Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Enhancing Immutable IoT Data Collection through Distributed Reinforcement Learning Framework

M. Shakila, A. Rama, Anitha Govindaram, A. Jegatheesan, Karpagam V, S. Christy

Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, India
Email: shakilam1006.sse@saveetha.com

The Intelligent Mobile Data Collection (IMDC) framework is presented in this study for use in IoT-based sensor networks. Clusters of nearby Internet of Things (IoT) devices and sensors are organised using this approach. A gateway node acts as the coordinator of each cluster and is responsible for gathering information from its members, processing it, and then sending it on to the main data collector (MDC). In order to examine trends in data creation, the framework uses a labelling method called FRL (Very Rare Labels). Time intervals, packet numbers, and types are some of the parameters that can be used to classify clusters using these labels. Local models that take into account states, actions, and rewards are developed by individual Internet of Things (IoT) sensors or devices within the FRL framework using reinforcement learning (RL) approaches. After collecting data from various regional models, the gateway compiles it into a global model and returns it to the IoT-Internet of Things sensors. In addition, the framework uses cluster categories to find the MDC's sleep length, visitation time, and Time Division Multiple Access (TDMA) slots, among other factors. This framework's efficacy in lowering latency and energy usage during data collecting and enhancing accuracy is demonstrated by its implementation using NS2.

Keywords: Federated, heterogeneity, Q-learning, Reinforcement, Aggregation.

1. Introduction

Modern developments in computing and networking have given rise to the phrase "Internet of Things" (IoT) in the last several years [1]. In an IoT system, every item that can be linked to the web over a wired or wireless network is considered a part of it. People, machines, or a mix of the two can make use of these systems. Internet of Things (IoT) systems often make use of Wireless Sensor Networks (WSNs), a flexible technology that can accommodate a wide range of applications and users. Energy savings, better resource optimisation, increased urban safety, and improved environmental sustainability are all possible outcomes of implementing Internet of Things (IoT) solutions.

As the Internet moves away from its current state of centralization and towards decentralisation, new computational models such as , edge computing, cloud computing and fog computing are being proposed. The exponential development of AI and the IoT has piqued the interest of academics [3].

An essential feature of the Internet of Things (IoT) is its capacity to gather and process data. Mobile data collecting made possible by IoT networks allows for substantial energy savings. The main problem with a mobile sink system is figuring out how to get data from nodes and getting everyone on the same page. Issues with preset variables are the only focus of traditional mobile data collection methods [4][5].

Improving the efficiency of data transmission can lead to better network performance and lower communication expenses. Intermittent power sources necessitate protocols and connections that can manage delays. Data throughput over radio connections can be improved by implementing effective scheduling algorithms for data transfer. Even in cases by high statistics rates and varied network dimensions, data loss can be minimised by optimising data reception rates. The presence of different devices is responsible for the decrease in system performance in the current setup [6].

The management, retrieval, exploration, and analysis of massive data sets produced by contemporary data gathering equipment for smart cities is becoming more and more of a challenge. Among these data sources are surveillance networks and consumer videos [8][9]. Data collecting has grown increasingly intricate in many IoT ecosystems. The ever-shifting placement of devices presents these systems with their greatest obstacle [10][11].

1.1 Problem Identification and Objectives

Efficient data sharing or dissemination systems are crucial for Pervasive Sensing (PS) to guarantee cost-effective and efficient data distribution across a comprehensive model. Optimising data collection requires careful consideration of nodes' latency, energy usage, and storage capacity. Furthermore, a dependable system with the ability to manage errors is essential in order to guarantee precise data collection.

It is not feasible to allocate a fixed time slot to apiece node in the network because the rate at which packets are generated by different nodes varies. Reallocating slots for each node before every scheduling round may not be feasible in real-time scenarios due to the varying quantities and sources of dynamically generated packets [3].

When using mobile data collectors (MDCs) for data collection, if a cluster does not have any data to transmit during a particular time slot, the MDC must either skip that cluster or decrease the amount of time spent visiting it [5].

The primary goals of this study, derived from the identified issues, can be summarised as follows:

- Create TDMA schedules for nodes according to their data generation patterns.
- Develop a sleep scheduling policy for nodes based on their data generation patterns.
- Establish a visitation schedule for MDCs by analysing the data generation patterns of the nodes.
- Minimise the delay and decrease the amount of energy used when gathering data.

2. Related Works

One approach to the problem of how to distribute civic data efficiently in a smart city setting is the IPDCA, which was proposed by Walid-Osamy et al. [2]. In order to send data from several Access Points (APs) to a central Base Station (BS), the system employs data collectors, also called D-collection trucks. The discrete optimisation problem is solved by routing D-collectors using a modified Bat algorithm. Optimising resource utilisation is the goal of using a multi-objective suitability purpose to pick D-collectors in smart city scenarios. This function takes into account aspects including storage capacity, travel distance during collection, and the quantity of D-collectors. The results of the simulation show that the suggested approach works.

With the goal of reducing latency and energy ingesting, Xiang et al. [3] presented a DEEDC system-based matrix filling theory for efficient data assembly in WSN-Wireless Sensor Networks Using a clustering method, the DEEDC scheme determines transmission slots for each cluster while excluding nodes that generate data. Eliminating the need to allocate time periods to specific nodes and preventing data duplication, this strategy streamlines the process of gathering random data in a network.

Cos-mas Ifeanyi Nwakanma et al. [4] measured vibrations in actual time using the G-Link 200 sensor and transmitted the data to an online gateway using Long Short-Term Memory (LSTM). Being able to sort things into common and uncommon categories is a lifesaver in stressful times. In order to safeguard the privacy of its residents, this feature is commonly seen in more current models of smart homes. The sensor may also identify excessive or odd vibrations in smart manufacturing settings, eliminating the need for intrusive video monitoring. The study offers valuable insights for future analysis of sensor data.

Sana Benhamaid et al. [5] proposed using Deep Q-learning with experience replay to determine the path of a mobile node in cluster-based systems and mobile data collection scenarios. This method adapts to significant changes in the environment, such as changes in the amount of data that can be captured, by exercise a neural network (NN) to learn about the atmosphere and then selecting an energy efficient route for the portable device.

A modification to ZEAL was suggested by Aya H. Allam et al. [7] to improve the effectiveness of data transmission and energy usage in WSNs. An expanded set of subsink nodes as well as a mobile-sink node are made available by Improved ZEAL (E-ZEAL) through the utilization of K-means clustering. Experimental results obtained by the ns-3 simulator show that E-ZEAL outperforms ZEAL with respect to data collection speed, thanks to a 30% reduction in the number of hops and remoteness and an equal increase in the lifetime of the network.

3. Proposed Solution

3.1 Overview

Designed with IoT sensor networks in mind, this article lays forth a framework for intelligent mobile data collection. The framework uses the Frequent, Less Frequent, and Rare Labels (FRL) to examine data creation patterns, which include time intervals, packet amounts, and packet kinds. Every Internet of Things (IoT) sensor and device is trained individually using reinforcement learning (RL) methods under the FRL paradigm. This training incorporates states, actions, and rewards. When these Internet of Things (IoT) sensors collect data, they send the characteristics of their local models to the gateway, which then combines them into a global model. The Internet of Things sensors are then updated with the values from this unified global model. The architecture also uses clustering categories to determine MDC visiting times, sleep length, and Time Division Multiple Access (TDMA) slots.

3.2 System Model

In this article, we offer an IoT-specific Intelligent Mobile Data Collection framework for sensor networks. Temporal, Quantitative, and Rare (FRL) Labels are used to examine data generating patterns such as packet amounts, kinds, and frequency intervals. Using RL techniques that incorporate states, actions, and incentives, the FRL methodology trains separate models for each Internet of Things (IoT) sensor or device. After that, the sensors send the parameters of their local models to the entry, where they can be combined into a universal model. The Internet of Things sensors are subsequently given the parameters from this combined global model. Mobile Data Collectors (MDCs) sleep for how long, how often they visit, and what Time Division Multiple Access (TDMA) slots they are all determined by the framework's clustering categories.

3.3 Basics of FL

Using Federated Learning (FL) algorithms, a group of people use IoT campaigns, such as smartphone, tablet, or laptop to do IoT activities together. With FL, IoT systems at the network's periphery can get extensive intelligence, making them an essential component of the next-gen IoT networks. The reason behind this is that training AI and ML with data collected from scattered IoT devices is simply not feasible with a single Base Station (BS). The Internet of Things (IoT) and its users can work together to train a common global model using Federated Learning (FL), which preserves the raw data on the users' devices. As part of FL, every IoT user contributes their own dataset to train a shared local ML model. When an Internet of Things device has completed local training, it can send any revisions to its model to the base station (BS). After receiving these changes, the base station compiles them into one global model.

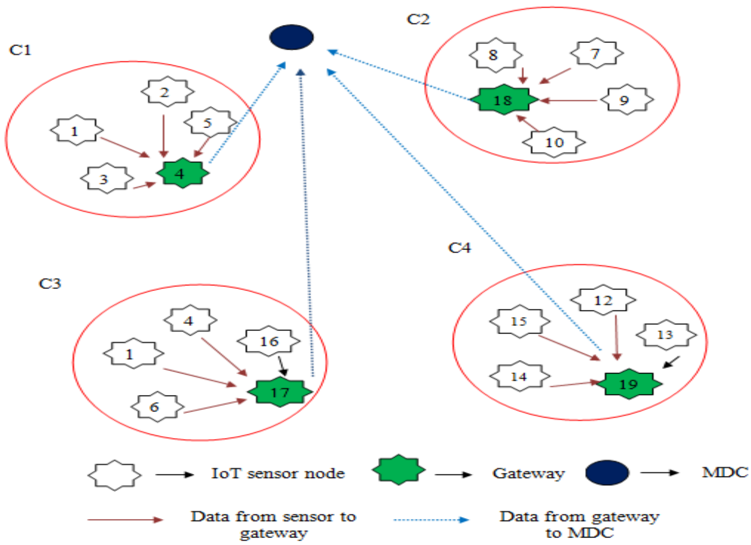

Figure 1 System Model

Typically, in a Federated Learning (FL) system, there is one FL server $C(|C| \geq 1)$, S, and several clients that work with it. Each client has their own private dataset, denoted as $m = U_{c \in C} m_c$. Each client c updates the FL server S with its local parameters after training a model mc with its own dataset dc. By following a predefined aggregation rule, S gathers all the local models and gets the global model MG. Note that this is dissimilar from the conventional cloud-centric training method, wherein all client data is centralized and processed to train the model MG $D = U_{c \in C} d_c$..

Federated Learning (FL) training follows a three-step process, as seen in Figure 1:

Step 1 (Task initialization and model distribution): S specifies the goal model, data requirements, and hyperparameters (such batch size) during round M_G^0. when choosing the exact training assignment. After that, it sends the initial configuration settings for the tasks and global model to every client that is involved.

Step 2 (Training and updating local model): During round t, every client c uses its local data to update its resident model M_G^t limits based on the global model m_t^c,. This is the second step, training and updating the local prototypical. Finding the optimal limits m_t^c that reduce the loss function is the goal $L(m_t^c)$. Then client c sends the efficient local parameters to client S.

Step 3 (Aggregating and updating global model): S collections all the received local models during round t to curtail the global loss function. This process is known as updating the global model. After that, S updates the universal model and sends it out to the clients for exercise in round $L(m_t^c)$ t+1, This progression continues until either the models converge or M_G^{t+1} the accuracy is up to scratch.

$$L(m_t^c) = \frac{1}{|C|}\sum_{C=1}^{|C|} L(m_t^c)$$

3.4 Deep Reinforcement -Learning (D R-L)

Reinforcement learning (R-L) is a mathematical outline that allows computation devices to learn from experience by interacting with the environment.

In the field of reinforcement learning, an agent makes decisions on which actions to take by following a predetermined set of rules called a policy. These actions are then carried out within the given environment. Afterwards, the agent monitors the changing condition of the environment and is given rewards based on that. The agent consistently updates its policy through iterative cycles in order to improve its action selections. The agent's behaviour is influenced by the rewards offered by the environment, and its objective is to acquire the most effective strategy that maximises the anticipated rewards. The learning process of the agent differs depending on the specific reinforcement learning approach utilised.

Deep reinforcement learning (DRL) expands upon the principles of reinforcement learning (RL) by incorporating deep neural networks. DRL utilises neural networks to train by using state and reward data obtained from actions, in order to determine the most optimal actions founded on the current state. In recent years, a multitude of DRL procedures have been introduced.

3.5 Federated Reinforcement Learning (FRL) Process

In order to improve reinforcement learning, FRL, which combines FL and RL methodologies [13], makes use of data from various settings. When compared to conventional DRLs, FRL performs better in situations where observations of the same environment are lacking. So, we use FRL to train the patterns of data creation of IoT devices and classify each cluster as either Very Rare, Less Frequent, Rare, or Frequent.

Let K stand for a set that represents a group of people using Internet of Things (IoT) devices to execute an IoT job using a FRL algorithm. This involves the following basic steps:In this setup, FRL is used to learn how to create data patterns, such as time intervals, packet amounts, and packet types.

- This instructional technique involves two individuals. The Data Client acts as the starting point, representing an IoT device.
- Base stations or access points contain Aggregation Servers.
- Internet of Things (IoT) users and the base station can collaborate on training a global model using FRL, with the raw data staying on the users' devices.
- Each Internet of Things user, represented as k, contributes their own dataset, Dk∈K, to the training of a shared model. Then there's the FL model that learned from the IoT expedient—the local prototypical wk.
- When the resident training is over, Internet of Things devices let the base station know by sending updates to their local models. The global model wG, a shared model, is formed by combining these updates.
- The combination server at the base station can enhance training recital and safeguard user data privacy through distributed data training on IoT devices.
- Finally, clusters are put into four groups:

(i) Most prevalent
(ii) Less prevalent
(iii) Infrequent
(iv) Extremely uncommon

3.5.1 FRL Algorithm

The system's FRL comprises the subsequent key stages.
i. All devices receive the initial global model from the gateway.
ii. Using local information like as states, actions, and incentives, each device learns its own model.
iii. States
To ensure that an agent carries out the most effective action for a given situation, the state must contain relevant information.
We created the initial value set to incorporate the cluster data.
iv. Actions
The actions involve displacing clusters in four cardinal directions, namely left, right, downward, upward, on the traffic map.
v. Rewards

- Appropriate rewards must be provided for an agent to learn well.
- A reward is granted when there is a reduction in traffic, and conversely, a negative reward is given when there is an increase in traffic.
- A reward is received if the traffic remains constant, the cluster remains in position, and it has fully utilised its network service capacity.
- If the capability is not utilised, then a negative reward is obtained.

1. The gateway receives the local model limits (W1,..., Wn) transmitted by the devices.

2. The gateway is responsible for integrating the model limits into the global model.

3. Before the global model is considered adequately trained, it is disseminated to the devices again with the limits of the combined model, WG. The process is then repeated as before.

4. Experimental Results

4.1 Mockup Parameters

The Intellectual Mobile Data Collection utilizing FRL (IMDC-FR-L) system can be applied in NS3 with the help of Python's FRL module. Table 1 displays the parameters used in the simulation.

Table 1 Simulation settings

Number of Node(s)	20 – 100
Size of the T-topology	50m * 50m
MAC- protocol	IEEE (802.15.4)
Traffic -Source	CBR and E(Exponential)
Traffic -Flows	0.6
Traffic -Rate	50-Kb
Initial- Energy	15 -Joules
Transmit -power	0.3 -watts
Receiving -power	0.3 -watts

4.2 Comparison Results

For the purpose of evaluating IMDC-FRL, we compare it to the Intellectual Proficient Data Collection Approach (IPDCA) [2]. One of the criteria utilized in the evaluation is the packet delivery ratio. Other metrics include computation cost, average residual energy, and packet drop. For this test, we shifted from using 20 nodes to using 100. Here you can view the outcomes of the comparison.

Table 2 Delivery Ratio output

Node(s)	IMDC-FR-L	IP-DCA
0-20	0.9595	0.9223
21-40	0.9536	0.9178
41-60	0.9448	0.9087
61-80	0.9427	0.9024
81-100	0.9414	0.8962

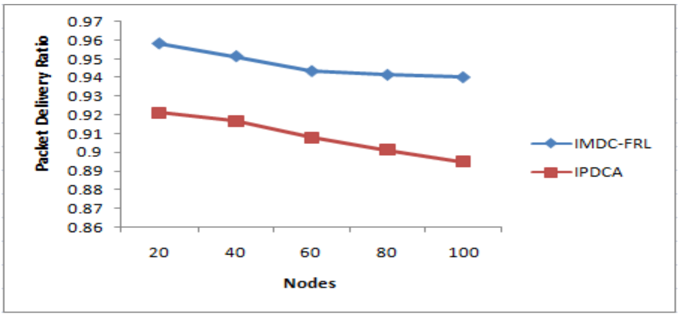

Figure 2 Packet Delivery ratio for varying nodes

The values of the packet distribution ratio for dissimilar numbers of nodes are shown in Figure 2. The figure shows that compared to IPDCA, IMDC-FRL has a packet delivery ratio that is 4% higher.

Table 3 Packet Drop output.

Node(s)	IMDC-FR-L	IP-DCA
0-20	3006	4582
21-40	5445	7253
41-60	6212	9465
61-80	6731	9866
81-100	7454	10497

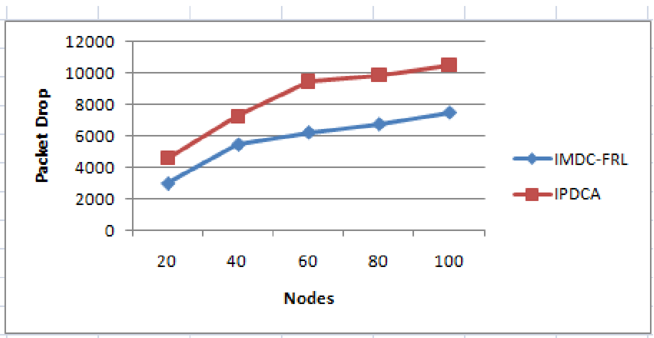

Figure 3 Packet Drop for varying nodes

For different numbers of nodes, Figure 3 shows the packet drop rates. It is evident from the figure that IMDC-FRL has a 31% lower packet drop rate compared to IP-DCA.

Table 4 Results of Computational Cost

Node(s)	IMDC-FR-L (Kb)	IP-DCA (Kb)
0-20	37.8	42.7
21-40	42.6	45.2

41-60	47.8	57.0
61-80	48.7	63
81-100	64.6	98.5

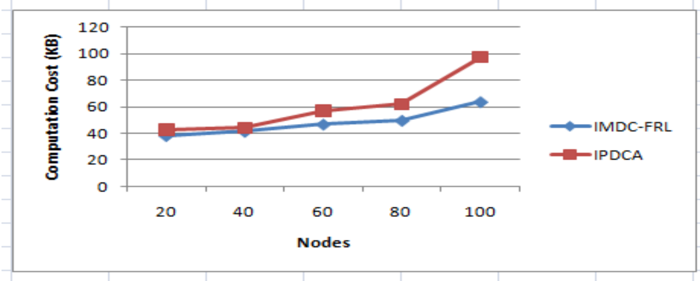

Figure 4 Computational Cost for varying nodes

Computational Cost figures for different node counts are shown in Figure 4. According to the figure, IMDC-FRL has an 18% reduced computational cost compared to IP-DCA.

Table 5 Output of Residual Energy

Nodes	IMDC-FRL (Joules)	IP-DCA (Joules)
20	12.57	11.10
40	12.74	11.57
60	13.55	12.33
80	13.57	12.65
100	13.73	13.10

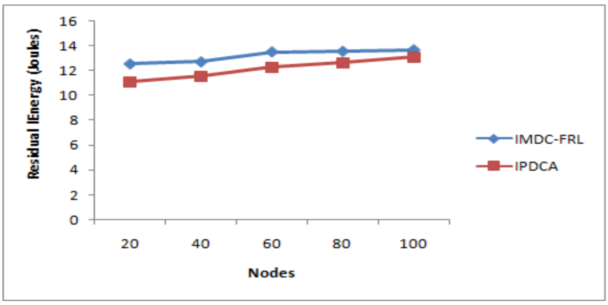

Figure 5 Residual Energy for erratic nodes

For different numbers of nodes, Figure 5 shows the Residual Energy values. It is evident from the figure that IMDC-FRL has an 8% higher Residual Energy than the IPD-CA.

4.3 Classification Results

The compare and contrast the classification performance of several techniques, including K-Means clustering, classic Federated Learning (F-L), and Federated Reinforcement Learning (FR-L).

The metrics for specificity, sensitivity, and accuracy for all three ML algorithms in Table 6 and Figure 6.

Table 6 Results of Accuracy, sensitivity and specificity

Metrics(M)	FR-L	F-L	K-Means
Accuracy	0.95	0.93	0.88
Sensitivity	0.85	0.78	0.74
Specificity	0.92	0.88	0.86

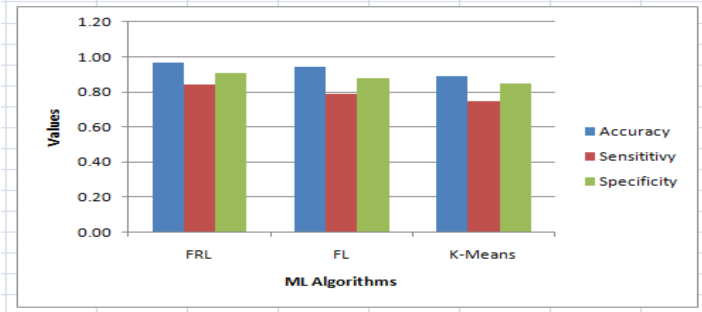

Figure 6 Results of Accuracy, sensitivity and Specificity

According to Figure 6, the FRL method has a 2% better precision compared to FL and an 8% better accuracy compared to the K-Means Algorithm. In addition, FRL's sensitivity surpasses FL's by 6% and exceeds the K-Means algorithm's by 11%. Similarly, the K-Means process has a specificity of 7%, but FRL has a specificity that is 3% higher than FL.

Figure 7 and Table 7 show the outcomes of the three Machine Learning algorithms' Recall(r), Precision(p), and F1-score(F1-s) assessments.

Table 7: Results of Recall, Precision and F1-score

Metrics(M)	FR-L	F-L	K-Means
Recall(r)	0.93	0.92	0.88
Precision(p)	0.45	0.44	0.42
F1 score (F1-s)	0.62	0.61	0.58

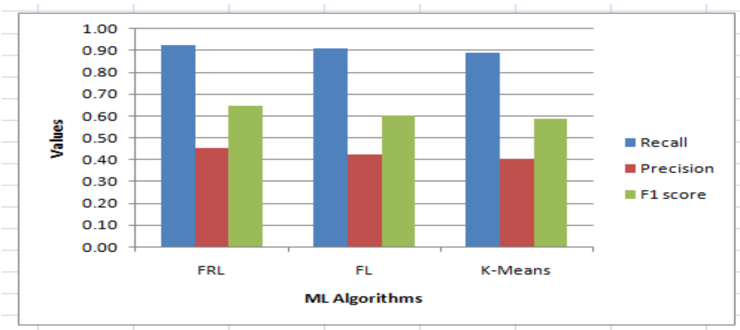

Figure 7. Results of Recall(r), Precision(p) and F1-score(F-1).

Figure 7 shows that when compared to FL and the K-Means Algorithm, FRL has a 2% higher Recall and a 4% higher Algorithm. With a 7% advantage over FL and an 11% advantage over the K-Means algorithm, FRL also has superior precision. Not only that, but the FRL's F1-score is 7% better than FL's and the K-Means algorithm's is 9% better.

5. Conclusion

To summarize, the framework suggested for Intelligent Mobile Data gathering (IMDC) in IoT-based sensor networks offers a methodical way to increase the effectiveness of data collection and the usage of resources. The system efficiently categorizes clusters of IoT devices by arranging them based on geographical proximity and use Frequent Reinforcement Learning (FRL) to identify patterns in data creation. It then modifies data gathering tactics accordingly. The use of local and global models created using reinforcement learning techniques enables the efficient gathering and distribution of data collection parameters, resulting in decreased latency and energy usage. The application of the suggested framework in NS2 illustrates its efficacy in improving precision and efficiency. In the future, additional research and testing might focus on the ability of intelligent mobile data gathering systems to handle larger amounts of data and be used in real-world Internet of Things (IoT) installations. This would help to ensure that these systems continue to progress and can be used in an extensive range of practical applications.

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 V ibra tion Sensor Data C ollection 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. Wenyuan Xu, Weiwei Fang, Yi Ding, Meixia Zou And Naixue Xiong,"Accelerating Federated Learning for IoT in Big Data Analytics With Pruning, Quantization and Selective Updating", Special Section On Intelligent Big Data Analytics For Internet Of Things, Services And People, IEEE Access, March 2021, DOI: 10.1109/Access.2021.3063291
13. Seungho Yoo and Woonghee Lee, "Federated Reinforcement Learning Based AANs with LEO Satellites and UAVs", Sensors, 2021, 21, 8111. https://doi.org/10.3390/s21238111
14. Qiang Yang,Yang Liu,Tianjian Chen,Yongxin Tong, "Federated Machine Learning: Concept and Applications', ACM Transactions on Intelligent Systems and Technology, Volume 10, Issue 2,March 2019, Article No.: 12, pp:1–19, https://doi.org/10.1145/3298981
15. Dinh C. Nguyen, Ming Ding, Pubudu N. Pathirana, Aruna Seneviratne, Jun Li, and H. Vincent Poor, "Federated Learning for Internet of Things: A Comprehensive Survey", IEEE Communications Surveys & Tutorials, 16 April 2021