Indian Journal of Geo Marine Sciences
Vol. 50 (11), November 2021, pp. 897-903

Task allocation based multi-agent reinforcement learning for LoRa nodes in gas wellhead monitoring service

Z H Ismail*, B L L Hong & A Elfakharany

Centre for Artificial Intelligence and Robotics, Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, 54100, Kuala Lumpur, Malaysia

*[E-mail: zool@utm.my]

Received 31 August 2021; Revised 30 November 2021

This paper investigates a new alternative approach to handle the tasks allocation problem that associate with numerous Long Range (LoRa) nodes in the High-Pressure High-Temperature (HPHT) gas wellhead monitoring service. A Multi-Agent Reinforcement Learning approach is proposed in this paper to overcome this problem with the Proximal Policy Optimization (PPO) is chosen as the policy gradient method. An action space is the spreading factor and other parameters such as frequency and transmission power has been kept constant. The reward function for the training process will be determined by two parameters which are the acknowledge flag (ACK) and collision between packets. Each node will be distributed across a defined disc radius. Each node will be represented as an agent. Each agent will undergo packet transmission and the packet will be evaluated according to the reward function. The results show that PPO with Multi Agent Reinforcement Learning was able to determine the optimal configuration for each LoRa node. The total reward value corresponds to the total number of nodes. Furthermore, since this study also implements the use of CUDA, the training was able to done in 200 steps and 45 minutes.

[**Keywords**: Internet of things, Monitoring, Reinforcement learning, Task allocation, Wellhead]

Introduction

High-Pressure High-Temperature (HPHT) gas wellhead plays an important role in drilling and production[1]. Its purpose is to serve the suspension point and pressure seals for casing strings. During production operations, it serves as an attach point for a Christmas Tree. In other words, wellhead is one of the most important and crucial elements on an oil rig. One of the tasks in which the operator on board need to complete is evaluation of wellhead loads. During drilling or completion phase, the angle of a wellhead is crucial as it will affect the operation.

Monitoring of wellhead as shown in Figure 1 has to be completed manually daily and frequently and this has created a burden for engineers on board as they have another task as well[2]. An optimum solution would be using an Internet of Things (IoT) device such as a laser distance measurer with Long Range (LoRa). By introducing IoT devices, the main problem faced would be power source. There are a lot of choices of power source for a typical IoT devices such as battery powered. However, for this device to be installed on an offshore oil rig there are certain rules that need to be adhere. Due to the hazardous environment on the offshore platforms, there are many regulations which oil rig workers have to follow. One such rule is all electronic plug and devices has to be monitored and maintained to a safety standard. For instance, all such devices need to be mounted into a specialised box with ATEX certification. This special engineered box prevents potentially explosive, environments of various categories, both gaseous (petrochemical mainly) and dusty such as flour mills, saw mills and some food processing plants from expose to outer environment. In an environment which filled with explosive gases, this is crucial for prevention and safety measurement. For oil rig operators, they have to monitor wellhead everyday although any additional job such as monitoring additional electronic devices is tiring for them.

Thus, it would be magnificent if the IoT device to be plug-and-play devices and maybe disposable after the battery lifespan is over. This would reduce the burden for oil rig engineers and increase their work efficiency. The challenge that needs to solve would be maximize a battery usage on an IoT device. This can be done by well design resource allocation. The design of such sophisticated system required complex computation and testing. Thus, this is where Reinforcement Learning can be implemented to find the optimum policy for such system[3].

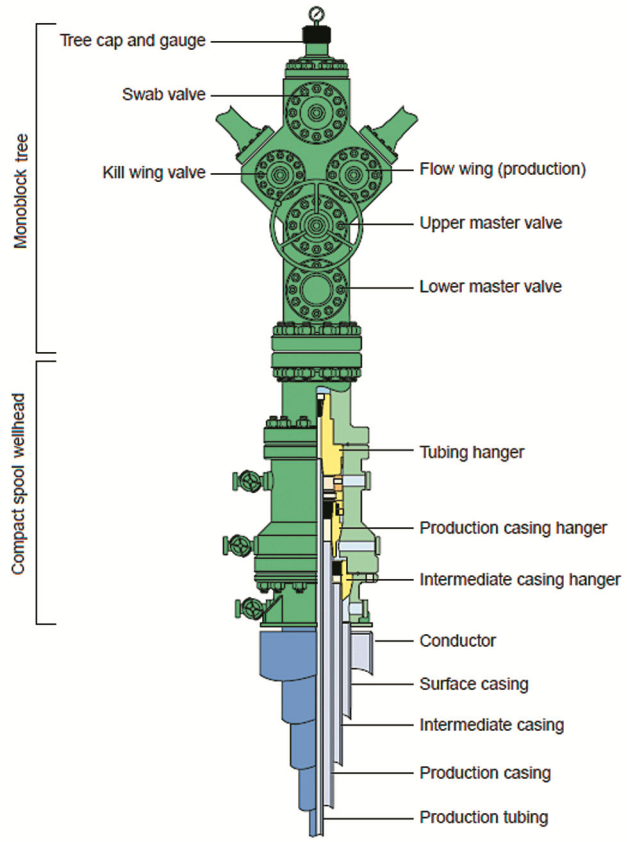

Fig. 1 — The structure of the HPHT gas wellhead[ref. 2)]

With all the regulations and condition faced on the platform, the most satisfactory approach for asset monitoring and management services with LoRa would be a battery powered. There are many types of battery may be implemented in this situation namely lithium polymer battery or LiPo battery, nickel–cadmium battery, dry cell battery and many more. Several issues need to be considered in which are cost and the safety. First, the device is disposable when its lifespan is over. Second, as the device is "disposable", the cost needs to be low. Third, such battery needs to meet the safety requirement on board which is less likely to be explosive. Thus, a dry cell battery will be suitable candidate for this device.

The device has to be run for a long period of time which is one whole day and it would be expected to run for at least several months. An IoT device with LoRa will have a fixed current consumption with corresponding action. The number of nodes and intelligent nodes would affect a lot on the system. Thus, the number of nodes will be fixed and the variable will be the task allocation on each node[4]. Task allocation may be done manually with testing and measurement; however, this will take lots of time and effort.

Related works

Due to the increased usage of wireless devices nowadays, the research trend is starting to focus on wireless devices. There are several types of wireless devices and its associated task allocation algorithm has been explored to ensure the further optimization of battery usage[5-9]. Figure 2 below summaries the recent trend in resource and task allocation algorithm and method within the IoT application.

Recently, a simulator called LoRa-MAB is being proposed in order to investigate the performance of resource allocation in LoRa WAN through simulation[5,6]. In addition, EXP3 algorithm is used to alter autonomously the decision of LoRa end-devices towards the most profitable resources (*e.g.*, spreading factors, sub-channels). Previously, only several simulators are available for simulating a LoRaWAN network. The most popular of them is LoRaSim[7], which utilize a radio propagation model. At the end of simulation, it reports the ratio of packet delivery and total energy of the network consumed. However, these simulators lack some configuration such as varying radio setting and physical settings are not considered. The manipulative parameters are spreading factors, frequency and transmission power.

Markov Decision Process (MDP) has been applied for a low power wide area network applications[8]. An agent operates according to a policy, which was expressed as an actionable distribution according to each state defined in the MDP. The value function was updated using Q-learning while a deep neural network has added to define the loss function. Moreover, two parameters were to manipulate namely spreading factor, transmission power and frequency. As a result, there are 90 fixed actions for each node. For the reward function, sum of packets received and sum of energy consumed were considered. As a result, the proposed method is improved about 15 % more the ADR. However, the proposed method is slow in learning and decrease in throughput.

A deep Q-learning model or Deep Q-Learning Model for Energy-Efficient Edge Scheduling (DQL-EES) was reported by Zhang *et al.*[9]. It was also compared with hybrid Dynamic Voltage and Frequency Scaling (DVFS) scheduling based on reinforcement learning (QL-HDS). Q-value of each DVFS technique has been calculated with a deep Q learning which consists of a stacked auto-encoder and

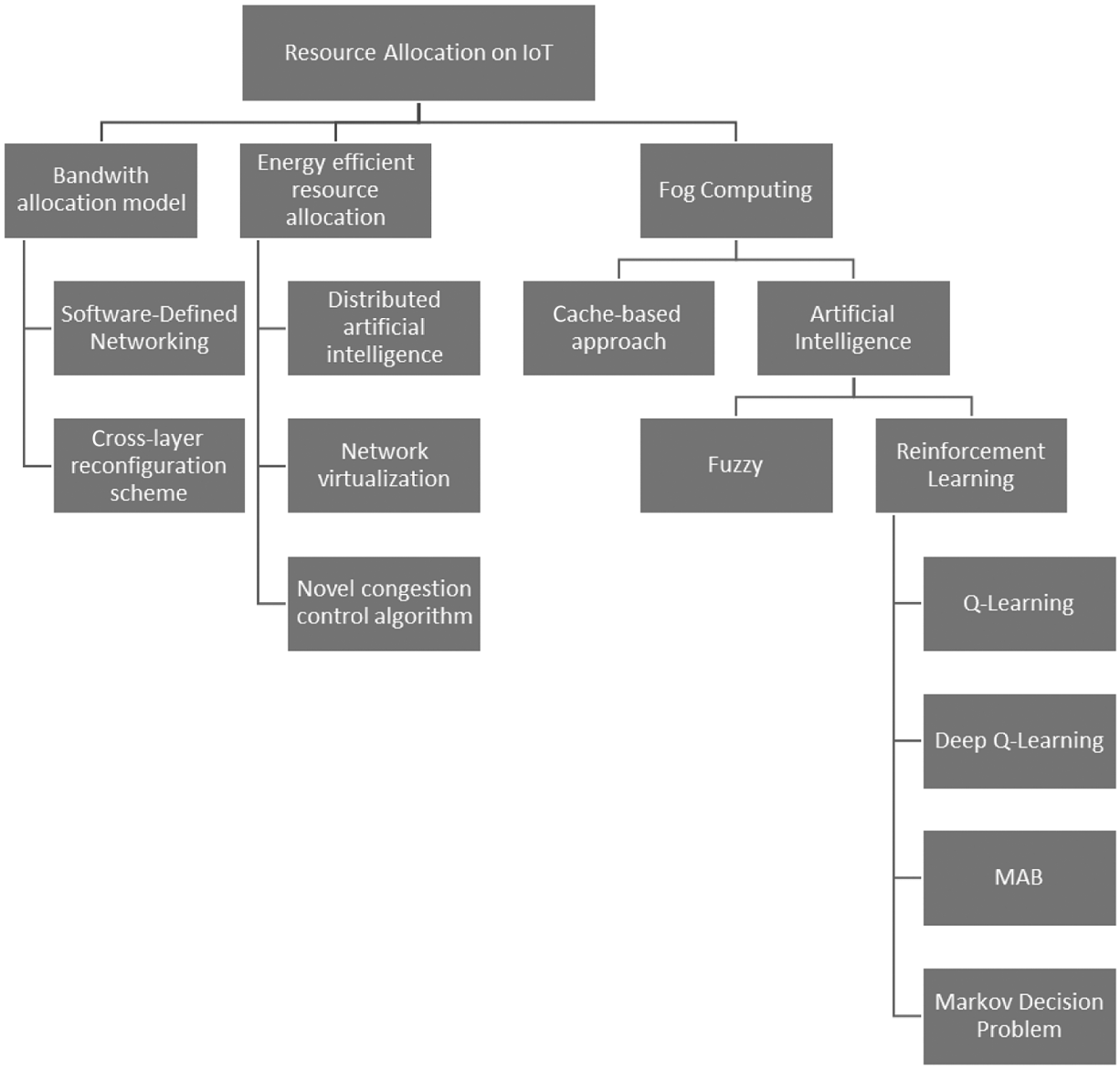

Fig. 2 — Knowledge map for task allocation

a Q-leaning model. The auto encoder was used to analyze the feature of each input system. The selected manipulative parameters are voltage and frequency. During the experiments, 20 different tasks were initiated and 4 different task sets are being bring about from the 20 different tasks. The energy consumption for each task set is used as a metric for performance. Simulation results on different task sets demonstrated that the proposed algorithm which is Deep Q Learning could save average 4.2 % energy than Conventional Q Learning.

The previous research also proposed a framework called Cooperative Deep Reinforcement Learning strategy (TAP CDQL) approach for task allocation problem[10]. They defined three types of agents namely Manager: the agent that request help, Participant: the agent which accept and perform the task, and the Mediator: the agent that assist task. Three states have been utilized namely Busy, Committed and Idle. The approach was compared with Greedy Distributed Allocation Protocol (GDAP). The methods were tested with two different settings where the first setting is the total number of agents. While, the second setting varies from 100 to 200. It proves Deep Q Learning work in task allocation problem, however, this paper did not fully exploit its ability to handle heterogeneous agent types. Furthermore, due to decentralization and reallocation features, it still has several deficiencies.

Methodology

Proposed DRL method

In this section, a proposed method of Multi-agent Deep Reinforcement Learning will be briefly discussed. A Proximal Policy Optimization or PPO is policy-based approach and it is a family of policy gradient method where policy is updated explicitly. The most commonly used gradient or loss function is given by Ben Noureddine *et al.*[10]

$$\hat{g} = \widehat{E_t}[\nabla_\theta \log \pi_\theta \ (a_t \mid s_t)\hat{A}_t] \quad \ldots (1)$$

where, π_θ represents the random or stochastic policy, while $\hat{A}_t$ is an estimator of the advantage function according to the timestep t. Thus, $\widehat{E_t}[\ldots]$ shows the observed average expectation over a batch of sample. In 2015, a trust region strategy has been introduced *i.e.* Trust Region Policy Optimization or TRPO[11]. TRPO uses Kullback-Leibler Divergence in the optimization process. Kullback-Leibler Divergence ensures the output of the execution from the new policy and will not have a big difference as compared to the old policy. In other words, the new policy will not be diverged and will stay within the "trusted region". PPO simplifies the optimization process by defining the probability ratio between the new policy and old policy and named as $r(\theta)$.

$$r(\theta) = \frac{\pi_\theta}{\pi_{\theta\ old}} = \frac{new\ policy}{old\ policy} \quad \ldots (2)$$

From TRPO, the ratio in eqn. 2 can be implemented as:

$$J(\theta)^{TPRO} = \widehat{E_t}\left[r(\theta)\ \hat{A}_{\theta\ old}\ (s,a)\right] \quad \ldots (3)$$

with

$$\hat{A}_{\theta\ old}\ (s,a) = Q(s,a) - V(s) \quad \ldots (4)$$

where, $Q(s,a)$ denotes as Q value and it is a result of function approximation between the input features and future discounted rewards values, while $V(s)$ is the value function or the goodness of state. The difference between these two functions is Q values that take account on the policy, action and state while $V(s)$ only considers state.

In contrast to TRPO, PPO imposes policy ratio, to stay within a range of small interval of 1 without adding Kullback-Leibler Divergence. The interval is defined as in between $1-\epsilon$ and $1+\epsilon$ where, ϵ is set to 0.2 in original PPO paper. The goal of PPO function is to obtain the minimum value between original value and clipped value.

$$J(\theta)^{CLIP} = \widehat{E_t}\, min[\, r(\theta)\ \hat{A}_{\theta\ old}\ (s,a)\ , clip(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{\theta\ old}\ (s,a)] \quad \ldots (5)$$

Next, in a complex LoRa network, there will be many nodes involved. In this paper, all nodes are required to react with each other. This can be done in RLlib where, the environment is based on gym environment. Each node will be represented by an agent and randomly distributed across a defined radius[5,6]. During the training, each agent will first configure its settings according to the action decided by PPO policy. As shown in Figure 3, each node will transmit a LoRa packet with size 50 bytes. In this

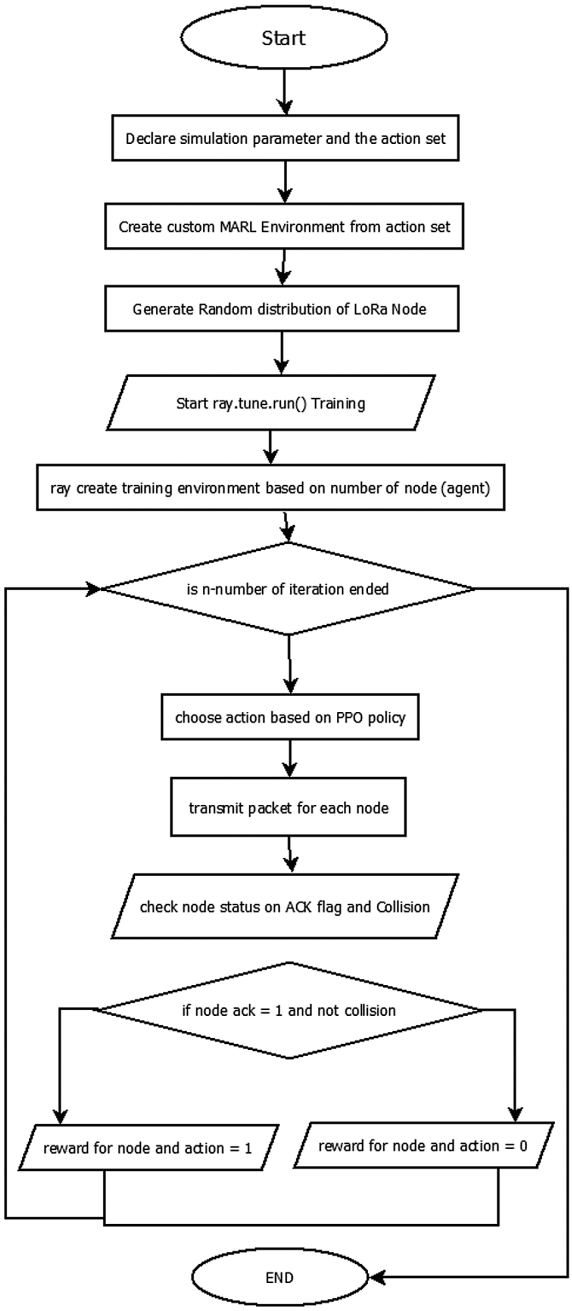

Fig. 3 — Process flow for proposed DRL in LoRa

work, the action space will be the spreading factor and each node will be assigned to a different spreading factor. Previous research has showed that with optimal configuration, the number of nodes with successful transmission may be significantly increased[12].

Note that it is important to determine a node with optimal configuration by two different values. In LoRa node, there will be two parameters to be considered for reward or "good action" which are the ACK flag and Collision. ACK flag is known as acknowledgement. The flag is to determine whether a sender node received a message that requires an acknowledgement.

In simpler term, it will check whether a packet has been sent[13]. If the flag is set to 1, it is a successful transmission and vice versa. For collision, LoRa packet will collide if two or more data are sent at same time. Collision of packet can be avoided with varying configuration in LoRa[13]. Any node with specific Spreading Factor with packet transmitted with ACK flag equal to 1 and have no collision is considered "good" action. Each agent will undergo this packet transmission and evaluate its reward. The training process will run until the defined iteration end.

Results and Discussion

As depicted in Figure 4, a total of 100 nodes were well distributed across 4.5 km radius at oil rigs deployment. The setup used for this research work is merely a NVIDIA Geforce MX150 with 2GB VRAM. This has also showed that with the help of CUDA devices, even a low specification GPU may help accelerate the training process.

The environment has run training with the following parameters as given in Table 1. The final training results are visualized with the help of Tensor

Table 1 — Parameters and constants used for training

Parameters	Values
Area	Disc of radius 4.5 km
Spreading factors	7, 8, 9, 10, 11, 12
Frequency	868100 Hz
Transmission power	14 dB
Number of nodes/agents	100

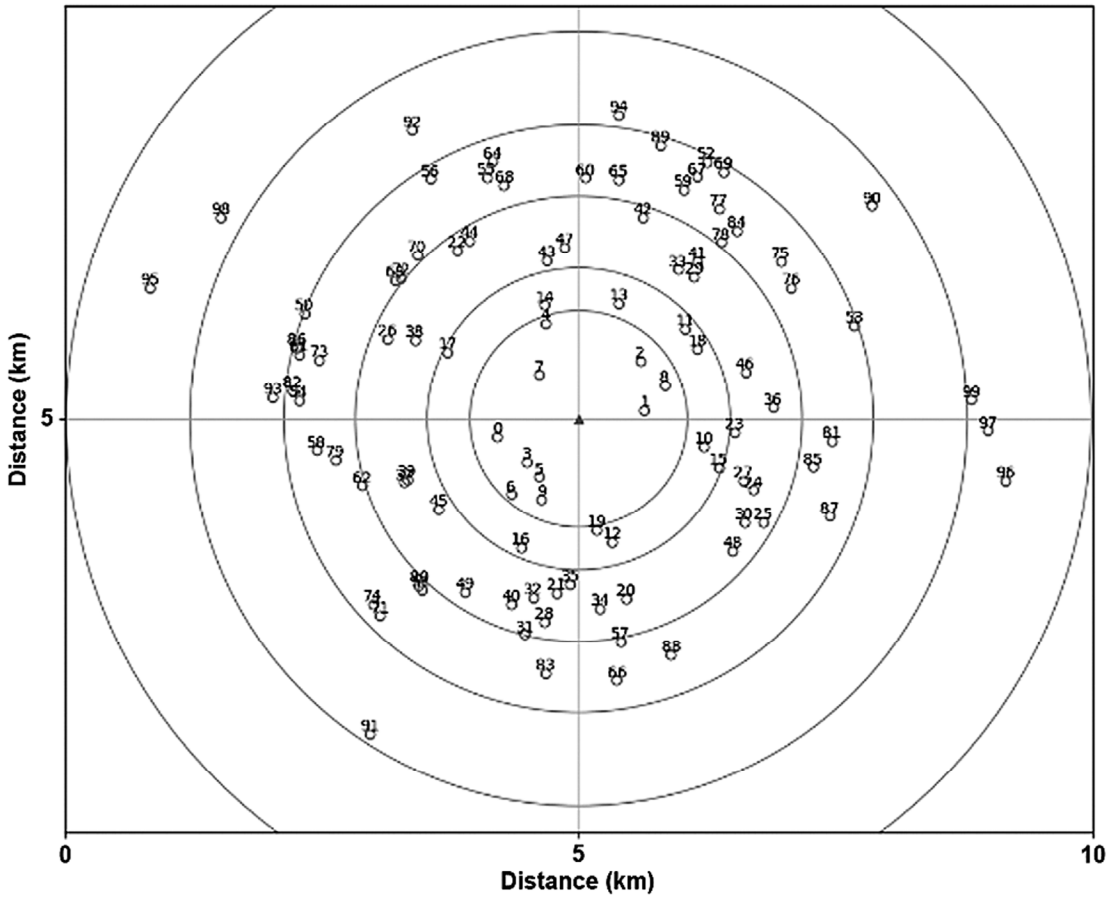

Fig. 4 — Distribution of LoRa nodes around wellheads in oil rigs deployment

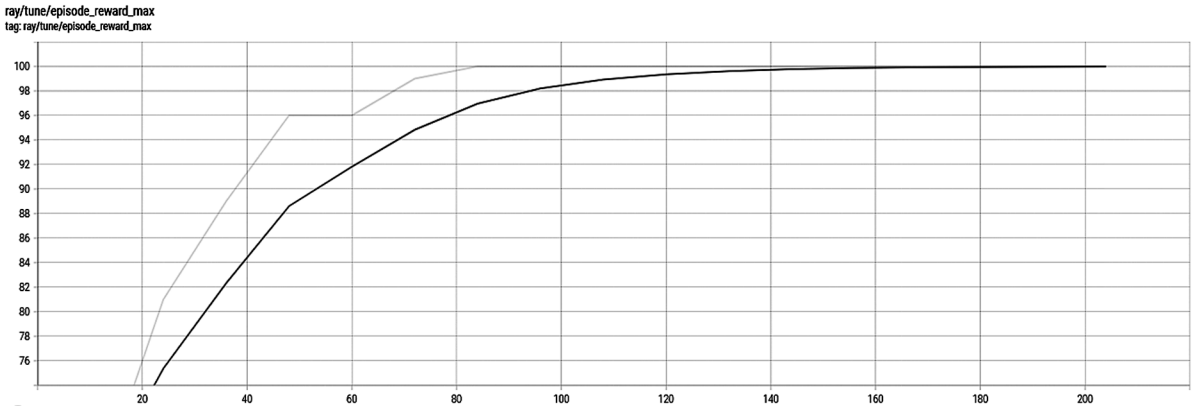

Fig. 5 — Iteration Process of DRL

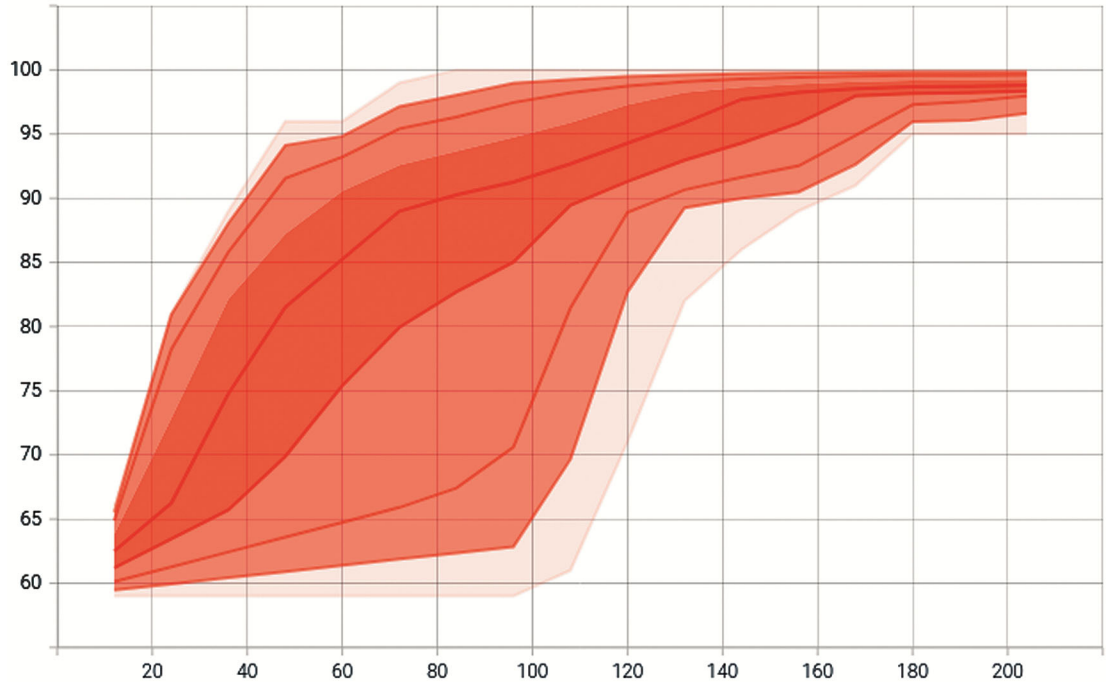

Fig. 6 — The total reward for each agent against timestep

Board. As shown in Figure 5, a total of 20 iterations have been run for the training process and it took about 45 minutes. The reward against timestep graph is plotted and shown (Fig. 5).

As there are 100 agents or nodes being simulated during the training process, the total reward of 100 should be observed. After over 140 timestep, the graph starts to settle and reach the desired performance which is 100. Figure 6 shows the reward for each agent against timestep. As observed from the graph, majority of the agents was able to identify the optimal spreading factor and able to reach 100, the higher gradient. A number of episodes can only reach about 60 for a period between 20 to 100 episodes, however, after that point, PPO policy was able to determine the optimal configuration and it increased to 90 in only 20 steps difference which is 100 to 120.

Conclusion

A new idea of PPO policy and Multi Agent Reinforcement Learning for task allocation in LoRa Network is investigated in this paper. It is implemented for deployment of sensors nodes in the application of gas wellhead monitoring service. With the use of CUDA device, the model was able to be trained in less than an hour and provided a promising result. The proposed method is best used when all the LoRa nodes are fixed with parameters such as location. This may be further improved in future when the location of LoRa nodes can be randomized.

Furthermore, the proposed method may be used in other 5G wireless technology such as SigFox. The proposed method will be implemented in other protocols as Industry 4.0 is introducing more protocols in this data driven era. Many of these protocols will be used at many places where it may be driven by battery. The proposed method will ensure that the device can operate for a suitable amount of time. For future work, transmission power and frequency may be included in the action space as it can also affect the performance and energy consumption in LoRa node[14,15].

Acknowledgements

This work was supported in part by the Universiti Teknologi Malaysia under Grant no. Q.K130000.2543.19H97.

Conflicts of Interest

The authors declare no competing interests.

Author Contributions

ZHI & BLLH: Conceptualization, methodology, validation, investigation, visualization, supervision, data curation, writing—original draft, writing-reviewing and editing. AE: Formal analysis, methodology, validation, investigation, visualization, data curation, writing—original draft, writing-reviewing and editing.

References

1 Qiao W & Wang H, Analysis of the wellhead growth in HPHT gas wells considering the multiple annuli pressure during production, *J Nat Gas Sci Eng*, 50 (2018) 43-54.

2 IndustriMigas, Wellhead Valve Operating Precautions, Industrimigas.com, 2021 [Online]. Available at: https://www.industrimigas.com/2013/06/wellhead-valve-operating-precautions.html

3 Sutton R, McAllester D, Singh S & Mansour Y, Policy gradient methods for reinforcement learning with function approximation, *NIPS'99: Proceedings of the 12th International Conference on Neural Information Processing Systems*, November 1999, pp. 1057–1063.

4 Khalil E A, Ozdemir S & Attea B A, A New Task Allocation Protocol for Extending Stability and Operational Periods in Internet of Things, *IEEE Internet Things J*, 6 (4) (2019) 7225-7231.

5 Ta D-T, Khawam K, Lahoud S, Adjih C & Martin S, LoRa-MAB: Toward an Intelligent Resource Allocation Approach for LoRaWAN, *IEEE Glob Commun Conf (GLOBECOM)*, 2019, pp. 1-6. doi: 10.1109/GLOBECOM 38437.2019.9013345.

6 Ta D-T, Khawam K, Lahoud S, Adjih C & Martin S, LoRa-MAB: A Flexible Simulator for Decentralized Learning Resource Allocation in IoT Networks, *12th IFIP Wirel Mob Netw Conf (WMNC)*, 2019, pp. 55-62. doi: 10.23919/WMNC.2019.8881393

7 Bor M C, Roedig U, Voigt T & Alonso J M, Do LoRa Low-Power Wide-Area Networks Scale? *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, *MSWiM '16*, 2016, pp. 59–67.

8 Park G, Lee W & Joe I, Network resource optimization with reinforcement learning for low power wide area networks, *J Wirel Comm Netw*, 176 (2020). https://doi.org/10.1186/s13638-020-01783-5

9 Zhang Q, Lin M, Yang L, Chen Z & Li P, Energy-Efficient Scheduling for Real Time Systems Based on Deep Q-Learning Model, *IEEE Trans Sustain Comput*, 4 (1) (2019) 132-141.

10 Ben Noureddine D, Gharbi A & Ben Ahmed S, Multi-agent Deep Reinforcement Learning for Task Allocation in Dynamic Environment, *Proceedings of the 12th International Conference on Software Technologies*, 2017, pp. 17-26.

11 Schulman J, Levine S, Moritz P, Jordan M I & Abbeel P, Trust region policy optimization, 2015. Available online at: arXiv preprint arXiv:1502.05477

12 Zorbas D, Papadopoulos G, Maille P, Montavont N & Douligeris C, Improving LoRa Network Capacity Using Multiple Spreading Factor Configurations, In: *25th Int Conf Telecommun (ICT)*, 2018, pp. 516-520.

13 Chinchilla-Romero N, Navarro-Ortiz J, Muñoz P & Ameigeiras P, Collision Avoidance Resource Allocation for LoRaWAN, *Sensors*, 21 (4) (2021) p. 1218.

14 Bor M & Roedig U, LoRa Transmission Parameter Selection, In: *13th Int Conf Distrib Comput Sens Syst (DCOSS)*, 2017, pp. 27-34.

15 Petäjäjärvi J, Mikhaylov K, Pettissalo M, Janhunen J & Iinatti J, Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage, *Int J Distrib Sens Netw*, 13 (3) (2017) 1-16.