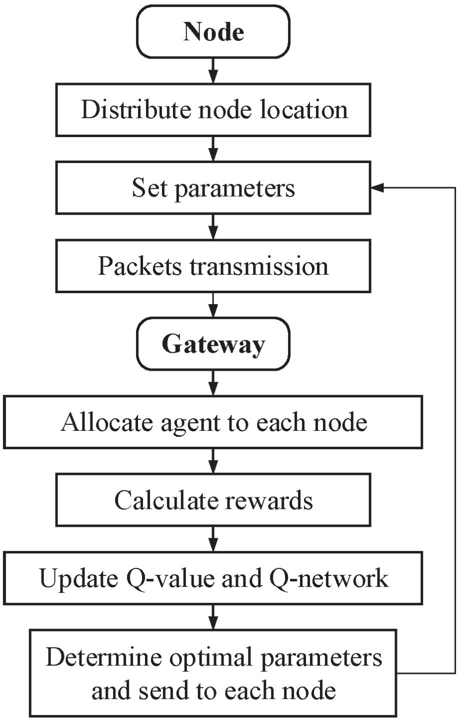
Node
Distribute node location
Set parameters
Packets transmission
Gateway
Allocate agent to each node
Calculate rewards
Update Q-value and Q-network
Determine optimal parameters and send to each node

Figure 4: Simulation flow of the MARL optimization algorithm.

5. Performance Evaluation

In this section, we evaluate the performance of the proposed MARL optimization algorithm. It is based on a simulator which we developed in Python with a SimPy simulation library. We adopt the EPP and DER to verify the optimization performance and the adaptability of this algorithm in different network conditions (e.g., with different numbers of deployed nodes and with different deployment scopes) and underground environments (e.g., various burial depths and VWC). Considering that the standard ADR scheme [12] is the most representative adjustment mechanism provided by LoRa Alliance, and the improvement by related variants (e.g., [13], [18–22], [25], [26]) is much less significant than that of the proposed MARL, we only take the standard ADR algorithm as the baseline.

5.1. Initial Parameters for the Simulation

In the simulations, we set up a large-scale LoRaWAN-WUSN consisting of thousands of sensor nodes buried underground. In this network, by default, 6000 nodes are randomly deployed in a circle with a radius of 1500 m and the gateway is placed at the centre of this circle, 3 m above the ground surface. Furthermore, we set a realistic underground soil composition as listed in Table 2. The default burial depth and VWC are 0.4 m and 20%, respectively. In addition, each node sends a 20-byte packet to the gateway every 15 minutes (4 episodes per hour) and the simulation time is set at 120 hours. The physical layer parameters of LoRa are illustrated in Table 3. Moreover, the centre frequency of each transmission is chosen from the CN470-510 frequency band, and the number of channels available for the uplink and

Table 2
Soil Composition and Soil Properties Used in Simulation

Soil texture	Sand	Clay	Silt	Particle density	Bulk density
loam	$40\ \%$	$20\ \%$	$40\ \%$	$2.66\ g/cm^3$	$1.5\ g/cm^3$

Table 3
Default values of parameters in simulation

Parameters	Value
Number of Nodes	6000
Communication Radius	1500 m
Simulation Time	120 hours
Transmission Rate	20 bytes per 15 minutes
Burial Depth	0.4 m
VWC	20 %
Spreading Factor (SF)	7-12
Transmission Power (TP)	2-20 dBm
Code Rate (CR)	4/5
Bandwidth (BW)	125 kHz

Table 4
Hyperparameters of MARL

Parameters	Values
Activation function	ReLU
Optimizer	Adam
Learning Rate	0.01
Epsilon	0.3
Gamma for Q-Values	0.9
Relay buffer size	24
Minibatch size	6

downlink transmissions is set at 8. If not otherwise specified, the simulation configurations remain the same as those listed in Table 3.

In Table 4, we present the hyperparameters of MARL, which is fine-tuned for the DQN architecture. The activation function is the non-linear function ReLU and the optimizer is Adam. As we know, reply buffer size is generally determined by the size of the sample that occurs in the learning process. In this simulation, the frequency of occurrence of the sample is low due to the large transmission interval. So if the replay buffer is too large, learning will not proceed. Hence, considering the frequency with which nodes send packets, we set a moderately sized relay buffer of 24. Furthermore, the minibatch size is set to a small value to speed up the learning process. To make sure the learning converges quickly, the learning rate is set to 0.01 and the gamma for Q-values is 0.9. For the $\epsilon - greedy$ algorithm, to converge efficiently while maintaining sufficient exploration, we obtain the maximum Q value with a probability of 0.7 and randomly select actions with a probability of 0.3.

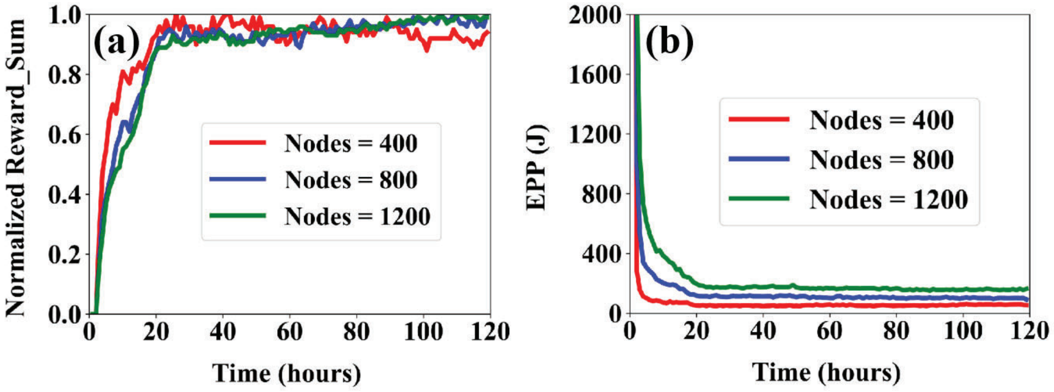

Figure 5: The convergence of the MARL algorithm with different number of nodes for (a) normalized *reward sum*, and (b) EPP.

5.2. Convergence of the Optimization Algorithm

We evaluate the convergence of the proposed MARL algorithm since the underfitting or overfitting of the algorithm can lead to failure in energy efficiency optimization. Furthermore, the convergence speed of the algorithm will have a strong impact on its adaptability to the dynamic environment, that includes not only the changing underground environment but also various network conditions such as capacity and scope. As an example, Fig. 5 displays the convergence of the algorithm with different number of nodes. The normalized sum of rewards every 60 minutes is used to represent the convergence of the algorithm, and a reduced deployment radius (500 m) is set for this preliminary test.

It is observed in Fig. 5(a) that our MARL optimization algorithm can quickly converge to over 80% of the maximum sum of rewards within 18 hours (72 episodes), and then gradually stabilize. As seen in Fig. 5(b), EPP also drops rapidly within 18 hours and then gradually stabilizes. Within these 72 episodes, the network energy efficiency is much greater than that when initial parameters are used. Furthermore, due to the reduction of entanglement among nodes, i.e. independent environmental state and action space for every node, the change in the number of nodes does not have a significant impact on the convergence speed of this algorithm. If we look at these convergence curves more closely, the convergence of the algorithm with more deployed nodes tends to reach the maximum of the normalized reward sum in a more steady manner with much fewer fluctuation, which may be related to the complex balance between the link quality and the packet collision.

5.3. Effect of the Expansion Factor

To illustrate the energy efficiency of the LoRaWAN-based WUSNs, the transmission quality and energy consumption are used as the optimization parameters. However, a good quality link is the prerequisite for reducing the energy consumption. The reward $R1$ proposed in Eq. 20 is used to balance the link quality and the energy consumption of data transmission. Therefore, we use the expansion factor β to enable the link quality, which is essential for the adaption to the challenging underground environment [4].

As displayed in Fig. 6, we investigate the effect of different expansion factor β on EPP under two different underground conditions, i.e., two burial depths representing different link qualities. It is clear that EPP can be optimized to a much lower level for a better link quality at the shallower burial depth of 0.4 m. In this case, the difference by various choices of β can be neglected. As the link quality becomes worse with a larger burial depth, more energy is required to sustain the link quality of transmission. For instance, as the burial depth is increased from 0.4 m to 1.0 m, with the same β (e.g., $\beta = 10^6$), EPP is increased by at least two-folds. In the other case where the link quality if poorer (e.g., burial depth is 1.0 m), the choice of a bigger β can lead to noticeable smaller EPP. It indicates that with the amplified reward $R1$ (Eq. 20), a new balance between the link quality and energy consumption can be reached. This can be particularly useful in optimizing the energy efficiency of WUSNs, where a bigger β is always preferred.

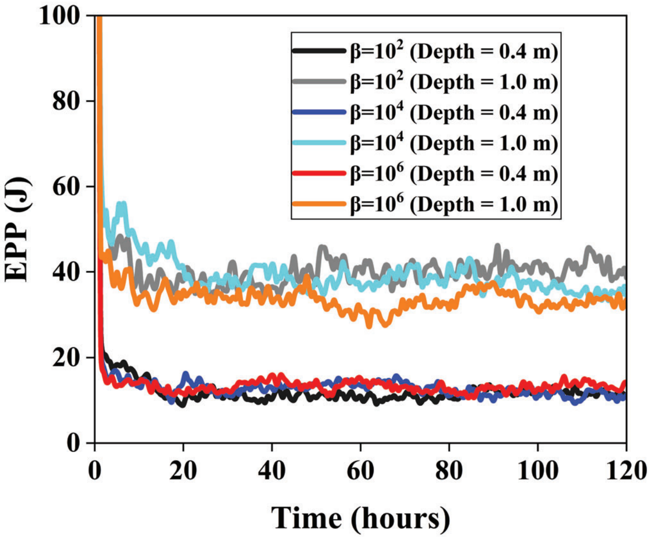

Figure 6: Effect of the expansion factor β on EPP with different burial depth.

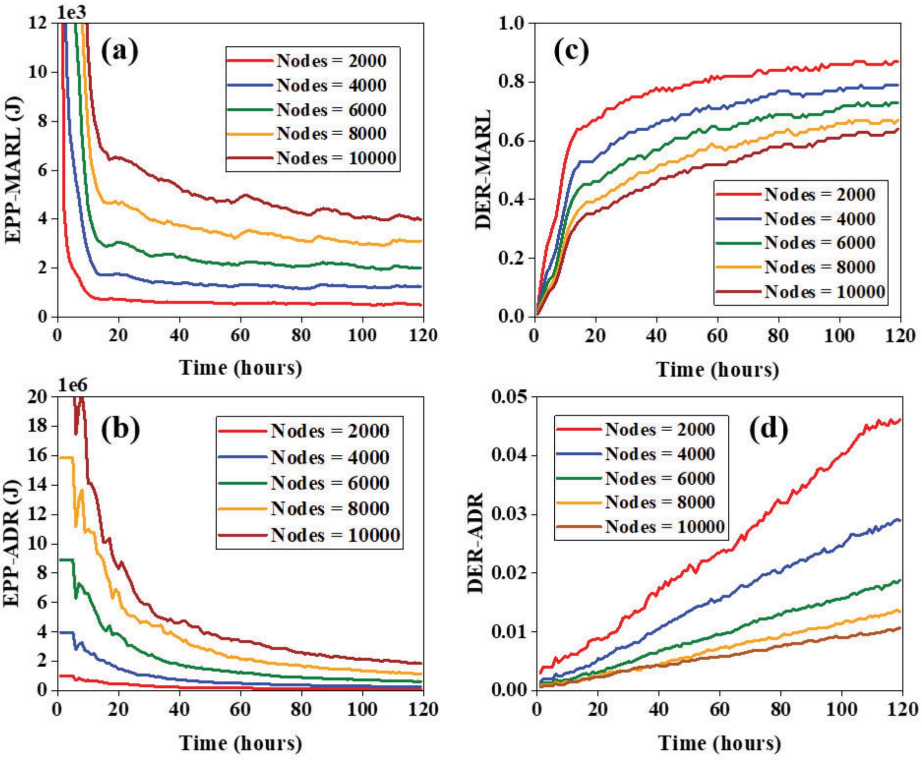

Figure 7: For different network capacities, the optimized EPP by (a) the proposed MARL algorithm, and (b) ADR; the corresponding achievable DER by (c) the proposed MARL algorithm, and (d) ADR.

5.4. The Performance of the Optimization Algorithm in Different Network Capacity and Scope

The scalability including capacity and scope is crucial for most of the networks [43], and it is also one of the key objectives of our proposed MARL algorithm. In this study, we explore the performance of MARL for network optimization in the large scale networks and set 2000 to 10,000 nodes to verify its capability. Fig. 7(a) illustrates EPP

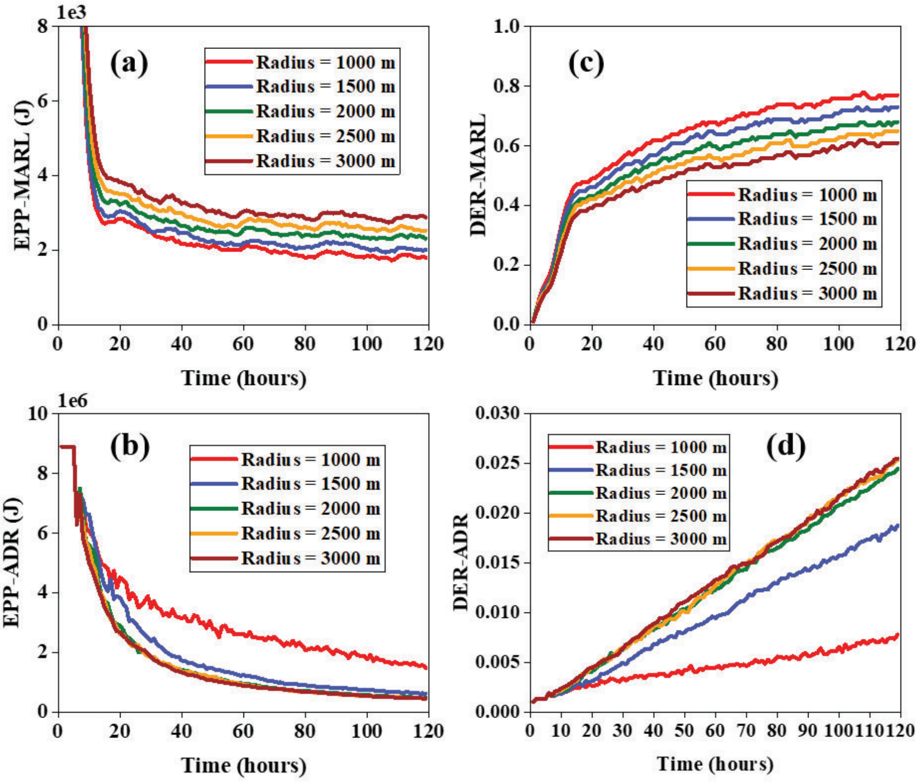

Figure 8: For different network scales, the optimized EPP by (a) the proposed MARL algorithm, and (b) ADR; the corresponding achievable DER by (c) the proposed MARL algorithm, and (d) ADR.

of the network by the proposed MARL algorithm for different numbers of nodes. It can be seen that our algorithm is capable of swiftly optimizing the energy efficiency and reducing EPP to small values. Specifically, if the network capacity is small (e.g., the number of nodes is 2000), the EPP can rapidly (after 56 episodes) converge to around 700 J and then stabilize. Although the convergence time of EPP increases with the number of nodes, the difference between them is small. If the number of nodes is increased to 10000, EPP can quickly drop within 18 hours (72 episodes) and then gradually decreases. This is only 4 hours (16 episodes) after the case of 2000 nodes. It is interesting to find that, the values of EPP optimized by our algorithm are at least a thousand times smaller than those optimized by ADR (Fig. 7(c)) for various network capacities. This is because our algorithm offers much wider choices of optimization parameters than ADR, and more importantly, our algorithm optimizes the collision between packets, which is impossible with ADR. It is also significant that the convergence time of ADR appears to be larger than the proposed algorithm.

As the proposed MARL optimization algorithm is based on a good transmission quality, we also demonstrate DER of the network for different numbers of nodes. With our algorithm, a much higher DER can be reached, compared to the tiny and unrealistic DER achieved by the ADR optimization. For instance, with the MARL algorithm applied to the 2000 deployed underground nodes, DER over 0.6 can be achieved after only 18 hours (72 episodes), and furthermore, over 0.8 after around 80 hours. The unrealistic DER (i.e., $\leq$0.05) by ADR is mainly due to the collisions between packets that cannot be effectively optimized by ADR.

For different network scales, the optimized EPP by the proposed MARL algorithm is displayed in Fig. 8(a), where the convergence behavior is similar to those in Fig. 7(a). Here, the increased EPP for larger scopes is due to that larger SF and TP are used to ensure reasonable link quality as the communication distance between the nodes and the gateway increases. Furthermore, EPP optimized by the proposed algorithm is much less than that optimized by ADR (Fig. 8(c)). The network using the proposed algorithm has a much higher transmission quality (better DER) than the network using ADR.

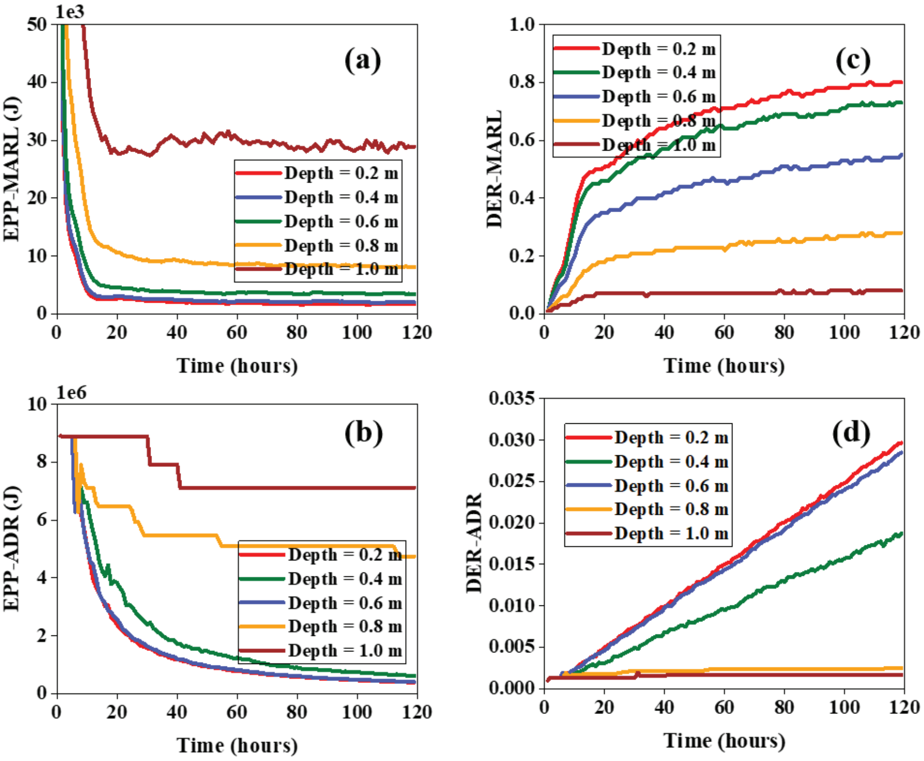

Figure 9: For different burial depths, the optimized EPP by (a) the proposed MARL algorithm, and (b) ADR; the corresponding achievable DER by (c) the proposed MARL algorithm, and (d) ADR.

5.5. The Performance of the Optimization Algorithm in Different Underground Environment

Unlike traditional WSNs, WUSNs are largely affected by dynamic underground environment, such as spatial variation in burial depth, and tempo-spatial variation in VWC. When the underground condition is good, such as VWC = 10% $\sim$ 20%, or the burial depth is 0.2 m $\sim$ 0.4 m, the proposed MARL algorithm allows EPP to swiftly converge and stabilize around a small value, as displayed in Figs. 9(a)-10(a). The corresponding DER is acceptable, i.e., it is above 0.6 after around 60 hours (240 episodes), as in Figs. 9(b)-10(b). By comparison, the optimized EPP and DER by the ADR algorithm is tens to hundreds of times worse (Figs. 9(c, d)-10(c, d)). In general, it is inevitable that both EPP and DER will deteriorate as the underground condition worsens. For example, it is evident in Figs. 9(a, c) that as the burial depth reaches 1.0 m, even after the quick convergence by the proposed algorithm, EPP and DER stabilize around 3×10^4 J and 0.05, respectively, both of which is impractical. It is then advised that our proposed MARL algorithm shall not be implemented for the severe underground conditions such as VWC > 40% or burial depth > 0.8 m.

5.6. Tests for Realistic Underground Environment

The underground environment is not static but dynamic. To verify the optimization performance and the adaptability of the proposed MARL algorithm in a real dynamically changing underground environment, we test it with the actual monitored underground data from a real farm, e.g., the Ward Farm [44]. In general, there are three levels of changes of the underground environment. Firstly, the underground conditions fluctuate within a range of variations, e.g., Day 60 $\sim$ Day 180 in Figs. 11-12. Secondly, the underground environment has general changes over a period of time, e.g., Day 30 $\sim$ Day 50 in Figs. 11-12, where VWC rises from 10% to 40 % within 20 days. The changes in the underground environment for both cases have been shown in our Experiment #1. Thirdly, the underground environment changes sharply and shortly as in Experiment #2, e.g., Day 120 $\sim$ Day 180 for the depth of 0.5 m in Fig. 13.

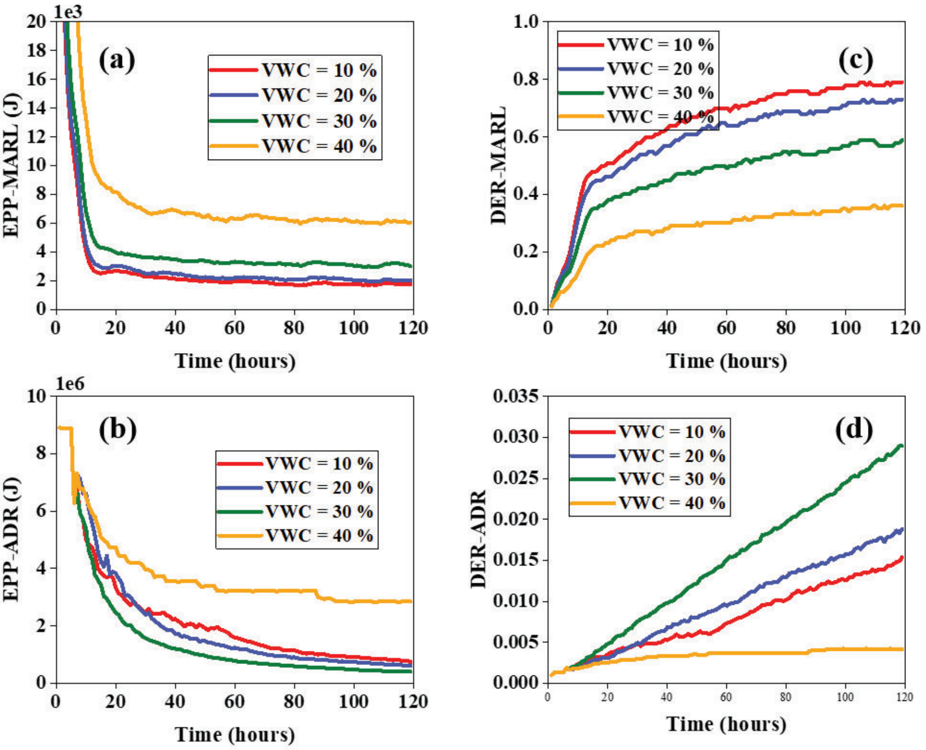

Figure 10: For different VWC, the optimized EPP by (a) the proposed MARL algorithm, and (b) ADR; the corresponding achievable DER by (c) the proposed MARL algorithm, and (d) ADR.

In Experiment #1, the actual monitored underground data comes from the Ward Farm [44], which the *in-situ* VWC data span from 1^{st} April 2011 to 30^{th} September 2011 at the burial depth of 0.2 m. The optimization results are shown in Figs. 11-12, and note that we decrease the transmission rate from 20 bytes per 15 mins to 20 bytes per 30 mins. To demonstrate the dynamically optimized EPP, it is calculated every hour but averaged per day. At the initial state, all nodes are configured with SF = 12 and TP = 20 dBm to ensure a higher quality of transmission. Therefore, the initial EPP of the network is high, reaching 1×10^5 J. As shown in Fig. 11, EPP optimized by the proposed MARL algorithm rapidly reduces to less than 1000 J in the first day and eventually stabilizes to around 850 J. Furthermore, it is interesting to see that after the first day (24 episodes) although VWC is significantly increased to almost 40% and vibrates around 30%, our MARL algorithm can not only sustain a high level of network energy efficiency, i.e., lower EPP, but also efficiently adapt to the change of VWC. For example, the VWC peak at Day 48 leads to a noticeable increase in EPP for the same day, but the algorithm quickly recognizes such change in the reward and adopts a more suitable combination of SF and TP to reduce EPP. On the contrary, due to the restriction within the ADR algorithm [24], it cannot quickly adapt to such changes. It is also evident that the ADR-optimized EPP is tens to hundreds of times higher than that of the MARL-optimized EPP, not to mention its slow convergence speed.

Furthermore, it is observed in Fig. 12 that our proposed MARL algorithm can rapidly upgrade DER to above 0.8 and maintain the strong DER (0.80 - 0.95) in a wide range of VWC. Similar to the poorer performance in EPP, the ADR algorithm cannot improve the transmission quality, i.e., DER, to a acceptable level, which is largely due to its failure in handling packet collisions.

Finally, we evaluate the performance of our MARL algorithm for another site (Sudduth Farm [44]) with VWC at two burial depths in Experiment #2. As displayed in Fig. 13, when the underground environment becomes worse, i.e., the burial depth is increased to 0.5 m, the proposed algorithm can still swiftly converge EPP to reasonable values, though it appears doubled compared to that of a shallower burial depth of 0.2 m. Similar to Fig. 11, the MARL algorithm can efficiently adapt to the temporal variation of VWC, especially evident for the case with the larger burial depth. Considering the balance between the energy requirement and available resources, it reveals that, with the aid

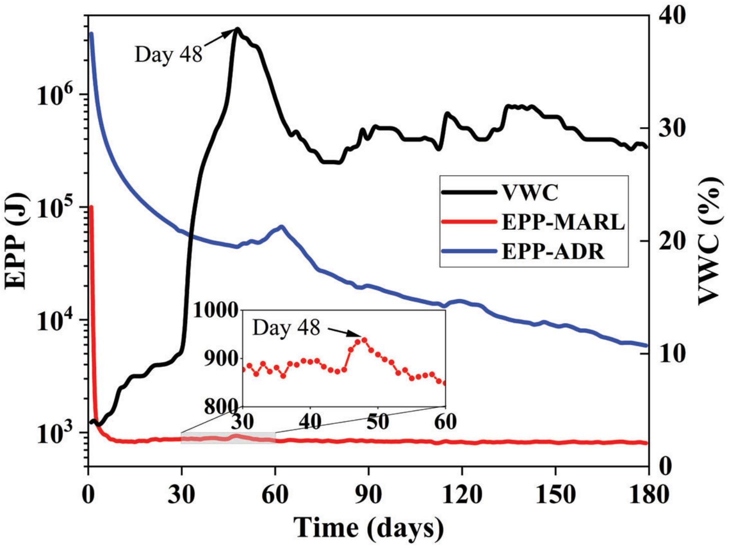

Figure 11: Comparison of EPP between the proposed MARL algorithm and ADR, for a realistic underground environment with daily variation in VWC.

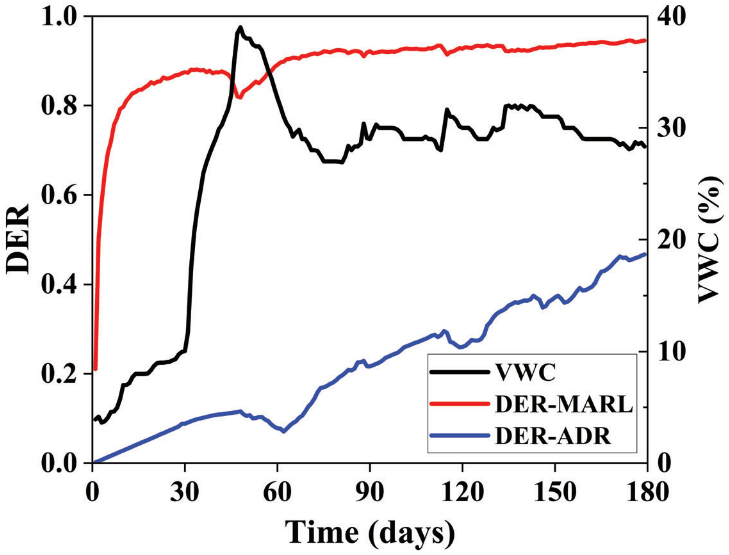

Figure 12: Comparison of the achievable DER between the proposed MARL algorithm and ADR, for a realistic underground environment with daily variation in VWC.

of the proposed MARL algorithm, the practical implementation of WUSNs is indeed feasible with improved energy efficiency.

These two experiments demonstrate that MARL is able to adapt to the dynamics of the subsurface environment and optimize the network energy efficiency. The main underlying reason is that the LoRa's unique physical layer parameters provide MARL a relatively 'stable' learning environment. The physical layer parameters are generally tolerant to the changes of the link quality caused by the varying underground environment. In other words, even if the link quality may change dynamically, as long as the RSSI received by the gateway is greater than the receiving sensitivity, MARL can still select the same action depending on the reward function in this period, and the current physical layer parameters of the node will remain the same. In the meantime, according to the selected action, we can see that the learnt environment is relatively 'stable' for MARL.

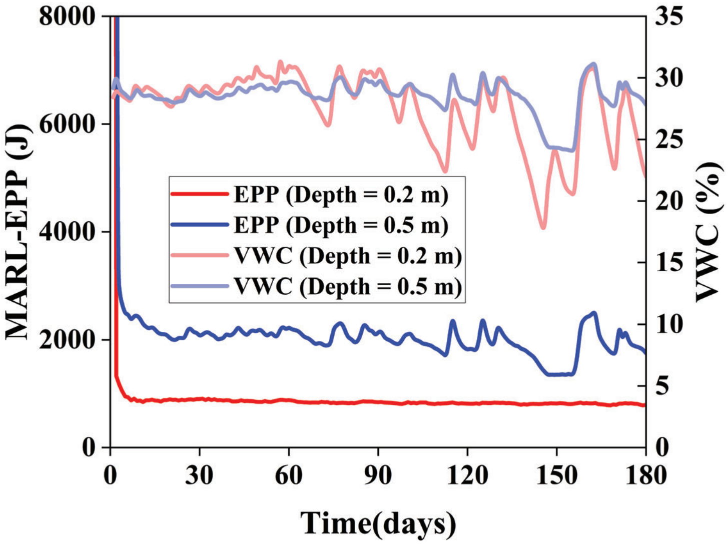

Figure 13: Comparison of EPP by the proposed MARL algorithm between different burial depths, for a realistic underground environment with daily variation in VWC.

6. Conclusion

In this paper, we develop an optimization algorithm based on MARL to improve the energy efficiency of the LoRaWAN-based WUSNs. This algorithm takes into account the link quality and energy consumption, and more importantly, it effectively reduces the collisions of packets at the network level. As the multiple agents are introduced in RL, we take advantage of the reduced action space, and significantly strengthen the adaptive capacity to the dynamic underground environment. A reward mechanism is also developed for the optimal action for the balance between the link quality and energy consumption at the node level as well as the reduction in collisions at the network level. Our comprehensive simulation successfully verify that, this MARL algorithm outperforms the standard ADR algorithm, with hundredth to tenth EPP and realistic DER. It is also demonstrated that the convergence speed of this algorithm is superior in various network capacity and scope, as well as the dynamically changing underground environment. In this study, we mainly focus on improving the energy efficiency of nodes buried underground under the consumption that the gateway has stable energy supplies in this work. However, if the gateway doesn't receive a steady supply of energy or be fed by renewable energy sources for minimizing the grid energy consumption, the power consumption on the gateway and its effect by the number of nodes must be considered. In a nutshell, we believe that this proposed MARL approach will pave the route for designing complex and agile WUSNs. In the future, we will consider the energy efficiency of the gateway, further study how to apply RL for updating the communication parameters in each node by considering the strict LoRaWAN duty cycle, as well as how to upgrade the energy efficiency of WUSNs with extended spatial ranges.

Declaration of competing interest

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

References

[1] I. F. Akyildiz, E. P. Stuntebeck, Wireless underground sensor networks: Research challenges, Ad Hoc Networks 4 (2006) 669–686.

[2] X. Zhang, A. Andreyev, C. Zumpf, M. C. Negri, S. Guha, M. Ghosh, Thoreau: A subterranean wireless sensing network for agriculture and the environment, in: 2017 IEEE conference on computer communications workshops (INFOCOM), IEEE, 2017, pp. 78–84.

[3] B. Silva, R. M. Fisher, A. Kumar, G. P. Hancke, Experimental link quality characterization of wireless sensor networks for underground monitoring, IEEE Transactions on Industrial Informatics 11 (2015) 1099–1110.

[4] K. Lin, T. Hao, Experimental link quality analysis for lora-based wireless underground sensor networks, IEEE Internet of Things Journal 8 (2020) 6565–6577.

[5] F. K. Banaseka, F. Katsriku, J. D. Abdulai, K. S. Adu-Manu, F. N. A. Engmann, Signal propagation models in soil medium for the study of wireless underground sensor networks: a review of current trends, Wireless Communications and Mobile Computing 2021 (2021).

[6] S. Wu, A. Austin, A. Ivoghlian, A. Bisht, K. I.-K. Wang, Long range wide area network for agricultural wireless underground sensor networks, Journal of Ambient Intelligence and Humanized Computing (2020) 1–17.

[7] K. Lin, T. Hao, Z. Yu, W. Zheng, W. He, A preliminary study of ug2ag link quality in lora-based wireless underground sensor networks, in: 2019 IEEE 44th Conference on Local Computer Networks (LCN), IEEE, 2019, pp. 51–59.

[8] L. Moiroux-Arvis, C. Cariou, J.-P. Chanet, Evaluation of lora technology in 433-mhz and 868-mhz for underground to aboveground data transmission, Computers and Electronics in Agriculture 194 (2022) 106770.

[9] C. Ebi, F. Schaltegger, A. Rüst, F. Blumensaat, Synchronous lora mesh network to monitor processes in underground infrastructure, IEEE Access 7 (2019) 57663–57677.

[10] K. Lin, O. L. A. López, H. Alves, D. Chapman, N. Metje, G. Zhao, T. Hao, Throughput optimization in backscatter-assisted wireless-powered underground sensor networks for smart agriculture, Internet of Things 20 (2022) 100637.

[11] R. Kufakunesu, G. P. Hancke, A. M. Abu-Mahfouz, A survey on adaptive data rate optimization in lorawan: Recent solutions and major challenges, Sensors 20 (2020) 5044.

[12] N. Sornin, A. Yegin, Lorawan tm 1.1 specification. lora alliance, 2017.

[13] J. Park, K. Park, H. Bae, C.-K. Kim, Earn: Enhanced adr with coding rate adaptation in lorawan, IEEE Internet of Things Journal 7 (2020) 11873–11883.

[14] R. M. Sandoval, A.-J. Garcia-Sanchez, J. Garcia-Haro, Optimizing and updating lora communication parameters: A machine learning approach, IEEE Transactions on Network and Service Management 16 (2019) 884–895.

[15] R. M. Sandoval, D. Rodenas-Herraiz, A.-J. Garcia-Sanchez, J. Garcia-Haro, Deriving and updating optimal transmission configurations for lora networks, IEEE Access 8 (2020) 38586–38595.

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

[17] Z. Mammeri, Reinforcement learning based routing in networks: Review and classification of approaches, IEEE Access 7 (2019) 55916–55950.

[18] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini, P. Pisani, Explora: Extending the performance of lora by suitable spreading factor allocations, in: 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2017, pp. 1–8.

[19] B. Su, Z. Qin, Q. Ni, Energy efficient uplink transmissions in lora networks, IEEE Transactions on Communications 68 (2020) 4960–4972.

[20] R. Marini, W. Cerroni, C. Buratti, A novel collision-aware adaptive data rate algorithm for lorawan networks, IEEE Internet of Things Journal 8 (2020) 2670–2680.

[21] F. Cuomo, J. C. C. Gámez, A. Maurizio, L. Scipione, M. Campo, A. Caponi, G. Bianchi, G. Rossini, P. Pisani, Towards traffic-oriented spreading factor allocations in lorawan systems, in: 2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), 2018, pp. 1–8. doi:10.23919/MedHocNet.2018.8407091.

[22] K. Q. Abdelfadeel, V. Cionca, D. Pesch, Fair adaptive data rate allocation and power control in lorawan, in: 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), 2018, pp. 14–15. doi:10.1109/WoWMoM.2018.8449737.

[23] M. Slabicki, G. Premsankar, M. Di Francesco, Adaptive configuration of lora networks for dense iot deployments, in: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2018, pp. 1–9.

[24] S. Li, U. Raza, A. Khan, How agile is the adaptive data rate mechanism of lorawan?, in: 2018 IEEE Global Communications Conference (GLOBECOM), IEEE, 2018, pp. 206–212.

[25] N. Benkahla, H. Tounsi, Y.-Q. Song, M. Frikha, Enhanced adr for lorawan networks with mobility, in: 2019 15th International Wireless Communications Mobile Computing Conference (IWCMC), 2019, pp. 1–6. doi:10.1109/IWCMC.2019.8766738.

[26] Y. Li, J. Yang, J. Wang, Dylora: Towards energy efficient dynamic lora transmission control, in: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, 2020, pp. 2312–2320. doi:10.1109/INFOCOM41043.2020.9155407.

[27] D.-T. Ta, K. Khawam, S. Lahoud, C. Adjih, S. Martin, Lora-mab: A flexible simulator for decentralized learning resource allocation in iot networks, in: 2019 12th IFIP Wireless and Mobile Networking Conference (WMNC), 2019, pp. 55–62. doi:10.23919/WMNC.2019.8881393.

[28] I. Ilahi, M. Usama, M. O. Farooq, M. Umar Janjua, J. Qadir, Loradrl: Deep reinforcement learning based adaptive phy layer transmission parameters selection for lorawan, in: 2020 IEEE 45th Conference on Local Computer Networks (LCN), 2020, pp. 457–460. doi:10.1109/LCN48667.2020.9314772.

[29] Y. Yu, L. Mroueh, S. Li, M. Terré, Multi-agent q-learning algorithm for dynamic power and rate allocation in lora networks, in: 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, 2020, pp. 1–5. doi:10.1109/PIMRC48278.2020.9217291.

[30] C. J. Watkins, P. Dayan, Q-learning, Mach. Learn. 8 (1992) 279–292.

[31] G. Park, W. Lee, I. Joe, Network resource optimization with reinforcement learning for low power wide area networks, EURASIP Journal on Wireless Communications and Networking 2020 (2020) 1–20.

[32] T. Mai, H. Yao, N. Zhang, W. He, D. Guo, M. Guizani, Transfer reinforcement learning aided distributed network slicing optimization in industrial iot, IEEE Transactions on Industrial Informatics 18 (2022) 4308–4316.

[33] A. Ivoghlian, Z. Salcic, K. I.-K. Wang, Adaptive wireless network management with multi-agent reinforcement learning, Sensors 22 (2022) 1019.

[34] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, S. Whiteson, Counterfactual multi-agent policy gradients, in: Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.

[35] M. C. Vuran, I. F. Akyildiz, Channel model and analysis for wireless underground sensor networks in soil medium, Physical communication 3 (2010) 245–254.

[36] R. W. King, M. Owens, T. T. Wu, Lateral electromagnetic waves: theory and applications to communications, geophysical exploration, and remote sensing, Springer Science & Business Media, 2012.

[37] M. Bor, J. E. Vidler, U. Roedig, Lora for the internet of things, in: EWSN '16 Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks., 2016, pp. 361–366.

[38] 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.

[39] M. L. Littman, Markov games as a framework for multi-agent reinforcement learning, in: Machine learning proceedings 1994, Elsevier, 1994, pp. 157–163.

[40] J. K. Gupta, M. Egorov, M. Kochenderfer, Cooperative multi-agent control using deep reinforcement learning, in: International conference on autonomous agents and multiagent systems, Springer, 2017, pp. 66–83.

[41] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533.

[42] L. Zhang, J.-H. Jiang, P. Liu, Y.-Z. Liang, R.-Q. Yu, Multivariate nonlinear modelling of fluorescence data by neural network with hidden node pruning algorithm, Analytica Chimica Acta 344 (1997) 29–39.

[43] K. Mikhaylov, J. Petaejaejaervi, T. Haenninen, Analysis of capacity and scalability of the lora low power wide area network technology, in: European Wireless 2016; 22th European Wireless Conference, 2016, pp. 1–6.

[44] G. L. Schaefer, M. H. Cosh, T. J. Jackson, The usda natural resources conservation service soil climate analysis network (scan), Journal of Atmospheric and Oceanic Technology 24 (2007) 2073–2077.