Federated Reinforcement Learning for Automated LoRaWAN Management in Industrial IoT

Ameer Ivoghlian(✉), Zoran Salcic, and Kevin I-Kai Wang

The University of Auckland, Auckland 1010, New Zealand
aivo002@aucklanduni.ac.nz, {z.salcic,kevin.wang}@auckland.ac.nz

Abstract. The scale of wireless networks is growing rapidly to support next generation industry 5.0 applications. LoRaWAN offers features such as low power consumption and long communication range, beneficial in many industrial applications. However, this comes at the cost of low data rates and link margins that rely on coding gain and careful network planning. These qualities make LoRaWAN networks vulnerable to interference and data-loss, which are particularly problematic in large-scale and mobile networks. It is vitally important that LoRaWAN networks are automatically and adaptively managed, to ensure that resources are optimally allocated, while satisfying multiple application requirements. In this paper, a novel multi-agent federated reinforcement learning framework is designed to enable knowledge sharing across multiple agents, and to resolve the issue of limited sampling coverage deep reinforcement learning. The goal is to allow individual node agents to adaptively control their network parameters to satisfy their own application requirements, while ensuring fair access across all nodes. The proposed method is evaluated against a non-federated approach. The results show that the proposed federated reinforcement learning approach can overcome the cold-start problem when new devices join a network. The proposed approach also demonstrates the ability to support mobile nodes travelling through locations, with changing network conditions, while maintaining communication performance.

Keywords: Federated Learning · Federated Reinforcement Learning · LoRaWAN · Industrial Internet of Things

1 Introduction

The trend of intelligent and collaborative manufacturing demands large-scale, geographically distributed networks to support the next generation Industrial Internet of Things (IIoT). LoRaWAN (LoRa Wide Area Network), like other LPWANs (Low Power Wide Area Networks), supports long-range and low power communication, and is becoming increasingly popular. However, the long-range communication comes at a price of low bandwidth and low data rate. In addition,

A. Zaslavsky et al. (Eds.): MobiQuitous 2023, LNICST 593, pp. 136–155, 2024.
https://doi.org/10.1007/978-3-031-63989-0_7

LoRaWAN operates in the international unlicensed ISM (Industrial, Scientific, and Medical) radio bands, which offer easy access, but can be very crowded [1]. Therefore, it is critical to ensure the network resource is automatically and optimally allocated to the connected devices to meet their application requirements.

The crowded spectrum, large number of nodes, and low data rate bring fundamental challenges to offering fair and smooth access to every connected node. The fact that different nodes may have different application requirements; for example different data delivery rate and power consumption, makes the problem even more complex. Therefore, an automated network management solution is necessary to ensure satisfaction of co-existing applications requirements and fair access to network resources for all connected nodes [2].

While LoRaWAN can be deployed in many industrial applications, in this study, we are targeting the specific scenario of indoor positioning and tracking applications, which is critical for warehouse and logistics management. For example, it is becoming increasingly important to be able to track goods/products, moving machines (e.g. forklifts), and employees, to ensure smooth logistics flow. Once location data are collected, they will need to be transmitted to a central server for higher level planning and management. LoRaWAN, which offers large coverage and low power consumption, is capable of collecting data from sensing devices (nodes) attached to stationery or moving objects.

However, location information has different importance to different objects and applications. For example, human location may be tracked for the purpose of safety and surveillance, and therefore it is critical to ensure regular data delivery so that location information is constantly updated. On the other hand, pallet tracking may only need to happen when a pallet changes its location, to prolong node operating life (i.e., lower power consumption) instead of constantly tracking its movement. Additionally, within a warehouse or logistics centre, due to different layouts, product content, and deployed network infrastructure, network performance can vary drastically at varying locations. Therefore, it is important that devices can adapt quickly to the changing environment to ensure reasonable communication performance.

While LoRaWAN has its own adaptive data rate (ADR) mechanism, it is designed for static network deployment and is not suited to rapid change after the initial configuration. This obviously does not offer good performance in a warehouse setting when the shelving contents are constantly changing and the objects/humans/machines are frequently moving. There are several machine learning based approaches developed to address this issue and offer better performance. However, these approaches also suffer some common issues. First, most machine learning algorithms require a large amount of training data, which is challenging to achieve in a typical IIoT system [3]. Second, the trained model is very often not generalised, and hence network performance may be degraded at certain physical locations within a warehouse, because no prior knowledge was acquired from that location [4].

In this study, we propose a new multi-agent-based federated reinforcement learning approach for automated and adaptive network management, specifically

targeting LoRaWAN. Each individual IoT node has its own agent which interacts with its environment and decides how its network parameters are configured to satisfy its requirements. While individual devices and agents conduct their own learning, periodically all the agent models are aggregated and a global model is redistributed back to each individual agent. The goal is to overcome the challenge of limited sampling coverage through knowledge sharing using federated learning, and therefore achieve a more generalised and robust model at the agent level [5].

The rest of the paper is organised as follows. Section 2 provides an overview of related works on existing federated and reinforcement learning, and their associated challenges. The motivating scenario and the overall framework architecture are presented in Sect. 3, followed by the detailed methodology and implementation explained in Sect. 4. Section 5 discusses the experimental set-up, scenarios, and results. We conclude this study with promising perspectives regarding future research directions in Sect. 6.

2 Related Works

2.1 Reinforcement Learning in Wireless Network Management

With the growing complexity of wireless networks, and the limited resources, research efforts have been focused on optimising the access and allocation of the available resources.

Huang et al. [6] presented a decentralised reinforcement learning approach using the policy hill climbing (PHC) algorithm to improve LoRaWAN performance by reducing the collision rate. The results are compared with the slotted ALOHA protocol and demonstrate better communication performance.

Nisioti et al. [7] proposed a different MAC layer design targeting wireless sensor networks in general that use the irregular repetition slotted ALOHA (IRSA) protocol. A mult-agent reinforcement learning approach is used where the agents are deployed on individual nodes. The proposed reinforcement learning is based on tabular Q-learning, with limited state and action spaces to facilitate the building of the Q table. The performance is further enhanced by using a virtual experience approach to achieve accelerated learning. The results demonstrate that the proposed approach is able to find a near-optimal solution for the IRSA protocol.

Yu et al. [8] proposed using multi-agent reinforcement learning to dynamically control the scaling factor and transmission power of individual nodes in LoRaWAN, with the aim to improve communication reliability by reducing the chances of collision, while simultaneously reducing power consumption. The state space is simple and consists of only a binary indicator of connection reliability for each node. The traditional tabular Q-learning is used and the reduced state space dimensionality allows efficient computation and construction of the Q table. The approach is evaluated on a small-scale static network that contains only 5 nodes (and their corresponding agents), and the results show both reduced collision and power consumption.

Fedullo et al. [9] used the SARSA (State-Action-Reward-State-Action) algorithm for LoRaWAN management. Similar to Q-learning, SARSA is a tabular-based method, and hence it is also limited to relatively small problem spaces, to ensure sufficient observation (or sampling) coverage. To address this issue, the training is done by sharing and populating the Q table between all nodes to achieve a more general and robust model. This is likely to be achieved by including the signal-to-noise ratio in the state space and the reward function, in order to generalise the learning with respect to node position. The approach is evaluated with the standard ADR approach of LoRaWAN, under a static network scenario. The results demonstrate improved communication reliability.

Generally, many reinforcement learning methods attempt to simplify the complexity of their state space to allow, for example, low computational complexity and creation of the Q table with limited sampling coverage. While the results demonstrate better performance, the resultant models lack generality and cannot manage network dynamics and node mobility very well. Unlike traditional reinforcement learning, deep reinforcement learning approaches have been developed to accommodate more complex problems and applied to a wider range of wireless network management applications.

Zhu et al. [10] looked at how to find an optimal transmission schedule in a cognitive IoT network scenario using deep reinforcement learning. The goal is to address the issue of poor communication efficiency in modern IoT applications when the spectrum is very crowded. The simulation results demonstrate that the proposed approach achieves similar performance as the strategy iteration algorithm, which can provide the optimal solution, but the computation complexity is not acceptable for wireless network applications.

Ilahi et al. [11] proposed using deep reinforcement learning (specifically Double DQN) to configure the communication parameters in LoRaWAN at runtime. The state space consists of several adjustable parameters and the distance between node and the base station. The action space consists of two configuration parameters, the scaling factor and the transmission power. The reward function considers packet delivery ratio (PDR), airtime, and min-max normalised transmission power (TXP). The approach was evaluated using a simulated network with 100 nodes and with mobile node scenarios. The simulation results demonstrate an improved PDR performance compared with the approach implemented in the LoRa-MAB simulation framework [12].

In [13], the authors proposed a single-agent reinforcement learning approach for automatically managing a large-scale LoRaWAN by considering application awareness and fairness in allocating network resources through a reward function. The application awareness is reflected in the reward function. Experimental results showed that the proposed reinforcement learning approach offers better network performance than standard ADR when interference is present.

In [14], the authors further proposed a multi-agent-based reinforcement learning approach with a reward function that reflects application awareness and network fairness at individual node level to allow quicker responses to changing network dynamics while supporting the overall network access fairness.

2.2 Federated Reinforcement Learning in Wireless Network Management

While deep reinforcement learning approaches are able to handle more complex problem spaces, they still suffer from limited sampling coverage. If certain situations have not been observed (or sampled) before, the model may perform poorly when those situations arise. Different approaches, such as [9], are proposed to share the knowledge of multiple nodes and to enrich one global agent's experience, with the compromise of limited consideration of individual node application requirements. Federated reinforcement learning is becoming a promising direction in solving the issue of sampling coverage. However, there are limited federated reinforcement learning efforts related to automating wireless network management.

One popular application for federated reinforcement learning is in the control of Radio Access Networks (RAN) that offers more interoperable network infrastructure. Cao et al. [15] proposed a federated reinforcement learning approach to find a user access control policy for open RAN, with the aim of optimising the long-term system throughput while preventing frequent user handover. In [16], the authors proposed horizontal federated reinforcement learning for finding the more efficient scheme for RAN slicing that will improve the overall network throughput.

Federated reinforcement learning is also more commonly applied to solving complex resource allocation and task scheduling problems. For example, Zhang et al. [17] proposed a method based on DQN to find the optimal transmission mode selection and resource allocation for cellular-based vehicular networks. Tianqing et al. [18] proposed a new concurrent federated reinforcement learning approach for finding a resource allocation policy that can improve the task completion speed and efficiency across an IoT-based edge computing scenario.

Unlike the existing approaches, this study aims to explore the possibility of using multi-agent federated reinforcement learning in finding the optimal network configuration for individual nodes within a LoRaWAN. The goal is to allow individual agents to automatically and adaptively configure their network parameters in response to their application requirements and changing network conditions, while sharing their knowledge/experience with other agents periodically to achieve more robust agent models across the entire network.

3 Federated Reinforcement Learning Framework for Adaptive Network Management

3.1 Motivating Scenario

In the next generation IIoT, the need for multiple applications to co-exist in an industrial site will be inevitable. These applications (and their associated nodes) are likely to occupy the same radio spectrum. Therefore, in this study, we aim to address two challenges that have been overlooked in modern IIoT systems, which are to tolerate multiple co-existing applications, and drastically changing

network dynamics within an industrial site. These two challenges can be easily illustrated by a real-world application of human/object tracking in a warehouse and logistics management context, as shown in Fig. 1.

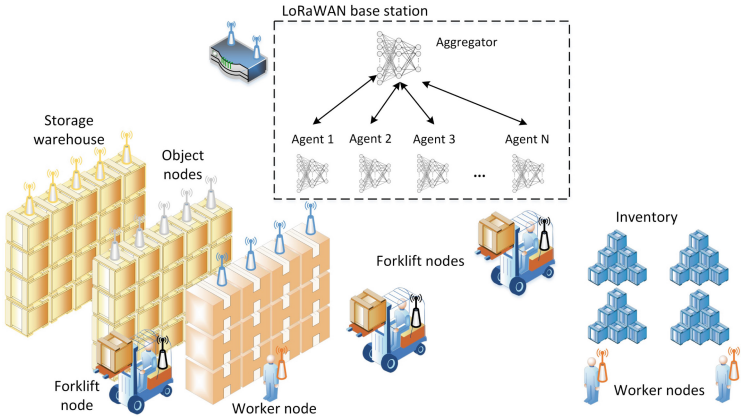

Fig. 1. The federated reinforcement learning framework architecture.

It has been demonstrated previously that location information and location-based services are extremely useful for warehouse/logistics control applications. However, different tracking applications may have different requirements. For example, an application for tracking workers may need to ensure a balanced performance of reliable communication and low energy consumption to ensure a worker's position can be tracked accurately and for a long period of time. In contrast, tracking moving machinery such as forklifts may have lesser energy consumption constraints, and tracking an object (e.g., a pallet) may have lesser real-time constraints. In addition, within a warehouse, depending on the deployed infrastructure, the warehouse layout, and the storage content, the network dynamics may change drastically across space causing variations of network performance.

In this study, three types of nodes are considered, namely worker nodes, forklift nodes, and object nodes, to reflect the following three application requirements:

1. Worker nodes: These nodes are typically wearable devices that have a high energy constraint. The nodes also need to reliably communicate their location data to the base station.
2. Forklift nodes: These nodes are installed on moving machinery, which have lower energy constraints, but require fast and reliable communication to ensure location information is transmitted to the base station.
3. Object nodes: These nodes are typically embedded devices that have very high energy consumption constraints. However, location information is only

needed at certain fixed locations so real-time communication is typically not necessary.

This motivating scenario will be used to establish the proposed federated reinforcement learning framework and will also be used in the evaluation to guide the experiment design.

3.2 Federated Reinforcement Learning Framework for Adaptive LoRaWAN Management (FRL-ALM)

Most existing (deep) reinforcement learning approaches suffer from the issue of sampling coverage. This is mainly due to the fact that individual agents are working in silos. In such cases, each agent can only observe a limited amount of data and hence the agent may perform poorly in an unknown situation. This limitation will be amplified in networks with mobile devices and with varying network conditions, like the ones in our motivating scenario.

Federated learning, as a distributed machine learning approach, is a natural step to be integrated with reinforcement learning to achieve experience/knowledge sharing across multiple nodes (and their corresponding agents). In this study, a unique Federated Reinforcement Learning framework for Adaptive LoRaWAN Management (FRL-ALM) is proposed to allow reinforcement learning to be conducted independently for each node (and agent) based on their own application requirements, while sharing their experience with one another so that the model of each individual agent can become more generalised and robust to changing locations and environmental conditions (i.e. network dynamics).

As mentioned in Sect. 1, LoRaWAN's long-range and low power features come at the price of limited data rate. Therefore, the proposed FRL-ALM framework is designed to incur only a small amount of communication overhead. In ordinary multi-agent reinforcement learning, the agents typically reside on the corresponding nodes, which allows nodes to operate independently according to their own situation. However, this also means that the nodes are running in silos and can only observe and react to what they have observed. In the proposed framework, while each node still has its corresponding agent, all agents are running on the base station.

Each agent still trains its own model based on its own state space (i.e. the environment observed by each node). Periodically, the aggregator will perform an aggregation process to collect and fuse all agent models. Once the aggregation is performed, the global model will be distributed to all agents and the agents will adopt the new model to update their network configurations (i.e. their action spaces) and carry on with their own learning until the next aggregation process. Since LoRaWAN typically utilises a star topology, a typical horizontal federated learning architecture is adopted across all agents and the aggregator.

Although running all agents on the base station will incur a higher workload, the base station typically has higher computational power and is more likely to have a permanent power source. More importantly, this architecture also significantly reduces the communication overhead of sending all the model parameters

of every agent to the base station. The aggregation process allows agents to share their knowledge with each other and hence the learned model will be more generalised and robust against changing network dynamics and node position. This will be evaluated and verified in our experiment with moving node scenarios. In addition, the global model also significantly reduces the training time, by providing any new node with the latest global agent model and is able to achieve a reasonable level of performance within a much shorter time-frame.

4 FRL-ALM Implementation

Extending from the Federated Reinforcement Learning framework presented in Sect. 3, this section discusses further, the actual design and implementation of the framework that can be fully integrated and deployed with the LoRaWAN protocol, in real-world industrial applications.

4.1 Multi-agent Reinforcement Learning

Reinforcement learning offers superior long-term performance for complex questions by interacting with the environment through actions and adjusting its agent's future action based on a reward function. Within a typical IIoT network, each individual node may have its own application requirements and therefore may require different network configurations to satisfy those requirements. In the previous study, multi-agent-based reinforcement learning has demonstrated better performance at both the individual agent and overall network level [14]. Individual agents will also allow their corresponding nodes to respond more quickly to their changing network dynamics and application requirements.

In this study, we continue with a multi-agent-based design. However, the agents are running in the base station instead of individual nodes in order to reduce excessive communication overhead. This is critical for LoRaWAN, which has a low data rate, and in some regions is further constrained by duty cycle limitations. In the proposed framework, each individual agent observes its own environment that is represented by the following state space, S (1):

$$S = \{PDR, ECP, T_{PDR}, T_{ECP}, U, PL\} \tag{1}$$

$$PDR = \frac{Packets\ received}{Packets\ sent} \tag{2}$$

$$ECP = V_{supply} \times I_{supply} \times T_{packet} \tag{3}$$

where the PDR is the observed packet delivery ratio of each node calculated by the base station as in (2); the ECP is the energy consumption per packet calculated by each node as in (3); the T_{PDR} and T_{ECP} are the target PDR and ECP set by the corresponding application requirements of each node; U is the overall network utilisation observed and calculated by the base station; and PL

is the average path loss of the node which gives an indication of the distance between the node and its base station and is calculated by taking the received signal strength and subtracting the transmission power value in the packet.

Referring to Fig. 2, to populate the state table, each node will inform the base station of its T_{PDR} and T_{ECP} during its initialisation phase (i.e. when it is first connected to the network). Each node will also include the ECP in each packet it sent. Other state space variables can be calculated by the base station, which has a better global view of the network. This allows for minimum communication overhead caused by agent learning. In comparison, if the agents are run on individual nodes, PDR, U, and PL will need to be provided to each individual node with more downlink communications, and the model parameters of all agents will need to be transmitted to the base station to enable aggregation, which significantly increases the overheads. The overall implementation and interaction of the base station and each node is shown in Fig. 2.

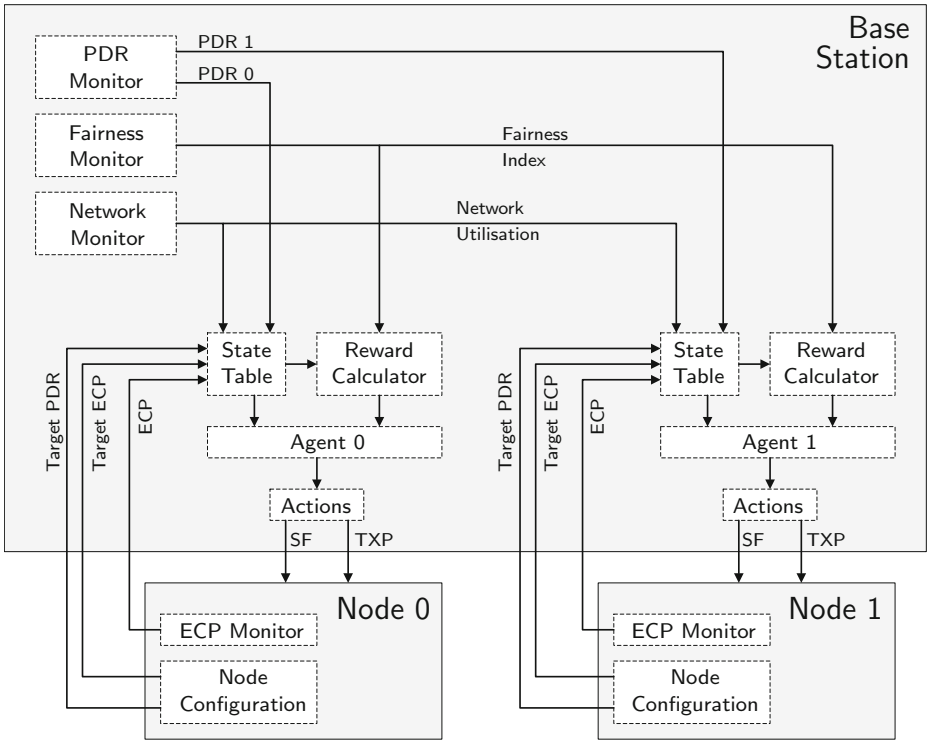

Fig. 2. FRL-ALM implementation architecture.

The purpose of the action space of each agent (and its corresponding node) is to interact with the environment and iteratively adjust its behaviour to satisfy its application requirements. In the proposed framework, the action space, A, is defined as:

$$A = \{SF, TXP\} \tag{4}$$

where SF is the spreading factor, which is a LoRaWAN-specific parameter that can be configured to balance the transmission range, energy consumption, data rate, and resistance to interference; and TXP is the transmission power. These two parameters can easily be configured and have direct impacts on the communication performance of the specific node.

The function of the reward function in reinforcement learning is to determine whether the performed action is beneficial towards the goal. In this study, the goals are to automatically manage and satisfy the application requirements of every individual node and at the same time ensure a fair resource allocation (or access) within the entire network. A unique reward function is shown in (8)–(6), which accommodates both the individual node's requirements and the overall network access fairness. It was first proposed in [14].

$$P_n = \prod_{m=0}^{M-1} Q_m \tag{5}$$

$$Q_m = \begin{cases} \frac{C_m}{T_m}, & C_m < T_m \\ 1, & C_m \geq T_m \end{cases} \tag{6}$$

$$f = \frac{(\sum P_n)^2}{N \sum (P_n^2)} \tag{7}$$

$$R = \frac{1 - 2|P_n - f| + |1 - 2f|}{1 + |1 - 2f|} \tag{8}$$

P_n, as defined in (5), captures how well a node is meeting its application requirements - the first goal. It is the product of the satisfaction Q for each metric m, with M representing the metric count. This research uses metrics PDR and ECP. Q_m is the ratio of the current measured metric value C_m to the configured target metric value T_m, capped at 1.

The second goal, fairness, is captured by f, as defined by (7). It is similar to Jain's Index [19], but has been adapted to accommodate differing application requirements. f represents how uniform each node's P_n is across the network.

Low levels of fairness are indicative of insufficient network capacity. In such cases, a reduction in network utilisation should be encouraged, which will likely result in a reduction of average P_n. To improve fairness, nodes with higher P_n should have a greater incentive to reduce. When there is excess network capacity available - a scenario which naturally results in higher fairness - nodes can increase their P_n without limit. It follows then, that when fairness is moderate, nodes with high P_n should be encouraged to reduce, and nodes with low P_n should be encouraged to increase, effectively motivating all nodes to target a moderate P_n.

To achieve this, the reward function R shown in (8), takes a triangular form, with the peak of the triangle at the current network fairness value f, and the

reward dropping off either side of this value. The $1-2|P_n - f|$ term in the reward function creates this triangular form. The remaining terms apply an offset and scaling factor to constrain the reward value between 0 and 1.

4.2 Federated Multi-agent Reinforcement Learning

In the proposed framework, the deep reinforcement learning approach, DQN (Deep Q Network), is adopted to implement each individual reinforcement learning agent. Different to the traditional Q-learning approach, which aims to derive a tabular result of the state-action function (i.e., the Q function), the DQN uses a neural network to approximate and estimate the Q values. While DQN has an improved ability to handle complex problems by replacing the Q table with a neural network, the method still suffers from the limited sampling space coverage and lack of generality. Since each individual agent only has limited observation of the network environment, the coverage of the sampling space is poor. This leads to poor performance when faced with network dynamics, when nodes are moving, or due to changing environmental conditions. Therefore, the federated learning approach is adopted in the proposed framework to facilitate experience sharing from every agent to ensure the agent model has better generality and is more robust against changing network dynamics.

Since all the agent training and model aggregation are performed in the base station, it is important to ensure a reasonable computational complexity. In the proposed framework, we adopted the typical FedAvg approach to aggregate the neural network parameters (i.e. the weights and biases of the DQN neural network). Figure 3 shows the overall federated learning process. Within the framework, individual agents operate independently, to interact with the network environment and adjust configuration parameters to iteratively attempt to satisfy their own application requirements. Periodically, the aggregator will request that all agents share their model parameters. Once parameters are received from all agents, aggregation will be performed and the aggregated model parameters will be distributed to all agents to update their models and action values. In the proposed framework, the node path loss component within the state space is introduced for the purpose of identifying and differentiating node behaviours incurred due to network dynamics at different distances from the base station. If a new node (and hence its agent) is joining the network, the latest global model parameters will be provided to the agent to allow the new node to achieve reasonable performance without excessive training time.

4.3 Framework Simulation Environment

The proposed approach has been developed in a modelling environment, consisting of a network simulation framework, and interconnected reinforcement learning tools. Building on the work of Bor et al. [20], a LoRaWAN modelling framework has been designed using Python and the SimPy library [21]. The framework builds the foundation for emulating LoRaWAN and evaluating the

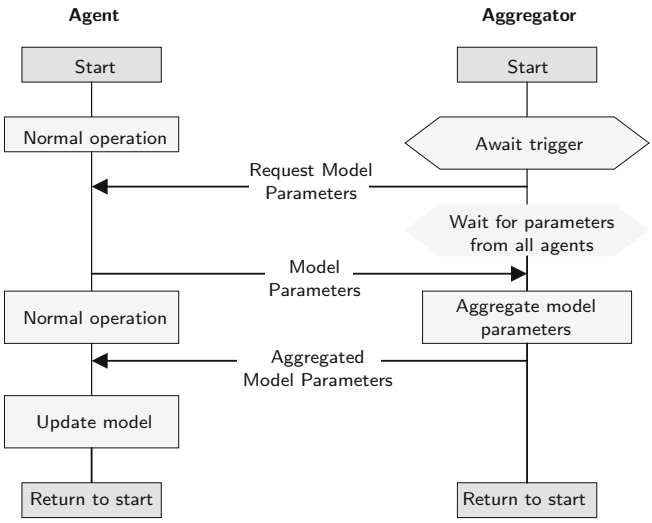

Fig. 3. FRL-ALM federated learning process.

proposed method by enabling the execution and interaction of multiple independent processes. In order to model the propagation environment, Log Distance path loss and shadow fading are used. By adjusting the path loss exponent and the standard deviation of the normally distributed shadow fading component, different transmission environments can be simulated. In this study, the path loss exponent has been configured to emulate an indoor warehouse environment.

The LoRa communication mechanisms are implemented following the processes and equations in Semtech's LoRa Application Notes [22,23], which allow for the calculation of packet link margin and detection of packet collision. The base station model utilised for our implementation contains an eight-channel radio, enabling reception and demodulation on eight channels, while simultaneously transmitting on one channel.

The reinforcement learning agents have been implemented using the Tensorforce library [24], which is built on top of Tensorflow [25]. The Tensorforce library implements a range of reinforcement learning algorithms and provides an interface for application integration. The modular structure of Tensorforce makes it possible to substitute other simulation frameworks, or even a physical LoRaWAN network, without changing the reinforcement learning components. The reinforcement learning algorithm can also be changed, without requiring changes to the simulation framework.

5 Experimental Evaluation

5.1 Experiment Setup

As mentioned in the previous sections, industrial environments are a targeted use case for LoRaWAN and the proposed approach. In this study, the target scenario is specifically the different types of tracking scenarios in a real-world warehouse and logistics management application.

LoRaWAN base stations typically have the capability of demodulating multiple (up to 8) channels simultaneously. From a network capacity perspective, each channel is independent of others. A single channel containing 32 nodes is equivalent to an eight-channel network that consists of 256 nodes. This characteristic allows simulation to be scaled down and accelerated. The scenario is simulated for an indoor environment, containing many absorbing bodies (e.g., stored goods) and reflective surfaces. The average path loss is therefore high, as is the variability. Table 1 lists the specific simulation parameters. The agent training parameters are also listed in Table 2, with the node positions randomly generated at the beginning of each episode.

Table 1. LoRaWAN Simulation Parameters

Parameter	Value
Nodes	32
Cell Radius	100 m
Node Distribution	Random Uniform
Spreading Factors	SF7 .. SF12
Transmission Power Range	−4 dBm .. 14 dBm
Transmission Power Steps	2 dB
Coding Rate	4/5
Frequency Band	915 MHz
Uplink Channel Bandwidth	125 kHz
Uplink Channel Spacing	200 kHz
Uplink Channels	1
Message Period	10 min
Payload Length	48 Bytes
Frame Length	532 bits
Path Loss Exponent (λ)	5
Shadow Fading Std Dev (σ)	9

5.2 Scenario I: Learning Efficiency for New Nodes

The first objective of the proposed federated reinforcement learning is to overcome the limitation of sampling coverage and achieving a more generalised agent

Table 2. Reinforcement Learning Training Parameters

Parameter	Value
Training length	100 episodes
Episode length	10,000 iterations
Memory	100 episodes
Batch Size	4
Discount Factor (γ)	0.1
Exploration (ϵ)	0.1
Learning Rate	e^{-3}
Update Frequency	1

model. A principle motivation for generalising the state space and aggregating the agent parameters is to allow agents to benefit from the experience of others. One advantage of the proposed approach can be demonstrated through the introduction of new nodes to the network, where these new nodes (or their agents) typically experience the cold start problem due to the agent's lack of observations, experiences, and interactions with its environment. A new node with no training, or with generic training that is poorly matched to its deployed situation, will initially perform poorly until it can learn about its environment. This is a very time-consuming process, especially in LoRaWAN where the communication and experience gathering is less frequent (e.g., a few times per hour).

The cold-start problem of new nodes can be significantly improved by taking advantage of the knowledge of other nodes in the network. In this scenario, the PDR target T_{PDR} is set to 0.5, and the target ECP T_{ECP} is set to an inverse normalised value of 0.5 (406 mJ per packet). This represents an application with balanced requirements on network reliability and energy consumption. Immediately after joining the network, the node's agent will request the current aggregated model parameters. The results presented in Fig. 4 and 5 show the PDR and ECP performance of a joining node with and without federated learning.

A node taking advantage of federated learning is compared with one which does not. It is strikingly obvious that, despite having similar levels of ECP, the PDR performance is vastly superior for the node that has the advantage of aggregated knowledge from the rest of the network. The poorly performing node is clearly using completely inappropriate parameters, and it will take significant time to explore the problem space and find appropriate ones.

The results of this scenario demonstrate that, with federated learning, the newly added nodes avoid the cold start problem, attaining high levels of performance immediately. It achieves this by taking advantage of the prior experience of all other nodes in the network.

5.3 Scenario II: Networks with Moving Nodes

The second objective of the proposed approach is to achieve a more robust model that can accommodate drastically changing network dynamics and node mobil-

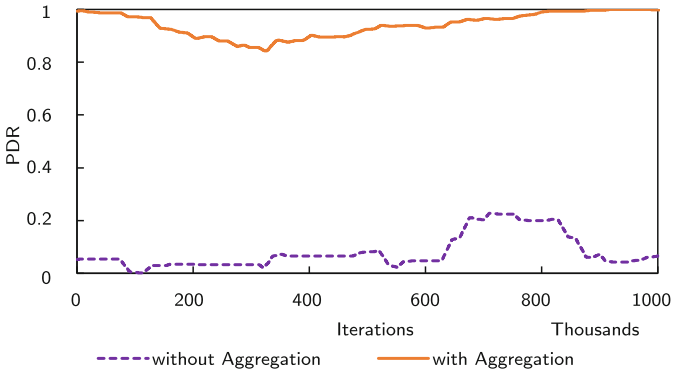

Fig. 4. PDR performance for a newly added node with and without federated learning.

ity. In a simplified scenario where a network consists of only stationary nodes, it is ok for individual nodes (and their agents) to operate independently in silos and only observe their nearby environment to achieve a reasonable performance. This is however rarely the case in real-world industrial applications, especially in the case of warehouse logistics tracking, where nodes are completely mobile with different tracking and communication requirements as discussed in Sect. 3. An agent must be able to respond to this movement without significant learning time. This experiment aims to evaluate if the proposed approach will allow nodes to maintain an acceptable level of performance, despite a significant change in their environment.

In this scenario, two types of mobile node are evaluated. The first represents a person moving in the warehouse on a forklift. As the node is on a person, power consumption is a concern, and so the metric targets are set at 0.4 and 0.6 for PDR and ECP respectively. The second node is attached to a forklift. It follows the same path as the first moving node but is attached to the vehicle. Its energy consumption is therefore less important, and the metric targets are set to 1.0 for PDR and 0.05 for ECP. The experiment is run with and without knowledge aggregation.

Figure 6 illustrates the PDR performance of the node attached to the person. The chart is segmented to show which position the node is in at the specific time. While both nodes show a reduction in performance when moving away from their initial position, when a node utilises federated reinforcement learning, it maintains a better and more consistent performance level and recovers more quickly when returning to its original position. Without federated learning, performance is worse, with more fluctuations and the PDR dropping to zero for some time. Even after returning to its origin, some instability can be observed, and it takes longer for the performance to return to its original level.

Figure 7 illustrates the PDR performance for the mobile node on the forklift. Unlike the pedestrian scenario, this scenario is focusing on PDR because there is

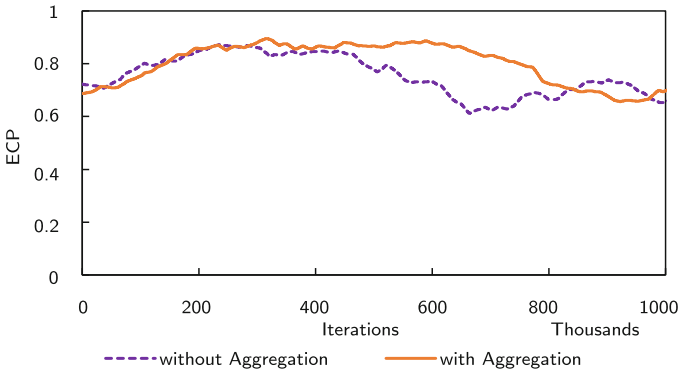

Fig. 5. ECP performance for a newly added node with and without federated learning.

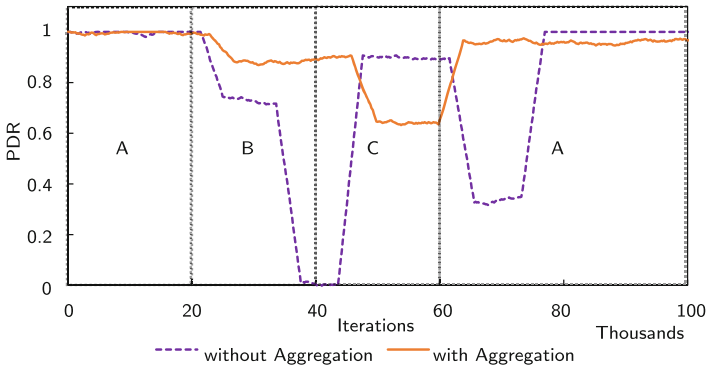

Fig. 6. PDR performance for mobile nodes on a person.

very little energy constraint. The node utilising federated reinforcement learning again offers very consistent performance, showing no impact at all caused by the node's movement. The PDR performance in the case where there is no federated learning clearly suffers more fluctuation. This performance fluctuation is likely to be worse when more complicated real-world factors such as environmental conditions, deployed infrastructures, and storage contents, are considered.

Figure 8 and Fig. 9 illustrate the ECP performance for the same two node types as above. ECP is presented as an inverted min-max normalised value, where values closer to 1 represent low energy consumption per packet, and values closer to 0 represent high energy consumption per packet. Two things can be observed in these results. Firstly, that the node utilising federated learning demonstrates intentional parameter change. It has changed position, and has chosen another set of appropriate parameters, informed by the experience of other nodes. These new parameters necessarily result in a change in ECP performance. A change which is less distinct for the node not utilising reinforcement learning.

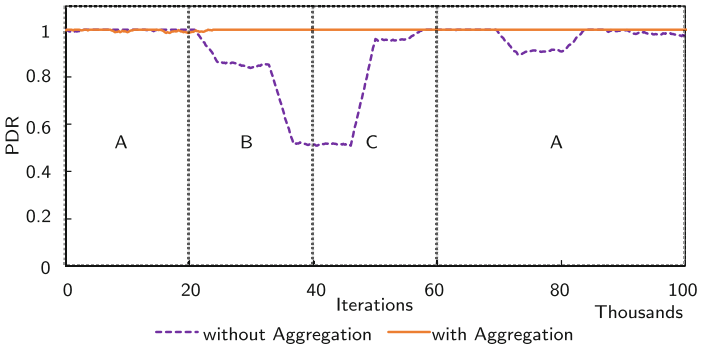

Fig. 7. PDR performance for mobile nodes on a forklift.

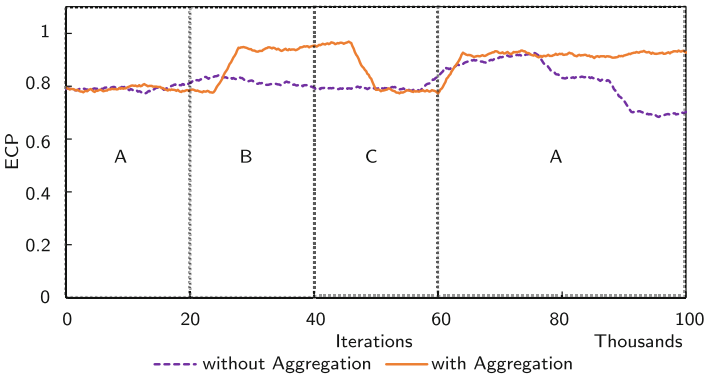

Fig. 8. ECP performance for mobile nodes on a person.

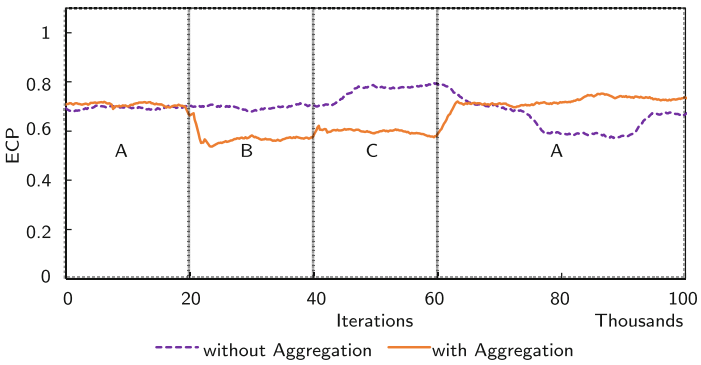

Fig. 9. ECP performance for mobile nodes on a forklift.

The second observation is the alignment with application requirements. ECP performance is clearly lower for the forklift node, as captured in its 1.0 and 0.05 metric targets for PDR and ECP respectively. This is the case regardless of whether reinforcement learning is utilised.

6 Conclusion

Future intelligent and collaborative manufacturing is underpinned by advances in next generation networks. While large-scale, long-range communication can be achieved to support large numbers of IIoT devices, the more pressing issue is how to automatically and adaptively manage the limited network resources to ensure fair access to all connected nodes, such that they can satisfy their application requirements. Many solutions have been proposed, ranging from traditional reinforcement learning to single and multi-agent deep reinforcement learning. While those solutions offer good performance in some environments, one common challenge that existing approaches are facing is the limited sampling coverage. In this study, we proposed a new multi-agent federated reinforcement learning framework specifically targeting LoRaWAN with mobile nodes. The proposed approach allows individual devices to have their own learning agents that control their independent network parameters to satisfy different application requirements. At the same time, agents can share their knowledge through federated learning to improve the agent model with better sampling coverage. The proposed method showed its ability to address the cold-start problem by achieving reasonable communication performance immediately at the start of agent operation. This also shows that the federated reinforcement learning agent model is more generalised. The proposed approach also demonstrated consistent communication performance when nodes are moving within the network with changing network dynamics. A more universal benefit is an improvement in learning efficiency. By regularly sharing experience between nodes, agents can build their models without direct experience, and much more quickly than if they attempted to gain that experience independently.

While the current results are promising, the adopted federated learning architecture and aggregation function are kept simple intentionally to reduce the overall computational complexity for the base station execution. In the future, we will investigate more sophisticated aggregation functions and other potential approaches to selectively aggregate agent models with higher credibility to achieve better performance with reasonable computation complexity.

References

1. Sherazi, H.H.R., Grieco, L.A., Imran, M.A., Boggia, G.: Energy-efficient LoRaWAN for Industry 4.0 applications. IEEE Trans. Ind. Inform. **17**(2), 891–902 (2021). https://doi.org/10.1109/TII.2020.2984549
2. Zhou, X., Yang, X., Ma, J., Wang, K.I.K.: Energy-efficient smart routing based on link correlation mining for wireless edge computing in IoT. IEEE Internet Things J. **9**(16), 14988–14997 (2022). https://doi.org/10.1109/JIOT.2021.3077937
3. Liang, W., Hu, Y., Zhou, X., Pan, Y., Wang, K.I.K.: Variational few-shot learning for microservice-oriented intrusion detection in distributed industrial IoT. IEEE Trans. Ind. Inf. **18**(8), 5087–5095 (2022). https://doi.org/10.1109/TII.2021.3116085

4. Lei, L., Tan, Y., Zheng, K., Liu, S., Zhang, K., Shen, X.: Deep reinforcement learning for autonomous internet of things: model, applications and challenges. IEEE Commun. Surv. Tutor. **22**(3), 1722–1760 (2020). https://doi.org/10.1109/COMST.2020.2988367
5. Sun, W., Lei, S., Wang, L., Liu, Z., Zhang, Y.: Adaptive federated learning and digital twin for industrial Internet of Things. IEEE Trans. Ind. Inf. **17**(8), 5605–5614 (2021). https://doi.org/10.1109/TII.2020.3034674
6. Huang, X., Jiang, J., Yang, S.H., Ding, Y.: A reinforcement learning based medium access control method for LoRa networks. In: 2020 IEEE International Conference on Networking, Sensing and Control (ICNSC), pp. 1–6 (2020). https://doi.org/10.1109/ICNSC48988.2020.9238127
7. Nisioti, E., Thomos, N.: Fast Q-learning for improved finite length performance of irregular repetition slotted ALOHA. IEEE Trans. Cogn. Commun. Netw. **6**(2), 844–857 (2020). https://doi.org/10.1109/TCCN.2019.2957224
8. Yu, Y., Mroueh, L., Li, S., Terré, M.: 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, pp. 1–5 (2020). https://doi.org/10.1109/PIMRC48278.2020.9217291. iSSN 2166-9589
9. Fedullo, T., Morato, A., Tramarin, F., Bellagente, P., Ferrari, P., Sisinni, E.: Adaptive LoRaWAN transmission exploiting reinforcement learning: the industrial case. In: 2021 IEEE International Workshop on Metrology for Industry 4.0 IoT (MetroInd4.0 IoT), pp. 671–676 (2021). https://doi.org/10.1109/MetroInd4.0IoT51437.2021.9488498
10. Zhu, J., Song, Y., Jiang, D., Song, H.: A new deep-Q-learning-based transmission scheduling mechanism for the cognitive Internet of Things. IEEE Internet Things J. **5**(4), 2375–2385 (2018). https://doi.org/10.1109/JIOT.2017.2759728
11. Ilahi, I., Usama, M., Farooq, M.O., Umar Janjua, M., Qadir, J.: LoRaDRL: deep reinforcement learning based adaptive PHY layer transmission parameters selection for LoRaWAN. In: 2020 IEEE 45th Conference on Local Computer Networks (LCN), pp. 457–460 (2020). https://doi.org/10.1109/LCN48667.2020.9314772. iSSN 0742-1303
12. 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: 2019 12th IFIP Wireless and Mobile Networking Conference (WMNC), pp. 55–62 (2019). https://doi.org/10.23919/WMNC.2019.8881393. iSSN 2473-3644
13. Ivoghlian, A., Wang, K.I.K., Salcic, Z.: Data-driven adaptive network management with deep reinforcement learning. In: Proceedings of the 2021 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech). IEEE (2021). iSBN 978-1-6654-2174-4
14. Ivoghlian, A., Salcic, Z., Wang, K.I.K.: Adaptive wireless network management with multi-agent reinforcement learning. Sensors **22**(3), 1019 (2022). https://doi.org/10.3390/s22031019. https://www.mdpi.com/1424-8220/22/3/1019
15. Cao, Y., Lien, S.Y., Liang, Y.C., Chen, K.C.: Federated deep reinforcement learning for user access control in open radio access networks. In: IEEE International Conference on Communications, ICC 2021, pp. 1–6 (2021). https://doi.org/10.1109/ICC42927.2021.9500603. ISSN 1938-1883

16. Liu, Y.J., Feng, G., Sun, Y., Qin, S., Liang, Y.C.: Device association for RAN slicing based on hybrid federated deep reinforcement learning. IEEE Trans. Veh. Technol. **69**(12), 15731–15745 (2020). https://doi.org/10.1109/TVT.2020.3033035
17. Zhang, X., Peng, M., Yan, S., Sun, Y.: Deep-reinforcement-learning-based mode selection and resource allocation for cellular v2x communications. IEEE Internet Things J. **7**(7), 6380–6391 (2020). https://doi.org/10.1109/JIOT.2019.2962715
18. Tianqing, Z., Zhou, W., Ye, D., Cheng, Z., Li, J.: Resource allocation in IoT edge computing via concurrent federated reinforcement learning. IEEE Internet Things J. **9**(2), 1414–1426 (2022). https://doi.org/10.1109/JIOT.2021.3086910
19. Jain, R., Chiu, D., Hawe, W.: A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems (1984)
20. Bor, M.C., Roedig, U., Voigt, T., Alonso, J.M.: Do LoRa low-power wide-area networks scale? In: Proceedings of the 19th International Conference on MSWiM, pp. 59–67. ACM (2016). https://doi.org/10.1145/2988287.2989163
21. Simpy - discrete event simulation for Python. https://simpy.readthedocs.io/. Accessed 30 Sept 2010
22. Semtech Corporation: SX1272/3/6/7/8: LoRa Modem Designer's Guide (2013). https://lora-developers.semtech.com/library/product-documents/
23. Semtech Corporation: LoRa Modulation Basics (2015). https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276
24. Kuhnle, A., Schaarschmidt, M., Fricke, K.: Tensorforce: a tensorflow library for applied reinforcement learning. Web page (2017). https://github.com/tensorforce/tensorforce
25. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). https://www.tensorflow.org/. Software available from tensorflow.org