Power System Technology
ISSN:1000-3673

Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

A Q-Learning Clustering Approach for Load Balancing in Edge IoT Networks

Saifeddine Choudar, Chirihane Gherbi, Zibouda Aliouat

LRSD Laboratory, computer science department, Setif 1 Ferhat ABBAS University, Setif, Algeria

Abstract: The Internet of Things (IoT) has become one of the most crucial topics in technology, used in large fields such as healthcare systems, industrial automation, traffic monitoring, and smart city infrastructure. In IoT networks, smart devices are equipped with sensors and/or actuators that allow them to perceive and react to the environment, and this gives them the ability to collect and analyze massive volumes of data from various sources; the main challenges are managing and analyzing this data generated by the IoT devices. Clustering is a technique used in IoT to group devices together based on their characteristics, such as location, type, and usage patterns. It is an important tool for managing and making sense of the large amounts of data generate by IoT devices. It can help improve IoT networks' efficiency, security, and scalability. Several clustering protocols have been developed to enhance data collection performance in IoT networks. Most focus on partitioning networks with static topologies, which are not optimal in IoT networks. This paper proposes a dynamic clustering approach in IoT networks using reinforcement learning. Our solution can improve the load balancing. Our preliminary simulation results show that the proposed Q-learning solution can achieve higher load balancing scores compared to current static solutions.

Keywords*: Internet of Things, Clustering, Edge computing, Machine Learning, Reinforcement Learning, Q-Learning.*

1. Introduction

The Internet of Things (IoT) represents a novel paradigm in which devices, sensors, and systems can collect and exchange data over the Internet. These objects can communicate and collaborate to achieve common goals [1]. Interest in Internet of Things (IoT) technology has been sparked as it covers various fields such as smart cities, industrial automation, transportation, military, and others [2, 3]. However, the limited resources of Internet of Things network components including memory, energy, and processing present several challenges, such as device management, handling large amounts of data, communication, storage, processing, and security [4, 5]. Edge Computing involves placing storage and processing units close to the data source. It processes data faster, more securely, and in real time [6]. Moreover, it has demonstrated its efficiency as a solution for data processing in Internet of Things scenarios [7].

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

1.2. Motivation

The Internet of Things (IoT) has revolutionized modern technology, enabling seamless integration between physical devices, communication networks, and data processing systems. As IoT networks growing exponentially, both in scale and in heterogeneity, ensuring efficient operation has become increasingly critical. Among the many challenges facing IoT networks, load balancing and clustering stand out as pivotal to optimizing performance, conserving energy, and maintaining scalability.

Load balancing is essential to distribute tasks across network resources effectively, preventing bottlenecks and over-utilization of specific nodes. Without an efficient load balancing mechanism, IoT networks are prone to latency, energy inefficiency, and reduced quality of service (QoS). Similarly, clustering has emerged as a key approach for managing the complexity of IoT networks by organizing devices into manageable groups. Proper clustering improves network scalability, enhances data aggregation, reduces communication overhead, and extends the lifespan of resource-constrained devices.

As the number of IoT devices grows, the data generated significantly increases, making it essential to find efficient solutions for storing and processing this vast volume of data. Clustering emerges as a crucial technique, conserving device resources and simplifying data management. It effectively addresses challenges such as energy consumption, latency, scalability, routing overhead, and the handling of large data volumes by enabling efficient data aggregation with minimal communication and distributing it for processing.

However, achieving optimal load balancing and effective clustering in edge IoT networks is far from straightforward. IoT devices are inherently resource-limited, geographically dispersed, and operate under dynamic conditions such as fluctuating workloads, intermittent connectivity, and mobility. These complexities necessitate the development of intelligent, adaptive strategies that can balance trade-offs between energy efficiency, latency, and throughput while ensuring the scalability and reliability of IoT ecosystems. Therefore, integrating machine-learning techniques with network clustering has the potential to enhance accuracy and performance, leading to improved energy efficiency and an extended lifespan of IoT networks.

This paper addresses these challenges by proposing an innovative approach to load balancing and clustering in edge IoT networks. By leveraging reinforcement learning, we aim to enhance network performance and reliability. Our contributions seek to provide scalable, energy-efficient solutions that can adapt to the dynamic nature of IoT environments, ultimately paving the way for more robust and sustainable IoT ecosystems.

1.3. Contribution

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

In this article, we propose an innovative approach to enhance load balancing in edge IoT networks. Based on a reinforcement learning (RL) model. The approach dynamically identifies network-clustering configurations and uses the Q-learning algorithm to predict the optimal future configuration, thereby improving network performance and efficiency.

In summary, the approach is described as follows:

- The network installation is initially performed randomly.
- The formation of the first clusters is determined by calculating the distance from the edge servers, measured in hop count.
- After that, the clustering configuration is dynamically obtained, followed by re-clustering and prediction using the Q-learning algorithm, which exploits the current configuration and the data collected from each edge server.

1.4. Outline

The paper's reminder is organized as follows: Section 2 presents the background and related work. Section 3 outlines the proposed approach. Section 4 presents the evaluation and discussion. Finally, in section 5, we conclude and provide a perspective.

2. Related work

In the last few years, researchers have explored how edge computing can enhance Internet of Things environments, proposing various solutions to tackle challenges like clustering and load balancing in IoT networks.

A. Edge computing for IoT networks

Many studies have explored the paradigm of edge computing in the Internet of Things (IoT), examining its architecture, security mechanisms, and key technologies. These studies highlight its potential to support diverse IoT applications such as smart cities, industrial automation, and agriculture. Furthermore, they emphasize its role in enhancing privacy and reducing latency, positioning edge computing as a critical enabler for the digital transformation of various industries.

Quy et al. [8] provide a comprehensive study on the role of edge computing in enabling real-time Internet of Things (IoT) applications, emphasizing its advantages over traditional cloud computing. The authors highlight the limitations of cloud computing in meeting the demands of latency-sensitive IoT services and discuss how edge computing addresses these challenges by bringing computation closer to the data source. Their study explores the evolution of edge computing, its applications, and the associated research challenges, while advocating for the integration of intelligence at the network edge to usher in the next era of intelligent edge systems. This work underscores the transformative potential of edge computing in

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

revolutionizing IoT applications, particularly in domains requiring low latency and high reliability.

In [9], Yu et al. present a comprehensive survey on edge computing, highlighting its role in enhancing the performance of IoT networks. The study categorizes edge computing architectures and evaluates their impact on key performance metrics such as network latency, energy consumption, and computational overhead. And examine security challenges in edge computing, assessing availability, integrity, and confidentiality. The survey also provides a comparative analysis of various IoT applications, under both edge and cloud computing architectures, demonstrating the advantages of edge computing in real-time data processing and resource optimization.

Ali et al. integrated the IoT edge-computing paradigm with three machine learning algorithms decision trees, SVM, and CNN to enhance data accuracy and network performance [10].

B. Clustering and load balancing in IoT networks

K-means is a fundamental clustering algorithm in unsupervised machine learning widely used in the literature, as mentioned in [11]. It starts by initializing K centroids randomly. Then, it assigns each data point to the nearest centroid. The centroids are then updated to reflect the average of the assigned data points. This process is repeated until the centroids no longer change. Finally, the algorithm provides K distinct clusters of data points. In [12], S. Kumar and Z. Raza applied K-Means clustering to improve resource allocation in an IoT network.

Liu et al [13] introduce a dynamic clustering solution for IoT networks in the context of edge computing to balance the number of communication hops and solve data collection challenges, especially in the case of mobile objects. Their solution use deep reinforcement learning to create an IoT clustering solution.

In [14], Malha et al. introduce a novel approach for optimizing load balancing in IoT networks. Their technique partitions the traffic load by grouping devices, leverages a deep learning model to predict future traffic fluctuations, and employs a genetic algorithm to distribute the load across servers based on these predictions.

To achieve efficient routing within IoT networks, Thangaramya et al. [15] introduces an innovative routing algorithm that combines energy-aware clustering with a neuro-fuzzy approach, effectively balancing the energy load among sensor nodes. This algorithm takes into account four parameters related to the cluster head: residual energy, distance from nodes and the base station, and degree. The neuro-fuzzy algorithm utilizes these parameters to form effective clusters.

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

3. Proposed model

A. Network model

Our model represents a network consisting of IoT devices and edge computing servers. The IoT devices are scattered randomly throughout the area and generate data at regular intervals, which they transmit via multi-hop communication to the edge-computing servers. We assume that each IoT device periodically sends data of a fixed size to the edge server of its respective cluster.

An event such as a node's death or revival can cause a change in a network's topology. This change can affect the network's connectivity, routing, and overall performance. Sometimes, the network may need to reconfigure itself to adapt to the new topology and maintain its functionality.

The architecture in our model is composed of clusters. Each cluster includes some IoT devices and an edge server as a cluster head (CH). Each edge server is responsible for collecting and processing data in one cluster. At first, the clusters are formed based on the edge computing servers. After that, the clustering depends on the quantity of data received for each edge server.

To ensure a balanced distribution of data size among all Edge servers, Clustering should be adjusted when one edge server receives a large amount of data compared to others. In this context, we propose a Reinforcement Learning (RL) approach to find optimized clustering in the IoT network to ensure load balancing.

B. Proposed Approach

In our model, the network consists of many IoT devices and edge servers partitioned into clusters. In the initial phase, the network is partitioned based on the edge servers. Each node calculates its distance to the edge server and selects the closest one as the CH. After that, clusters are formed, with each edge server is located in a cluster. The process for the initial installation of clusters is outlined in Algorithm 01.

Algorithm 01: clusters initialization
Begin
For each edge server ES:
Broadcast HELLO_MESSAGE (ES.id, hop_counter=0)
For Each node N:
N.min_hops = ∞
N.CH = NULL
On receiving HELLO_MESSAGE (ES.id, hop_counter):
If hop_counter < N.min_hops ***then***
N.CH = ES.id
N.min_hops = hop_counter
hop_counter = hop_counter + 1
forward HELLO_MESSAGE (ES.id, hop_counter)
to neighbors
Else:

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

```
            Remove HELLO_MESSAGE
        End if
End
```

Data generated by IoT devices is stored on dedicated edge servers. When one edge server receives a large amount of data compared to others, clustering should be adjusted to ensure a balanced distribution of data size among all Edge servers. Figure 1 provides a detailed description of the proposed approach.

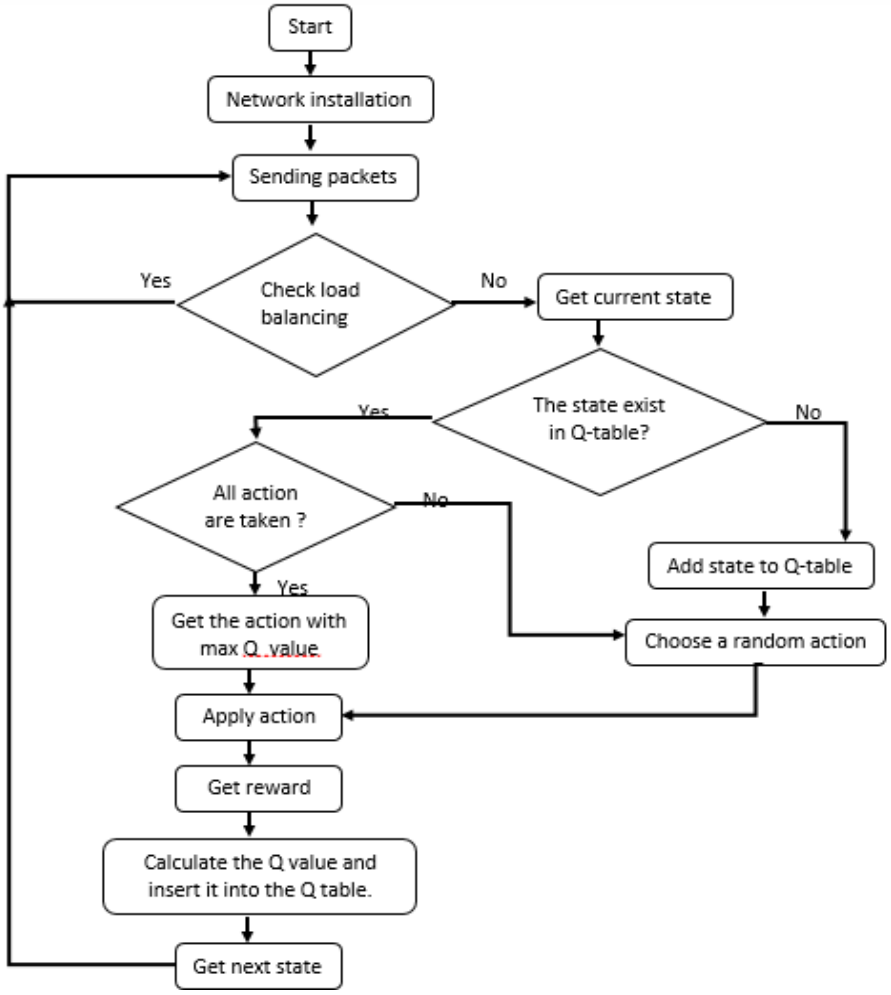

Fig. 1. The description of the proposed approach .

The proposed approach uses Q-learning as a reinforcement learning (RL) model to find the best network clustering. To manage the partitioning of the network into clusters, an agent running the QL model resides in one of the edge servers.

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

The details of Q-learning algorithm are as follows:

Action:

In our model, the action involves modifying the cluster of nodes. For each load-balancing problem, the agent selects nodes and determines the new cluster to which they should be assigned to ensure load balancing.

Reward:

We are interested in load balancing on the edge servers. As this factor becomes more important, the associated reward should increase. The network clustering must satisfy the following requirements:

Supposing di the size of data collected in edge server ESi.

The data collected D in all edge servers is:

$$D = \sum_1^N di \quad (1)$$

These collected data must be balanced across all edge servers. Therefore, in each instant Ti, di should be approximately equal to the average of the data collected in all edge servers.

Standard deviation is a statistical metric that reflects the extent of variation or spread within a dataset. We use it to indicate how dispersed the data collected in a cluster are from the average of the data collected (the mean value). Load balancing is ensured when the standard deviation is equal to zero. Moreover, if the standard deviation increases, load balancing is not ensured.

The formula of the reward R is as follows:

$$R = \sqrt{(\sum_1^N (di - m)^2)/N} \quad (2)$$

Where:

m: is the mean of data collected in all clusters.

di: is the data collected in the cluster i.

N: the number of clusters.

Q-table:

The Q-table is a rectangular array in which the columns represent the potential actions that the agent can take, while the rows correspond to the possible states of the environment. In our methodology, the states represent the clusters (the set of nodes), and the actions determine the transitions of nodes between clusters, specifying both the source and destination clusters. Moreover, a Q-value is assigned to each pair (state - action).

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

The Q-value, represented as Q(s,a), indicates the expected cumulative future rewards that an agent can obtain by taking a particular action in a particular state and following a particular policy. The Q-value is updated based on the Q-learning update rule by following:

$$Qnew(s,a) = Qold(s,a) + \alpha \times [\frac{1}{R} + \gamma \times maxQ(s',a') - Qold(s,a)] \quad (3)$$

Where:

Qold(s, a): Current Qvalue for state s and action a.

α: Learning rate (0<α≤1).

R: the reward obtained after taking action a in state s.

γ : Discount factor (0<γ≤1).

s′: The next state reached after taking action a in state s.

a′: The action that maximizes the Qvalue in the s′

The update rule computes the new Qvalue for the current state-action pair [Q(s,a)] by combining the current Qvalue (exists in the Q-table), the immediate reward (R), and the maximum Qvalue of the next state max(Q(s',a')).

To maintain load balancing at each time t, the agent runs Algorithm 02 for every series of packet transmissions, since multiple packets are sent at once:

Algorithm 02:
Begin
Do
Get the current state
Get nodes to send
Sending packets
Calculate the collected data in each edge server
// Check the load balancing
If (standard deviation > = 150) ***then***
// Check if the current state exists in the Qtable
If exist ***then***
//Check if the six action is taken
If all actions are taken ***then***
Get the action with max Qvalue from Qtable
Else
Get random action
End if
Else
Add the state to the Qtable
Get random action
End if
Apply action

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

```
    Calculate the reward
    Calculate the Q_value
    Make the Qvalue in the Qtable
    // Q(s,a) = Qvalue
    Get the next state
  While (End sending)
End
```

4. Evaluation and discussion

We evaluated the performance of our model in the context of a square network with a side of 100 m, using three methods of deployment: random, Gaussian, and cluster-Gaussian distribution. As a first test, we evaluate the proposed model with 20 nodes and 3 clusters. Each node sends several packets containing 100 bytes of data; the range of transmission is 20 m. To verify the effectiveness of the proposed approach, we evaluate its performance by comparing it to a static clustering algorithm (in this study K-means).

The parameters of the simulation are described in table 1:

PARAMETERS OF SIMULATION

Parameter	Value
Number of nodes	20
Number of clusters	3
Network size (m × m)	100 × 100
Iteration number	up to 2000
Discount factor	0.9
Learning rate	0.9
Max difference	150
Communication range	20
Round of sent	200 to 2000

Figure 2 illustrates the amount of data collected over time across three edge servers (d1, d2, and d3) using a static clustering approach (k-means). Each curve represents how data accumulation evolves on a specific edge server. Where we notice the widening gap in the data collected over time, between d1, d2 and d3. As for Figure 3, which represents the data collected at each edge server using the proposed approach, the curves are very close, which indicates that the data is well distributed.

Figure 4 represents the standard deviation of the collected data, which indicates the load balancing among the three edge servers. The standard deviation in the static approach is higher compared to the proposed approach, meaning that load balancing is better ensured in the

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

dynamic proposed approach. This is because a higher standard deviation indicates instability in load balancing.

The results obtained from the simulation show that the proposed approach efficiently ensures load balancing. Figures 2, 3, and 4 provide an overview of the difference between the proposed dynamic clustering protocol and the k-means clustering algorithm (static approach).

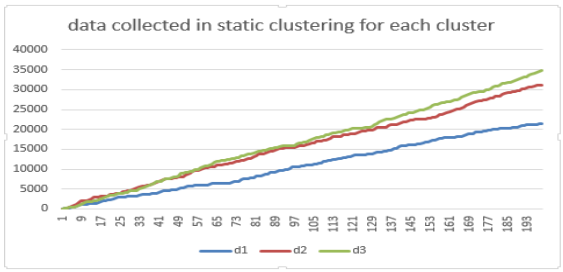

Fig. 2. Data collected in static clustering for each edge server.

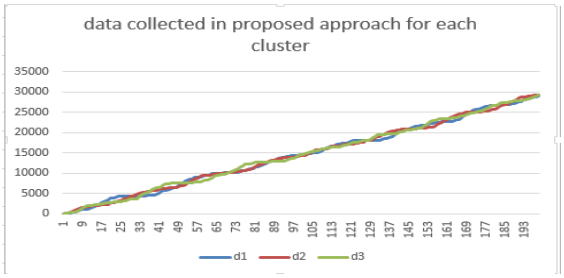

Fig. 3. Data collected in in the proposed approach for each edge server.

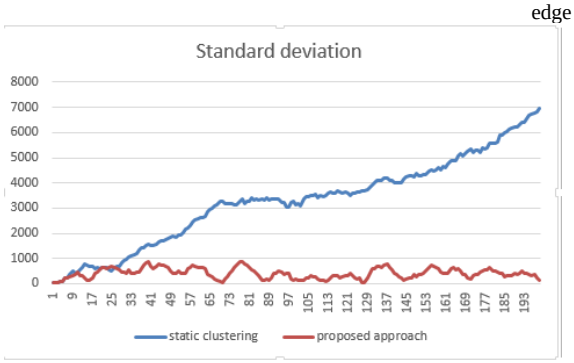

Fig. 4. The difference in load balancing between static clustering and our approach.

5. Conclusion

This paper proposes a load balancing clustering approach for IoT networks with edge servers based on a Q-learning algorithm. The objective is to improve the clustering optimization of data collection by ensuring load balancing data collected in Edge servers. The preliminary experiments show that the proposed approach is more effective than the static clustering solutions in load balancing. In this research, we are interested in load balancing. However, in future studies, we will expand the study to include other parameters, such as network lifetime and energy consumption.

Refrences

[1] D. Giusto, A. Iera, G. Morabito, L. Atzori (Eds.), The Internet of Things, Springer, 2010. ISBN: 978-1-4419-1673-0.

[2] I.K.L. Lee, "The internet of things (IoT): applications, investments, and challenges for enterprises," Business Horizons, vol. 58, no. 4, pp. 431-440, 2015.

Power System Technology
ISSN:1000-3673
Received: 06-02-2025 Revised: 15-03-2025 Accepted: 10-04-2025

[3] A.H. Ngu, M. Gutierrez, V. Metsis, V.S. Nepal, Q.Z. Sheng, "IoT middleware: a survey on issues and enabling technologies," IEEE Internet of Things Journal, vol. 4, no. 1, pp. 1-20, 2016.

[4] S.H.X. Chen, "A vision of IoT: applications, challenges, and opportunities with China perspective," IEEE Internet of Things Journal, vol. 1, no. 4, pp. 349-359, 2014.

[5] L. Farhan, S. Aslam, I. Khan, "A Survey on the Challenges and Opportunities of the Internet of Things (IoT)," in Proceedings of the 2017 Eleventh International Conference on Sensing Technology (ICST), Sydney, Australia, 4–6 December 2017, pp. 1–5.

[6] K. Cao, Y. Liu, G. Meng, Q. Sun, "An Overview on Edge Computing Research," IEEE Access, vol. 8, pp. 85714–85728, 2020. doi:10.1109/access.2020.29917

[7] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.

[8] Quy, N.M., Ngoc, L.A., Ban, N.T., Hau, N.V., Quy, V.K.: Edge computing for real-time internet of things applications: future internet revolution. Wireless Pers. Commun. 132(2), 1423–1452 (2023)

[9] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, “A survey on the edge computing for the internet of things,” IEEE access, vol. 6, pp. 6900–6919, Nov. 2017.

[10] M. Ali, H. Khan, M. Rana, A. Ali, M. Baig, S. ur Rehman, Y. Alsaawy, " A Machine Learning Approach to Reduce Latency in Edge Computing for IoT Devices" Engineering, Technology & Applied Science Research, vol. 14, no. 5, pp. 16751-16756, August, 2024.

[11] M. Usama, J. Qadir, A. Raza, et al., "Unsupervised machine learning for networking: techniques, applications and research challenges," IEEE Access, vol. 7, pp. 65579-65615, 2019.

[12] Kumar, S., & Raza, Z. (2018, January). A K-means clustering based message forwarding model for Internet of Things (IoT). In 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 604-609). IEEE.

[13] Q. Liu, L. Cheng, T. Ozcelebi, J. Murphy, J. Lukkien, "Deep Reinforcement Learning for IoT Network Dynamic Clustering in Edge Computing," in 2019 19th IEEE/ACM International Symposium on Cloud and Grid Computing (CCGRID), 2019.

[14] Merah, Malha, Zibouda Aliouat, and Hakim Mabed. "Dynamic load balancing of traffic in the IoT edge computing environment using a clustering approach based on deep learning and genetic algorithms." Cluster Computing 28.2 (2025): 77.

[15] K. Thangaramya, K. Kulothungan, R. Logambigai, M. Selvi, S. Ganapathy, A. Kannan, "Energy aware cluster and neuro-fuzzy based routing algorithm for wireless sensor networks in IoT," Computer Networks, vol. 151, pp. 211-223, 2019.