Journal of Information Systems Engineering and Management
2025, 10(17s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Federated Reinforcement Learning (FRL) Framework for Mobile Data Collectors in IoT Applications

[1*]Mario Infant Raj, [2*]Dr. K. Kamali, [3*]Dr. R. Manikandan,
[1]Research Scholar , Annamalai University. Chidambaram, Tamilnadu. (corresponding author)
marioinfantrajphd@gmail.com
[2]Assistant Professor, Dept of Computer science,Annamalai University. Chidambaram,, Tamilnadu.
Kamaliaucse2006@gmail.com
[3]Associate Professor, Dept of Computer science,Annamalai University. Chidambaram, Tamilnadu.
rmkmanikandan1111@gmail.com

ARTICLE INFO

Received: 05 Dec 2024

Revised: 24 Jan 2025

Accepted: 08 Feb 2025

ABSTRACT

Introduction: During Mobile Data Collector (MDC) based data aggregation in Internet of Things (IoT) applications, latency and energy consumption increases, due to incorrect visiting schedules of MDCs.

Objectives: To determine data collection schedules of MDC, based on IoT sensor's data generation rate.

Methods: In this paper, a Federated Reinforcement Learning (FRL) framework is proposed for MDCs in IoT applications. In this technique, the data generation rate of IoT nodes are learned by applying FRL framework from which the nodes are classified as Emergency, Normal, Less Frequent and Rare. Then depending on the category of the nodes, the visiting schedule and stopping time of MDCs are determined.

Results: The proposed FRM framework is implemented in NS2 and it has been shown the proposed MDC-FRL framework reduces the data collection latency and energy consumption and improves the accuracy.

Keywords: Internet of Things (IoT), Mobile Data Collector (MDC), Federated Reinforcement Learning (FRL), Data generation patterns.

Introduction

IoT has emerged in recent years as a result of substantial advances in communication among devices[1]. In IoT, a Wireless Sensor Network (WSN) is a common technology that provides multi-user access through a multi-app platform. Lower energy costs, better use of natural resources, safer cities, and a healthier environment may all be achieved via the usage of IoT solutions [2].Artificial Intelligence (AI) and IoT have gained the attention of academics because of their fast increase [3].

IoT networks may save a lot of power by using mobile data collecting. When using a MDC, the biggest problem is determining and arranging the MDC's course to gather data from nodes. Static techniques of obtaining mobile data only identify a solution to a problem with predetermined variables[4][5].

Data loss may be reduced even at high data rates and over a wide range of network sizes by ensuring that the data reception rate increases [6]. It has become more difficult to handle and analyse large data repositories generated by recent advances in data collecting devices for smart cities [8][9]. The collecting of data in the many IoT ecosystems has become a complex process. In MDC based data collection techniques, if a cluster has no data to send in a specific time slot, the MDC has to omit that cluster or to reduce the visiting times [10][11].

1.1 Objectives

The main objectives of this paper are stated as below:

- To generate data collection schedules for nodes based on their data generation pattern
- To determine a visiting schedule for MDC based on the nodes data generation pattern

- To reduces the latency and energy consumption during data collection

Related Works

Abdulsalam et al [12] have explored the topic of enhancing the WSN lifetime using a cluster-based data gathering method. They suggested a fresh approach to solving the issue. In this work, MDCs serve as CHs in an algorithm for cluster-based aggregation.

Li et al. [13] have introduced a new information collection architecture that allows the UAV to simultaneously gather sensed data, if the devices come into the communication range of UAV.

To examine the coverage efficiency of MDC with IoT devicess, Ma et al. [14] have suggested a analytical framework which has a mobility model. They've derived exact formulas for the common contact time (CT) and inter-contact time (ICT) from sensors and MDC.

A distributed MDC sojourn point nomination technique that Mazumdar et al. [15] suggested greatly reduces the WSN's message complexity and energy usage. The chosen Data Collection Points (DCPs) are communicated to the BS, which then applies a modified ACO algorithm to create the MDC traversal path.

Methodology

3.1 Contributions

The system model consists of IoT sensors grouped into clusters, a gateway and MDC. Initially, IoT devices and sensors are arranged into clusters depending on their geographical information. A gateway collects the data from each cluster, aggregates it and transmits to the MDC.

This work defines a FRL framework for MDC in IoT applications. In this work, the data generation rates of IoT sensors are learned using FRL. In FRL, each IoT device trains its local model and transmits it to the gateway. The model parameters are aggregated by the gateway into a global model which are transmitted to the IoT devices again. Then depending on the category of nodes, the number of slots, slot duration and visiting schedules of MDC, are determined.

3. Federated Learning (FL)

In FL, the IoT users and the gateway share a global model for training and the original dataset is stored at the users devices. Each user trains his local ML model using his dataset. Then, they upload their local model parameters to the gateway, which then aggregates them to generate a global model.

A normal FL model has a FL server and a set S of contending users. Each user U_i has a dataset DS_i which is trained by a local model ML_i . The parameters P_i of each model ML_i are transmitted to S, which aggregates all P_i to obtain a global model MG based on an aggregation policy.

The main steps performed in the training process of FL include:

1. **Initializing and sistributing the model:** In first iteration, S distributes the initial global model MG_1 and all task parameters to the contending users
2. **Training and Updating the local model:** In each iteration t, each user U_i updates its parameters P_i of ML_i based on the global model MG_t . Then U_i uploads its updated P_i to S.
3. **Aggregating local models and updating the global model:** In each iteration t, S aggregates all the ML_i received from U_i.
4. **Distributing the Update global model:** Finally, S broadcasts its updated global model MG_{t+1} to the users, in next iteration (t+1).

These steps are repeated until the local model converges or the required accuracy is achieved.

3.2 Reinforcement Learning (RL)

Let x = state space and y = action space. Reinforcement Learning (RL) or Q-learning is an algorithm for the environment, which is limited by small state spaces .Deep Q-Learning technique applies neural networks to approximate the optimal Q-functionQ_π (x, y).

Experience replay method allows to maintain the agent's encounters at each right time slot n within the information set.

At time slot n, the agent's encounters can be denoted as follows:

$AG_n = (x_n, y_n, r_{n+1}, x_{n+1})$ (1)

The policy system w approximates the insurance policy that is optimal locating the option Q function. The current state x_n is accepted. In addition, The evaluation of the value $A(x_n, y_n, w)$ will be determined. To boost stability of learning, a target system w' is employed. The goal community loads are frozen utilizing the policy that is initial and so are updated periodically.

The next state x_{n+1} is accepted by this, thereby making the Q-value outputs (x_{n+1}, y_{n+1}, w'). The Q-values are updated to fulfil the loss function, as given by

$LF(w) = E[TG_n - Q(x_n, y_n))^2]$ (2)

where TG_n is the target value, given by

$TG_n = w_n + c^{n-1}\max Q(x_{n+1}, y_{n+1}, w')$ (3)

where Q-value for the state x_{n+1} is given to the target network w' , for accurate training.

3.3 FRL Process

FRL is a combination of FL and RL techniques [13]. FRL techniques can use observations from many environments together for RL. Hence we utilize FRL to train the data generation patterns of IoT devices and classify each cluster into Emergency, Normal, Less Frequent and Rare.

The visiting schedules and stopping times of MDCs are determined based on the category of sensors, which are derived from the type of IoT sensor and data generation rate D_{rate} . Tab1e shows the type of sensor, D_{rate} and the corresponding category.

Type of sensor	$\mathbf{D_{rate}}$	**Classified category**
1	----	Emergency
2	$D_{rate} > D_{max}$	Normal
2	$D_{min} > D_{rate} <= D_{max}$	Normal
2	$D_{rate} <= D_{min}$	Normal
3	$D_{rate} > D_{max}$	Less Frequent
3	$D_{min} > D_{rate} <= D_{max}$	Less Frequent
3	$D_{rate} <= D_{min}$	Rare

Table 1 Categories of Assignment of scheduling priorities

Let U_j {j=1, 2, ..., K} is the set of users to collectively apply FRL model for uploading an IoT task. Here, the BS acts as an aggregation server for each IoT device.

The FRL algorithm is then presented below:

1. In data learning process, the type and D_{rate} of each U_j is learned using FRL.
2. Then its category C_j is determined.
3. Each U_j forms their dataset $DS_j=\{T_t, type, D_{rate}, C_j\}$, where T_t is the time at iteration t.
4. U_j and BS share a global model for training
5. Each U_j perform training using the shared model on their dataset DS_j
6. After training, U_j obtains its local model ML_j

7. U_j upload its ML_j to BS
8. BS aggregates all ML_j , to derive the global model MG
9. BS again shares the global model MG to all U_j.
10. Once the training is completed, the visiting schedules V_j of MDC is determined based on C_j

By applying the distributed training process at the IoT devices, the gateway and BS can enhance the training results devoid of affecting the privacy of user data.

Experimental Results

4.1 Experimental Parameters

The FRL-MDC framework is simulated in NS2 and the FRL model is represented in Python. The experimental settings consists of 100 nodes deployed in a 50mX50m region. The IEE 802.15.4 standard is applied for IoT networks. There are 6 exponential and constant bit rate traffic flows with a rate of 50-250Kbps.

4.2 Comparison Results

The performance of FRL-MDC is compared with the data collection method based on Improved Dragonfly Algorithm (DC-IDA) [16] in terms of packet delivery ratio, packet drop, and residual energy by varying the traffic rate from 50 to 250 Kbps.

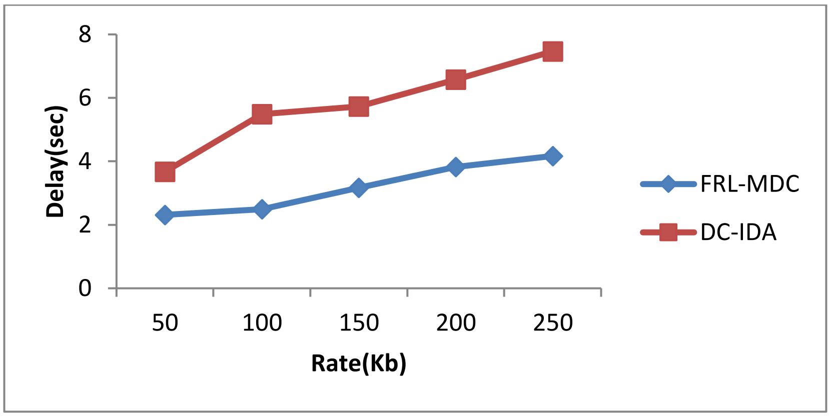

Figure 1 Delay for varying Rate

Figure 1 shows the delay values for different rate values. From the figure it can be see that the delay of FRL-MDC is 45% lesser than DC-IDA.

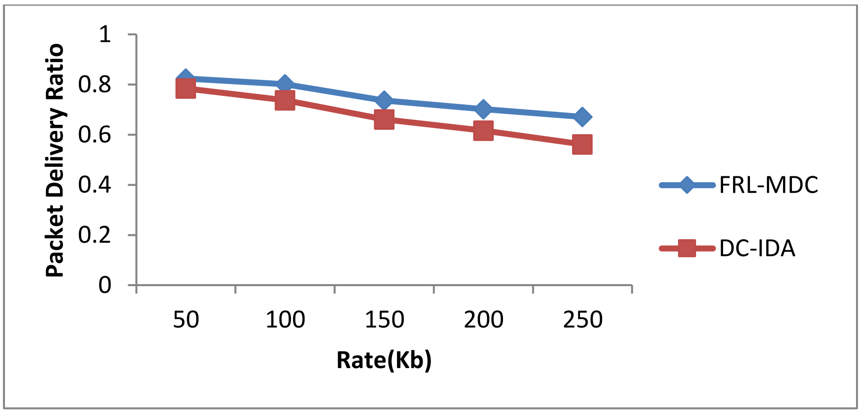

Figure 2 Packet Delivery Ratio for varying Range

Figure 2 shows the packet delivery ratio values for different rate values. It can be seen that packet delivery ratio of FRL-MDC is 10% higher than DC-IDA.

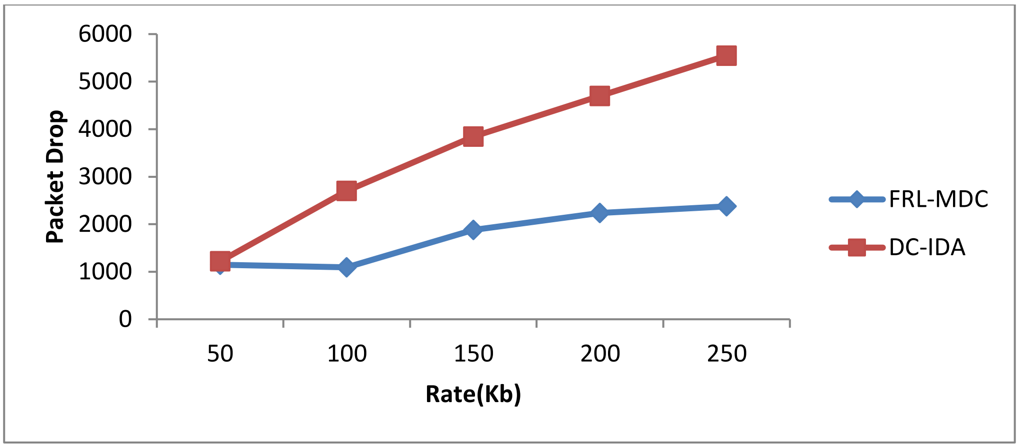

Figure 3 Packet Drop for varying Range

Figure 3 shows the packet drop values for different rate values. It can be seen that packet drop of FRL-MDC is 45% lesser than DC-IDA.

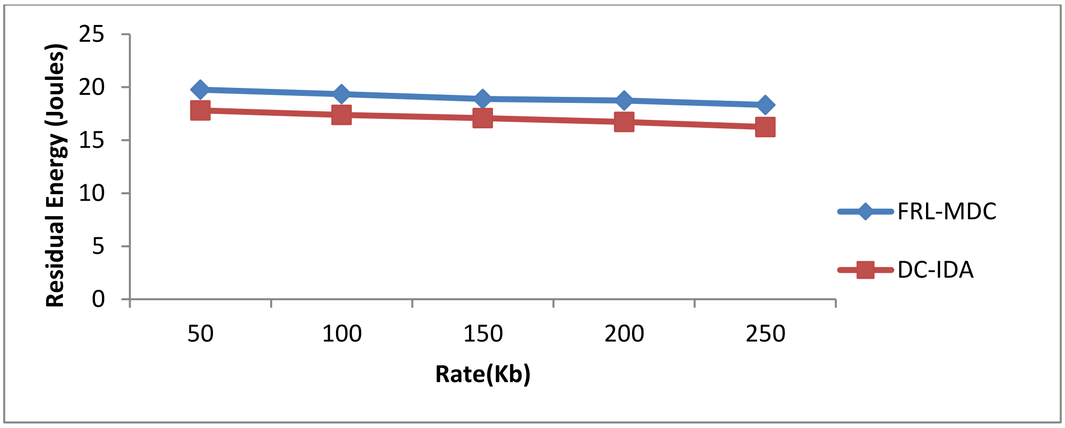

Figure 4 Residual Energy for varying Range

Figure 4 shows the average residual energy obtained for various data rates. It can be seen that FRL has 10% higher residual energy, when compared to DC-IDA.

4.3 Classification Results

The classification results of FRL are compared against the existing FL and CNN models. Figure 5 shows the results of accuracy, sensitivity and F1-score values for these models.

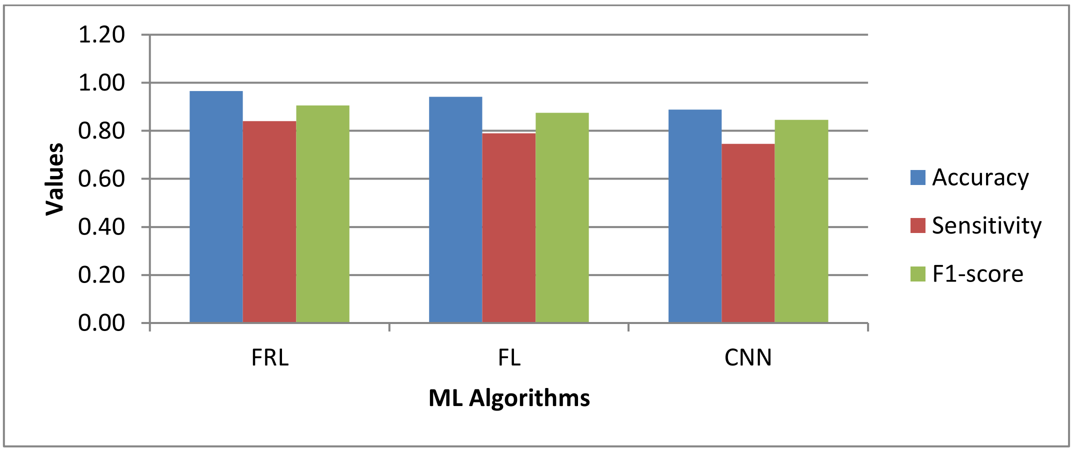

Figure 5 Comparison of Accuracy, sensitivity and F1-score

Figure 5 shows that FRL has 2% higher accuracy than FL and 8% higher accuracy than CNN. Similarly, FRL has 6$ higher sensitivity than FL and 11% higher sensitivity than CNN. The F1-score of FRL is 3% more than FL and 7% more than CNN.

Conclusion

In this paper, a Federated Reinforcement Learning (FRL) framework is proposed for MDCs in IoT applications. In this framework, the data generation rates of sensors are learned using FRL. The parameters of the aggregated global model are delivered to the IoT sensors again. Then depending on the category of sensors, visiting schedules of MDCs are determined. The proposed FRL-MDC framework has been implemented in NS2 and results have shown that FRL-MDC reduces the data collection delay and packet drop and improves the residual energy and packet delivery ratio.

References

[1] Andreas P. Plageras, Kostas E. Psannis, Christos Stergiou, Haoxiang Wang and B.B. Gupta, "Efficient IoT-based sensor BIG Data collection-processing and analysis in Smart Buildings", Future Generation Computer System, 2017.

[2] Walid Osamy, Ahmed M. Khedr, Ahmed A. El-Sawy , Ahmed Salim and Dilna Vijayan, "IPDCA: Intelligent Proficient Data Collection Approach for IoT-Enabled Wireless Sensor Networks in Smart Environments", MDPI, Electronics, 2021.

[3] Xuemei Xiang, Wei Liu, Tian Wang, Mande Xie, Xiong Li, Houbing Song, Anfeng Liu and Guoping Zhang, "Delay and energy-efficient data collection scheme-based matrix filling theory for dynamic traffic IoT", EURASIP Journal on Wireless Communications and Networking, 2019.

[4] Cosmas Ifeanyi Nwakanma, Fabliha Bushra Islam, Mareska Pratiwi Maharani, Dong-Seong Kim and Jae-Min Lee, "IoT-Based Vibra tion Sensor Data Collection and Emergency Detection Classification using Long Short Term Memory (LSTM)", IEEE, ICAIIC, 2021.

[5] Sana Benhamaid, Hicham Lakhlef, Abdelmadjid Bouabdallah, "Towards Energy Efficient Mobile Data Collection In Cluster-based IoT Networks",IEEE, 2021.

[6] R.Radhika and K.Kulothungan, "Improved Data Dissemination using Intelligent Stimulus Mechanism in IoT NEtwork", Advances in Engineering Research (AER), volume 142, 2018.

[7] Aya H. Allam, Mohamed Taha and Hala H. Zayed, "Enhanced Zone-Based Energy Aware Data Collection Protocol for WSNs (E-ZEAL)", Journal of King Saud University – Computer and Information Sciences 34, 2022.

[8] Ghyzlane CHERRADI, Adil El Bouziri and Azedine Boulmakoul, "Smart Data Collection Based on IoT Protocols", Proceedings of the Third International Conference on Advanced Informatics for Computing Research, 2019.

[9] Khan Muhammad, Jaime Lloret, and Sung Wook Baik, "Intelligent and Energy-Efficient Data Prioritization in Green Smart Cities: Current Challenges and Future Directions", IEEE Communications Magazine, 2019.

[10] Yi Huang, Shah Nazir , Xinqiang Ma , Shiming Kong, and Youyuan Liu," Acquiring Data Traffic for Sustainable IoT and Smart Devices Using Machine Learning Algorithm", Security and Communication Networks, Hindawi, Volume 2021, Article ID 1852466, 11 pages, https://doi.org/10.1155/2021/1852466

[11] Nataly Zhukova, Aung Myo Thaw, Man Tianxing, Mustafin Nikolay," IoT Data Collection based on Social Network Models", Proceedings of The 26th Conference Of Fruct Association.

[12] Hanady M. Abdulsalam∗, Bader A. Ali, EmanAlRoumi, "Usage of mobile elements in internet of things environment for data aggregation in wireless sensor networks", Computers and Electrical Engineering 000 (2017) 1–19.

[13] Yuchen Li, Weifa Liang, WenzhengXu, ZichuanXu, XiaohuaJia, YinlongXu, and HaibinKan, "Data Collection Maximization in IoT-Sensor Networks Via an Energy-Constrained UAV", IEEE Transactions On Mobile Computing, 2020.

[14] Yajun Ma, Hongguang Sun, Shuqin Li, Xijun Wang, Tony Q. S. Quek, "Performance Analysis of IoT networks with MobileData Collectors", IEEE 2021.

[15] NabajyotiMazumdar, Amitava Nag, Jyoti Prakash Singh, "Cache-aware mobile data collection schedule for IoT enabled multi-rate datagenerator wireless sensor network", Sustainable Computing: Informatics and Systems 31 (2021) 100583.

[16] Yinggao Yue , Dongwan Lu, Yong Zhang , Minghai Xu, Zhongyi Hu,Bo Li, Shuxin Wang and Haihua Ding,"A Data Collection Method for Mobile Wireless Sensor Networks Based on Improved Dragonfly Algorithm", Computational Intelligence and Neuroscience, Hindawi, Volume 2022, https://doi.org/10.1155/2022/4735687