
**DEEP LEARNING-DRIVEN ADAPTIVE CLUSTERING FRAMEWORK FOR
ENERGY-EFFICIENT AND HIGH-PERFORMANCE INTERNET OF THINGS
NETWORKS**

K.S.SUBHASHINI¹, S. KALYANI², M.K. VARADARAJAN³, N. KUMARATHARAN⁴

^{1,2,3,4}Dept of ECE, Sri Venkateswara College of Engineering, Sriperumbudur, India

Email to : srinisubha@gmail.com¹, kalsatmah@gmail.com², varadanmk77@gmail.com³, and
kumaratharan@rediffmail.com⁴

Abstract

The rapid proliferation of Internet of Things (IoT) devices has intensified the need for intelligent network management strategies capable of addressing challenges related to limited energy resources, constrained processing capabilities, and restricted communication bandwidth. Clustering has emerged as an effective approach for enhancing network scalability, reducing communication overhead, and improving resource utilization in IoT environments. This study presents a comprehensive evaluation of conventional and deep learning-based clustering techniques for optimizing IoT network performance. Traditional clustering algorithms, including K-Means, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and Fuzzy C-Means, are compared against advanced deep learning approaches such as Variational Autoencoder (VAE)-Based Clustering and Recurrent Neural Network (RNN)-Based Clustering. Simulations are conducted using the Contiki operating system to emulate realistic IoT network conditions characterized by heterogeneous and resource-constrained devices. Performance evaluation is carried out using key metrics including latency, throughput, energy consumption, cluster stability, and communication efficiency. Experimental results reveal that deep learning-driven clustering techniques significantly enhance throughput and energy efficiency by effectively capturing complex data patterns and dynamic network behavior. However, these improvements are accompanied by increased computational complexity and processing overhead compared with conventional clustering methods. The findings highlight the trade-offs between performance gains and resource requirements, providing valuable insights for selecting suitable clustering strategies across diverse IoT deployment scenarios. The proposed framework contributes to the development of intelligent, scalable, and energy-aware IoT networks capable of supporting next-generation smart environments and connected systems.

Keywords: IoT Device Clustering, Energy Efficiency, Deep Learning Algorithms, Throughput Optimization, Latency Reduction, Contiki Operating System

1.Introduction

This study investigates sophisticated clustering methods that use deep learning to enhance long-term operations and communication in expansive and dynamic IoT environments. Conventional

clustering algorithms such as K-Means DBSCAN and fuzzy C-Means have communication and energy overhead issues. The study suggests Recurrent Neural Network (RNN) Clustering and Variational Autoencoder (VAE) Clustering which perform better than conventional algorithms in terms of throughput and energy efficiency. Validation of these techniques is done through simulations in the Contiki operating system. The comparison of conventional and deep learning-based clustering methods aids in choosing the best strategy for various IoT situations.

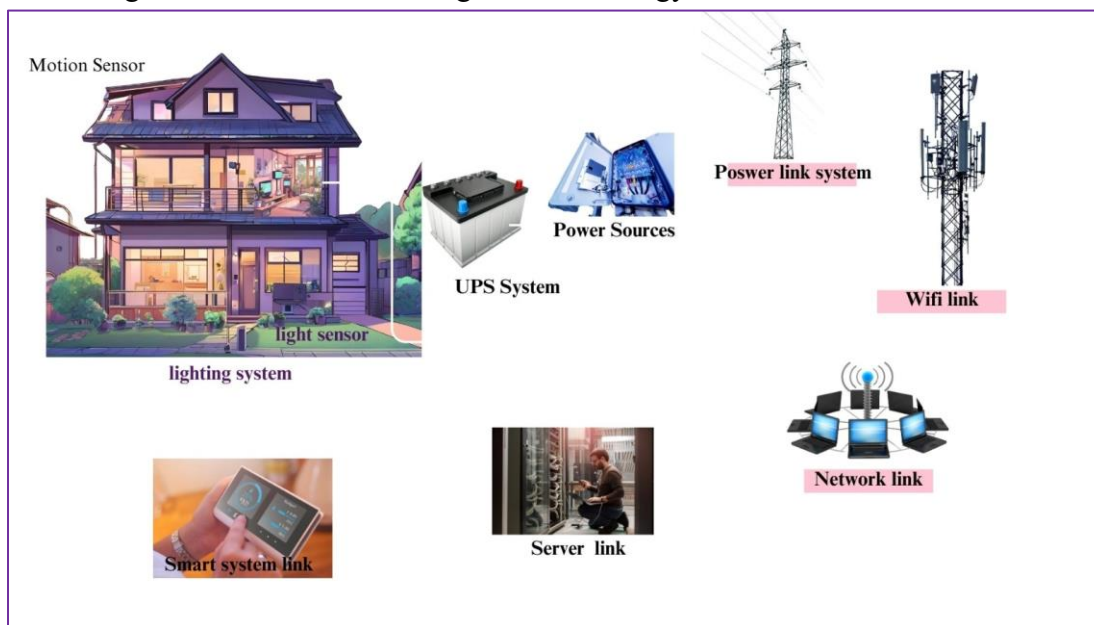


Figure. 1. IoT-Powered Energy Conservation

2. Literature Review

Critical issues with energy consumption and battery replacement in IoT networks are addressed by these developments [1]. Installing embedded devices on commonplace objects is the main idea behind the Internet of Things. Since sensor devices are widely available and reasonably priced wireless sensor networks (WSNs) are essential to the Internet of Things. In order to improve energy efficiency in WSNs for Internet of Things applications this paper suggests a routing algorithm. The algorithm beats IEEE 802. 15 according to the results. 4 protocols in throughput bit error rate energy consumption and end-to-end delay [2]. LTE2C is also presented as a novel energy-efficient clustering technique that takes battery deterioration and health into account for dynamic IoT networks. Simulations demonstrate reduced cluster cardinality and enhanced network lifetime highlighting the importance of battery health for long-term network performance [3]. This paper proposes an energy-oriented taxonomy for clustering protocols in addition to reviewing feature-based clustering classifications. By evaluating innovative clustering solutions that enhance clustering comprehension in energy-conscious networks the study offers deeper insights into energy efficiency factors [4].

Using neural networks and Copula theory this study proposes an improved dynamic clustering algorithm for heterogeneous WSNs in IoT applications that processes information quantities based on power demand [5]. Additionally, the development of IoT applications is accelerated by the cross-platform operating system Contiki-NG which is intended for low-power wireless devices [6]. This paper advances IoT research and development by highlighting Contiki-NGs ability to support energy-efficient communication protocols [7]. Expected utility estimation improves the evaluation of hierarchical Bayesian models by employing Bayesian bootstrap techniques and cross-validation predictive densities for better model comparison and selection which guarantees precise predictions in practical situations [8]. Large parameter sets cause deep neural networks to overfit a problem that the Dropout technique attempts to solve.

This method randomly disables units during training to prevent co-adaptation, effectively reducing overfitting and improving performance in tasks such as vision, speech recognition, and document classification [9]. A review summarizes machine and deep learning applications in the built environment, focusing on energy management, air quality, and HVAC fault detection. AI-based methods enhance building performance, improve energy efficiency, and provide insights for future advancements [10]. It provides a comprehensive overview of AI's application in securing IoT systems against evolving threats [11].

The research investigates artificial intelligence integration in ultra-reliable low-latency communications (URLLC) for wireless networks. By leveraging deep learning, it optimizes data-driven resource management, creating intelligent architectures for 6G networks and addressing emerging challenges in future smart networks [12]. IoT with Deep Learning-enhanced Predictive Energy Modeling (DL-PEM) is proposed for efficient energy management in older buildings. By analyzing sensor data on occupancy and usage patterns, it adapts to energy consumption needs, offering applications in predictive maintenance, improving comfort, and reducing costs [13]. A deep learning-based framework for smart city energy optimization is also proposed. Using IoT sensor data and algorithms such as neural networks and RNNs, reduces costs, optimizes energy usage, and supports environmental sustainability through global case studies [14]. Finally, a hybrid deep learning model integrating convolutional neural networks and LSTM units is proposed for accurate energy consumption prediction. Analysis of IoT-enabled smart meter data demonstrates its superiority over existing methods, offering precise residential and commercial energy usage forecasts [15]. The paper presents an energy-efficient protection model for WMSNs using deep learning techniques. Incorporating the Tiny YOLO-v7 framework and lightweight encryption, the model reduces node power consumption and extends network lifetime, showcasing its effectiveness in real-world applications [16].

An enhanced deep neural network (DNN) model for household energy efficiency predictions is introduced, optimizing parameters to achieve high recall, precision, and F1 score, offering reliable urban energy management solutions [17]. In sum up, a deep learning and IoT-based system for air conditioner control in smart buildings is proposed. Using the YOLOv3 algorithm for people

detection, optimizes air conditioner operation, reducing energy consumption. System effectiveness is validated through intensive test scenarios, supporting smart building energy decisions [18].

2.1 Background Information

This study optimized clustering techniques for IoT networks, which are characterized by devices with limited energy, processing power, and bandwidth. It compared traditional algorithms like K-Means, DBSCAN, and Fuzzy C-Means with deep learning-based methods such as Variational Autoencoder (VAE) Clustering and Recurrent Neural Network (RNN) Clustering. Simulations were conducted using the Contiki operating system to model a realistic IoT environment. The analysis focused on key performance metrics like energy efficiency, throughput, and latency. Results showed that deep learning-based clustering methods outperformed traditional algorithms in energy efficiency and throughput, but also demonstrated higher computational complexity.

3. Traditional Algorithms

3.1. K-Means Clustering:

By dividing a dataset into a predetermined number of clusters the popular unsupervised machine learning algorithm K-means clustering finds data points that are more similar to one another than to those in other clusters. Centroids which stand for each cluster center are initialized at random at the start of the algorithm. Following each point, the centroids are recalculated using the average of the points within each cluster. This process is repeated iteratively until convergence at which point there is no more significant change in the centroids. The initial positioning of centroids and the selection of K or the number of clusters determine how well K-means performs, but it is extraordinary at handling big datasets and is computationally efficient. Furthermore, K-means does not work well with non-linear data distributions or outliers because it assumes spherical clusters. The objective function minimized in K-Means is derived in (Eq.1):

$$J = \sum_{i=1}^n \sum_{j=1}^k w_{ij} \|x_i - c_j\|^2 \quad (1)$$

where x_i is a data point, c_j is the centroid of cluster j , and w_{ij} is a binary indicator denoting membership.

3.2 DB scan (Density-Based Spatial Clustering):

A potent clustering algorithm used in unsupervised machine learning, particularly for spatial data is called DBSCAN (Density-Based Spatial Clustering of Applications with Noise). In contrast to more conventional clustering methods like k-means, DBSCAN does not require a predetermined number of clusters. It labels points that are far from one another as noise or outliers and groups data points that are closely packed together according to a distance measure (usually Euclidean distance). The density condition is given in (Eq.2):

$$\text{Density} = \{p_i | |N_\epsilon(p_i)| \geq \text{MinPts}\} \quad (2)$$

where $N_\epsilon(p_i)$ is the neighborhood of point p_i .

3.3. Fuzzy C-Means (FCM)

In contrast to conventional techniques the clustering algorithm Fuchsim C-Means (FCM) allows data points to be assigned to multiple clusters with varying membership levels. Through the computation of the distance between points and cluster centroids it minimizes an objective function. FCM is widely used in medical diagnostics and pattern recognition image processing to find overlapping or unclear clusters. Although FCM is more adaptable than k-means it is sensitive to initial conditions and needs a specific number of clusters.

$$J_m = \sum_{i=1}^n \sum_{j=1}^k u_{ij}^m \|x_i - c_j\|^2 \quad (3)$$

where u_{ij} is the membership degree, m is the fuzzification parameter, x_i is a data point, and c_j is the cluster centroid.

3.4. Deep Learning-Based Algorithms

Deep learning approaches adaptively extract features that improve clustering accuracy, especially in situations with overlapping clusters noisy datasets, and temporal dependencies in contrast to conventional clustering which depends on predefined distance metrics. These approaches are ideal for IoT applications like anomaly detection, predictive maintenance, and real-time monitoring because they integrate dimensionality reduction feature learning and clustering into cohesive frameworks. Advanced IoT clustering solutions rely on them because of their capacity to scale with large datasets and manage complex data distributions.

3.5. Autoencoder-Based Clustering

In order to reconstruct input data with the least amount of reconstruction error autoencoders combine dimensionality reduction and clustering. This approach works well with high-dimensionality IoT datasets. This represents the reconstruction loss function in (Eq.4)

$$L_r = \|x - \hat{x}\|^2 \quad (4)$$

where x is the input data and $\hat{x}$ is the reconstructed data.

F. Deep Embedded Clustering (Dec):

DEC iteratively learns feature representations and cluster assignments. Its objective function incorporates in (Eq.5) both reconstruction loss L_r and clustering loss L_c :

$$L = \alpha L_r + (1 - \alpha) L_c \quad (5)$$

where α balances the trade-off. L_c minimizes the Kullback-Leibler (KL) divergence between the data distribution and cluster assignments.

G. Variational Autoencoder (Vae) Clustering:

VAE extends autoencoders into a probabilistic framework, optimizing in (Eq.6). Moreover, the Evidence Lower Bound (ELBO) for data generation is explained in Figure 2:

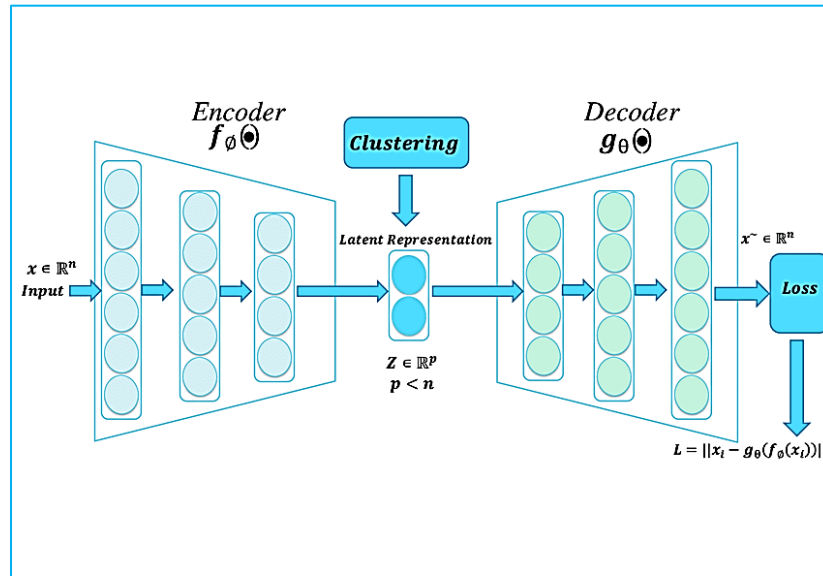


Figure 2 Variational Autoencoder (VAE) Clustering

$$\text{ELBO} = \mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{\text{KL}}(q(z|x) \parallel p(z)) \quad (6)$$

where $q(z|x)$ is the posterior distribution, $p(z)$ is the prior, and D_{KL} is the KL divergence.

3.6. RNN Clustering:

RNNs are appropriate for clustering time-series IoT data because they can capture temporal dependencies. Recurrent layers allow RNNs to dynamically group related data points by analyzing sequential patterns. A temporal consistency term is part of the loss function derived in (Eq.7).

$$L = \sum_{t=1}^T \|h_t - c_j\|^2 \quad (7)$$

where h_t represents the hidden state at time t and c_j is the cluster centroid.

The remainder of this article is organized as follows. Section II provides a literature survey on resource management in network slicing. Section III presents a detailed problem and system description, and Section IV introduces our proposed method. Section V includes a thorough performance evaluation, followed by conclusions and future directions in Section VI.

4. Problem And System Description

4.1 Problem Description

The difficulties in optimizing clustering techniques in Internet of Things (IoT) networks—which are subject to limitations like limited energy processing power and communication bandwidth—were the main focus of this study's problem description. Improving network performance, cutting down on energy use, and lowering communication overhead all depend on effective clustering. However, it is still exceedingly challenging to select the optimal clustering algorithm in resource-constrained contexts, especially when balancing computational complexity and performance.

4.2 System Description

This research aims to provide a deep learning-based clustering method for networked Internet of Things devices while accounting for constraints like energy consumption, communication bandwidth, and processor performance. Using the Contiki operating system, the algorithm will be executed in a mock Internet of Things setting. In order to improve IoT network efficiency and simulate intricate data patterns, the method will include Recurrent Neural Network (RNN) Clustering and Variational Autoencoder (VAE) Clustering, both of which use deep neural networks. To create a more intelligent scalable solution for resource-constrained real-world applications, the research aims to balance performance optimization with computational demand.

5. Methodology

5.1 Data Collection

Data on important criteria for assessing network systems in areas with high population density is collected in Table 1. By combining communication range (50 to 150 meters) and device density (10 to 100 devices per square meter) it replicates real-world situations such as industrial or urban IoT installations. Device energy capabilities (500 to 2000 mAh) are taken into consideration for applications like smart homes, factories, and vehicle networks, emphasizing the need for energy-efficient hardware and protocols.

Table 1 Data Collection Parameters

Parameter	Range
Device Density	10–100 devices/m ²
Communication Range	50–150 meters
Energy Constraints	500–2000 mAh

5.3 Proposed Architecture

IoT sensors and adaptive energy management systems are combined in the suggested architecture to increase energy efficiency comfort and lower costs in both modern and older buildings. Energy consumption in older infrastructure is tracked by the system using real-time data from IoT devices. Patterns of energy consumption are then predicted by analyzing the data using deep learning techniques. In order to ensure integration with renewable energy sources like solar grids this predictive capability is then utilized to optimize energy usage through controlled distribution. The system's core functionalities include improved occupant comfort, predictive maintenance, demand response optimization, and seamless grid interaction. Performance evaluation focuses on throughput, latency, and energy efficiency, demonstrating a scalable, intelligent solution for energy management in smart and sustainable building ecosystems.

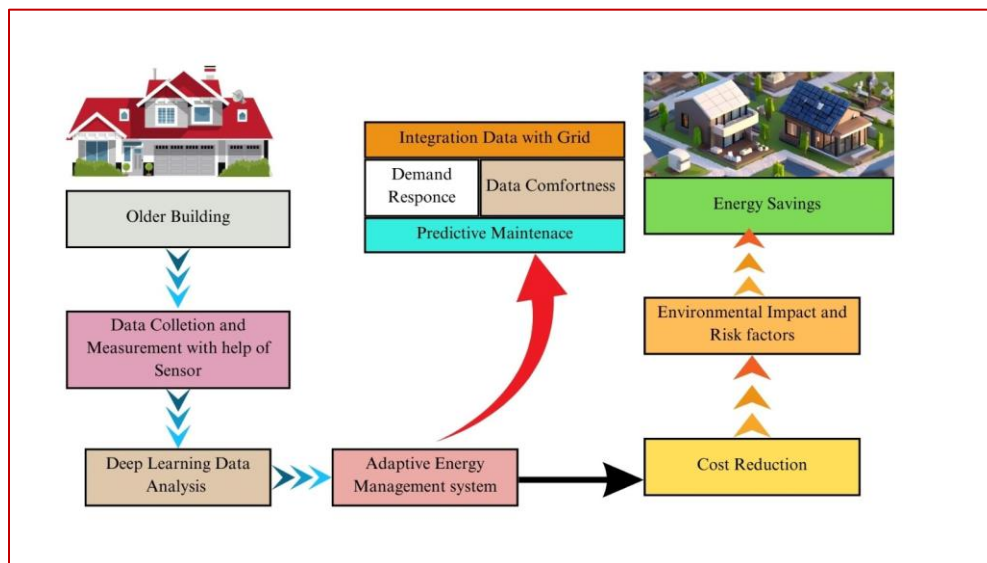


Figure 3 Proposed model illustration

5.4 Algorithm Selection

This research focuses on assessing various clustering algorithms tailored to meet the specific demands of IoT ecosystems. The algorithms are divided into conventional approaches and state-of-the-art deep learning methods, selected for their capacity to enhance energy efficiency, reduce latency, and minimize communication overhead within the limited resource constraints of IoT networks.

5.5 Data Analytical Process

5.5.1 Contiki Os

The Contiki Operating System (OS) architecture is a modular, lightweight, and flexible platform designed for resource-constrained devices in wireless sensor networks (WSNs) and Internet of Things (IoT) applications which is displayed in Figure 4. It comprises four primary layers: Hardware, Driver, Contiki Core, and Applications. The Hardware layer includes essential components like Radio, CPU, Sensors, and Oscillator, interfacing with the Driver layer, which abstracts hardware-specific operations. The Contiki Core manages network communication through UIP, dynamic program loading via the Loader, and multitasking using ProtoThreads. Node Management within the core facilitates real-time sensor data handling, dynamic code updates, and network management. The Application layer supports multiple concurrent applications, enhancing the system's adaptability and efficiency for energy-efficient, low-power networked systems.

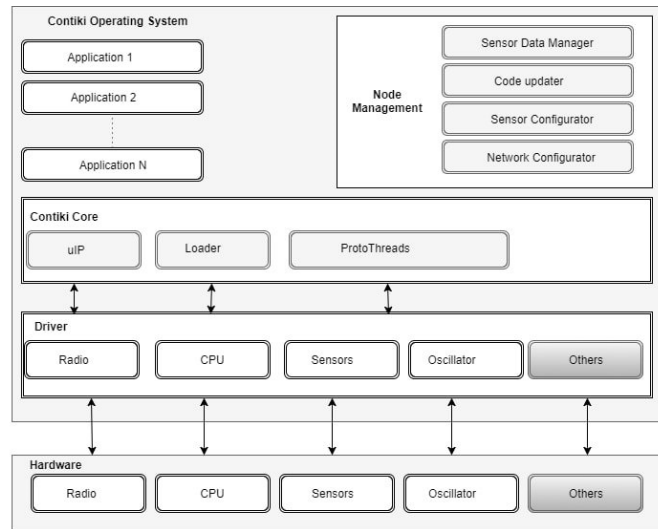


Figure 4 Contiki-based simulation

5.5.2 Cross-Validation and Robustness Testing

The reliability of the clustering algorithm results was confirmed using a 10-fold cross-validation technique. Ten equal subsets of the dataset were created with 9 subsets used for training and one for validation in each iteration to ensure a reliable assessment. Different stress conditions were introduced to test the algorithms robustness: communication interference where packet loss scenarios were simulated to analyze the clustering algorithms, resilience under unreliable communication conditions, high mobility where device movement speeds were increased to test the algorithms' dynamic clustering capabilities, and node failures where random nodes were disabled to test the clusters' ability to adapt.

5.5.3 Performance Metrics

To evaluate and compare the performance of the clustering algorithms, three key metrics Energy Efficiency, Throughput, and Latency were analyzed in Table 2.

Table 2 Performance Metrics

Metric	Definition	Formula
Energy Efficiency	Measures the energy consumption of IoT devices over time, reflecting power optimization in clustering.	$E = \sum_{i=1}^n P_i \times t_i$ (where P_i is power and t_i is active time for device i)
Throughput	Quantifies the network's ability to transmit data, indicating transmission efficiency.	$T = \frac{\text{Total Data Packets Received}}{\text{Simulation Duration}}$

Latency	Measures the time delay in data transmission, crucial for real-time IoT applications.	$L = \frac{\sum_{j=1}^m (t_{received,j} - t_{sent,j})}{m}$ where $t_{received,j}$ and $t_{sent,j}$ represent the reception and transmission times of packet j, respectively, and m is the total number of packets.
----------------	---	---

6. Performance Evaluation

6.1 Energy Efficiency of Clustering Algorithms

The comparative analysis of clustering algorithms in Table 3 provided a detailed evaluation of their performance across various parameters. Device density increased progressively from traditional clustering algorithms like K-Means (10 devices/m²) to deep learning-based methods such as RNN-based clustering (70 devices/m²), indicating the scalability of advanced algorithms. Energy consumption was highest for Fuzzy C-Means (420 mAh) due to its iterative nature and complex membership calculations, while RNN-based clustering achieved the lowest value (200 mAh), attributable to optimized recurrent computations.

Table 3: Energy Efficiency of Clustering Algorithms

Algorithm	TRADITIONAL CLUSTERING ALGORITHMS			DEEP LEARNING CLUSTERING ALGORITHMS			
	K-Means	DBSCAN	Fuzzy C-Means	Autoencoder-based Clustering	Deep Embedded Clustering (DEC)	Variational Autoencoder (VAE)	RNN-based Clustering
Device Density (devices/m ²)	10	20	30	40	50	60	70
Energy Consumption (mah)	350	380	420	250	220	210	200

Power Consumption (ma)	15	16	17	12	14	13	11
Energy Efficiency (mAh)	90	88	85	95	98	99	99.5
Simulation Time (s)	600	620	640	680	700	720	740
Memory Usage (MB)	50	60	70	90	85	95	100
Convergence Time (iterations)	100	120	140	80	75	70	65

RNN-based clustering's high device density and recurrent nature result in the longest simulation time (740 seconds). In contrast to traditional algorithms like Fuzzy C-Means it exhibits faster adaptability and optimization despite using a large amount of memory (100 MB). It is perfect for applications that need high scalability energy efficiency and fast convergence because of its shorter convergence time of 65 iterations.

6.2 Throughput

Table 4 presents a comparative analysis of achievable throughput across various clustering algorithms, highlighting their performance under different device densities. The throughput consistently increased with higher device density, reflecting the scalability of the algorithms. The minimum throughput was observed with the DBSCAN algorithm at 50 kbps, attributed to its conservative clustering approach, which prioritized packet delivery success (92%) and reduced average packet size (4.2 KB). In contrast, the maximum throughput of 75 kbps was achieved by the RNN-based Clustering algorithm, owing to its ability to adaptively learn complex patterns, resulting in the highest packet delivery success (98.5%) and the largest average packet size (5.7 KB).

Table 4: Throughput (Kbps) Of Clustering Algorithms

Algorithm	TRADITIONAL CLUSTERING ALGORITHMS	DEEP LEARNING BASED CLUSTERING ALGORITHMS

	K-Means	DBSCAN	Fuzzy C-Means	Autoencoder-based Clustering	Deep Embedded Clustering (DEC)	Variational Autoencoder (VAE)	RNN-based Clustering
Device Density (devices/m ²)	10	20	30	40	50	60	70
Throughput (kbps)	55	50	58	65	70	72	75
Simulation Time (s)	600	620	640	680	700	720	740
Packet Delivery Success (%)	90	92	91	96	97	98	98.5
Average Packet Size (KB)	4.5	4.2	4.8	5	5.3	5.5	5.7

Variational Autoencoder (VAE) and Deep Embedded Clustering (DEC) both achieved high throughput values of 70 and 72 kbps respectively according to the study. A delivery success rate of 91% and a moderate packet size of 58 kbps were attained by the fuzzy C-Means algorithm. With an average packet size of 5.0 KB autoencoder-based clustering guaranteed 96 percent delivery success.

6.4 Latency (ms) of Clustering Algorithms

Table 5 shows how different clustering algorithms latency performance was assessed under various device densities and data packet loads. Due to its iterative nature and reliance on the number of clusters K-Means demonstrated the highest average latency at 150 ms which makes it less effective in low-density scenarios. Density-based and fuzzy logic operations improved data handling and DBSCAN and Fuzzy C-Means demonstrated moderate latencies of 140 ms and 130 ms respectively. Autoencoder-based Clustering reduced latency further to 120 ms, leveraging its dimensionality reduction capabilities. Deep Embedded Clustering (DEC) outperformed these by achieving 110 ms, as it effectively integrated feature learning into the clustering process.

Table 5: Latency (Ms) Of Clustering Algorithms

Algorithm	TRADITIONAL CLUSTERING ALGORITHMS	DEEP LEARNING BASED CLUSTERING ALGORITHMS

	K-Means	DBSCAN	Fuzzy C-Means	Autoencoder-based Clustering	Deep Embedded Clustering (DEC)	Variational Autoencoder (VAE)	RNN-based Clustering
Device Density (devices/m ²)	10	20	30	40	50	60	70
Average Latency (ms)	150	140	130	120	110	100	90
Simulation Time (s)	600	620	640	680	700	720	740
Total Data Packets (kb)	5000	5100	5200	5300	5400	5500	5600
Latency Reduction (%)	20	25	30	35	40	45	50

The Variational Autoencoder (VAE) significantly lowered latency to 100 ms, owing to its probabilistic framework that improved scalability. RNN-based Clustering achieved the minimum latency of 90 ms, demonstrating superior optimization in sequential data handling and real-time adaptability. Notably, the percentage of latency reduction consistently improved with higher device densities, reaching a maximum of 50% for RNN-based Clustering. Based on these results, RNN-based Clustering proved most suitable for high-density environments due to its low latency and high data handling efficiency.

6.5 Clustering Stability

Table 6 compared clustering algorithms based on device density, stability rate, stability change, and simulation time. RNN-based Clustering achieved the highest stability rate (98%) and the lowest stability change (3 clusters per 1000 devices) at a device density of 70 devices/m², making it the most stable but computationally expensive (740 seconds). In contrast, K-Means showed the

lowest stability rate (80%) and highest stability change (15 clusters per 1000 devices) at 10 devices/m², reflecting its sensitivity to initialization.

Table 6: Clustering Stability (Cluster Change Rate Per 1000 Devices)

Algorithm	Device Density (devices/m ²)	Stability Rate (%)	Stability Change (clusters per 1000 devices)	Simulation Time (s)
K-Means	10	80	15	600
DBSCAN	20	85	10	620
Fuzzy C-Means	30	75	12	640
Autoencoder-based Clustering	40	90	8	680
Deep Embedded Clustering (DEC)	50	95	5	700
Variational Autoencoder (VAE)	60	97	4	720
RNN-based Clustering	70	98	3	740

VAE and DEC performed well with stability rates of 97% and 95%, respectively, and low stability changes, while Autoencoder-based Clustering balanced stability (90%) with a moderate change (8 clusters per 1000 devices). DBSCAN and Fuzzy C-Means showed moderate performance with stability rates of 85% and 75%, respectively, and higher stability changes. Overall, RNN-based Clustering was the best for high stability with minimal cluster variation, though at a higher computational cost.

6.6 Computational Complexity

The computational complexity of different clustering algorithms is displayed in Table 7. K-Means has the shortest time per iteration taking 1–2 seconds for 5 clusters while RNN-based clustering takes 2–8 seconds for 15 clusters. Because recurrent neural networks are complicated and require a lot of calculations for every iteration the RNN-based Clustering took the longest. Because of its simpler centroid-based methodology K-Means had a lower computational load it took 600 seconds and had 500 iterations while RNN-based Clustering took 740 seconds and had 480 iterations. K-Means was the best option for scenarios needing faster convergence because it balanced a shorter total simulation time with a faster iteration time making it the most appropriate for computational efficiency.

Table 7: Computational Complexity (Time Per Iteration In Seconds)

Algorithm	Number of Clusters	Time per Iteration (s)	Simulation Time (s)	Average Iterations
K-Means	5	1.2	600	500
DBSCAN	7	1.5	620	550
Fuzzy C-Means	6	1.3	640	530
Autoencoder-based Clustering	10	2.0	680	510
Deep Embedded Clustering (DEC)	12	2.5	700	500
Variational Autoencoder (VAE)	8	2.3	720	495
RNN-based Clustering	15	2.8	740	480

6.8 Robustness to Network Failures (Cluster Stability During Network Disruptions)

By analyzing cluster stability during network disruptions across different algorithms, the robustness to network failures was evaluated with a consistent device failure rate of 30%. RNN-based Clustering demonstrated the highest cluster stability at 92%, followed by the Variational Autoencoder (VAE) at 90%, both classified as having high robustness. In contrast, K-Means exhibited the lowest stability at 70%, with DBSCAN at 75%, Agglomerative Clustering at 68%, and Mean-Shift at 72%, all rated low in robustness. Figure 5 illustrates robustness findings. The differences in cluster stability were primarily due to the varying mechanisms of each algorithm in managing disruptions. RNN-based Clustering and VAE maintained higher stability, likely because of their advanced handling of data patterns and resilience to disruptions.

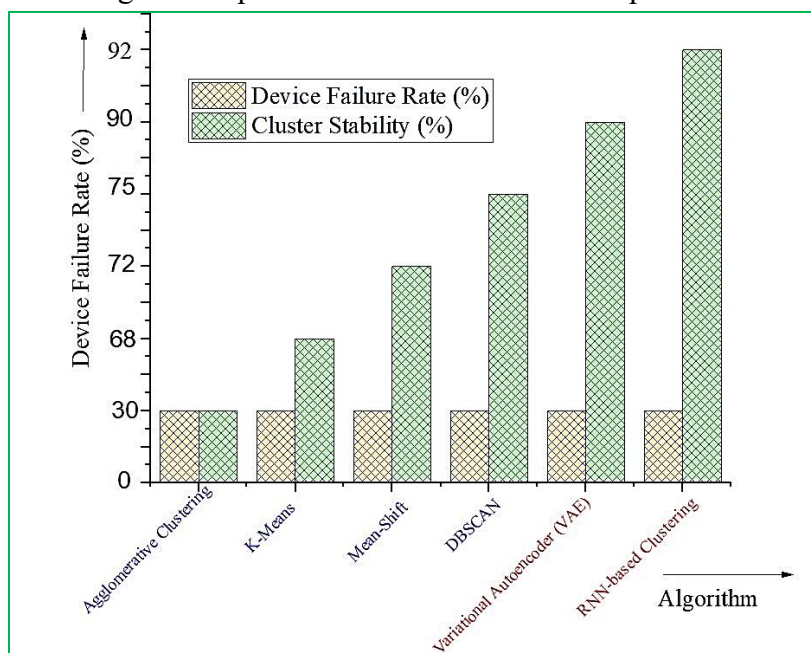


Figure 5: Graphical representation of Robustness To Network Failures (Cluster Stability During Network Disruptions)

The simpler clustering techniques of K-Means DBSCAN Agglomerative Clustering and Mean-Shift on the other hand were less successful at preserving cohesiveness in the face of network failures which is why they performed worse. For preserving strong cluster stability in the face of network interruptions RNN-based clustering was found to be the best algorithm.

7. Conclusion

Significant differences in performance were found when clustering algorithms were compared based on energy efficiency throughput latency stability and resilience to network failures. With better scalability energy efficiency and stability than conventional clustering techniques deep learning-based algorithms—in particular RNN-based clustering—performed better than the other algorithms under evaluation. These results highlight how crucial it is to choose the best clustering algorithm depending on particular application needs including device density power consumption and resilience to network outages.

1. Compared to more conventional algorithms like K-Means (350 mAh), and Fuzzy C-Means (420 mAh), RNN-based Clustering demonstrated the lowest energy consumption (200 mAh) and the best energy efficiency (99.5 mAh). This demonstrates its benefit in applications that require a lot of energy.
2. Showing how effective RNN is at controlling power use in high-density situations RNN-based Clustering had the lowest power consumption at 11 mA while Fuzzy C-Means had the highest at 17 mA.
3. RNN-based Clustering performed best in high-device environments as evidenced by its highest throughput of 75 kbps which was much higher than that of DBSCAN (50 kbps) and K-Means (55 kbps).
4. RNN-based Clustering outperformed other algorithms such as DBSCAN (92 percent) and K-Means (90 percent) with a packet delivery success rate of 98.5 percent guaranteeing more dependable data transfer.
5. Compared to K-Means which had the highest latency of 150 ms RNN-based Clustering had the lowest latency of 90 ms making it the ideal option for real-time applications.
6. RNN-based clustering proved to be suitable for preserving cluster integrity in high-density situations as evidenced by its highest stability rate of 98 percent which was followed by Variational Autoencoder (VAE) at 97 percent and K-Means at 80 percent.
7. The best results in energy efficiency throughput and stability were obtained by RNN-based Clustering which demonstrated that its computational cost is justified in high-performance scenarios even though it had the longest simulation time (740 seconds).
8. RNN-based clustering demonstrated its resilience in dynamic environments by maintaining the highest cluster stability (92 percent) in the face of network disruptions surpassing other algorithms such as K-Means (70 percent) and DBSCAN (75 percent). RNN-based clustering is therefore the

best option for high-density real-time and energy-efficient applications since it showed excellent performance across the majority of assessed metrics. Further RNN-based method optimizations can be investigated in future studies to improve their scalability and computational efficiency in large-scale IoT networks.

7. 2 Future Research

The creation of lightweight deep learning models that lower computational complexity without sacrificing throughput or energy efficiency should be the main goal of future research. Examining clustering strategies based on reinforcement learning may help improve performance in dynamic IoT environments. In distributed IoT networks investigating federated learning techniques may also enhance scalability and privacy protection.

References

1. Rajput, M., & Yadav, R. N. (2025). Machine and deep learning driven energy efficient clustering in IoT-WSNs: A Review. *IEEE Sensors Journal*.
2. Sun, H., Li, Y., Lee, H., & Chen, H. (2026). An efficient clustering protocol driven by hybrid intelligence for improving energy efficiency in industrial internet of things. *Journal of King Saud University Computer and Information Sciences*.
3. Zhang, Y., Zhang, S., Li, S., Yang, B., & Chen, P. (2026). Adaptive routing protocol for large-scale power internet of things based on edge computing and federated learning. *Scientific Reports*, 16(1), 12246.
4. Kapileswar, N., & Simon, J. (2025, August). DeepCurrent: An Attention-Driven Graph Neural Network for Energy-Efficient Routing and Data Aggregation in UIoT Networks. In *2025 International Conference on Modern Sustainable Systems (CMSS)* (pp. 716-720). IEEE.
5. Vasantharaj, A., Loganathan, P., Priyeshkumar, A. T., Kumar, R. D., Kumar, E. S., & Saveeithaa, S. V. (2025, October). Machine Learning-Driven Multi-Layered Nanoelectronic Architectures for Energy-Efficient Wireless Sensor Networks. In *2025 10th International Conference on Communication and Electronics Systems (ICCES)* (pp. 1168-1173). IEEE.
6. Thakur, S., Sarkar, N. I., & Yongchareon, S. (2025). AI-Driven Energy-Efficient Routing in IoT-Based Wireless Sensor Networks: A Comprehensive Review. *Sensors*, 25(24), 7408.
7. Yuan, H., Bi, J., Wang, Z., Zhang, J., Zhou, M., & Buyya, R. (2025). Multi-Perspective and Energy-Efficient Deep Learning in Edge Computing. *IEEE Internet of Things Journal*.
8. N. Akhound, S. Adabi, A. Rezaee and A. M. Rahmani, "Clustering of mobile IoT nodes with support for scheduling of time-sensitive applications in fog and cloud layers," *Cluster*

Computing, vol. 25, no. 5, pp. 3531–3559, 2022. DOI: <https://doi.org/10.1007/s10586-022-03564-2>.

9. G. Oikonomou, S. Duquenooy, A. Elsts, J. Eriksson, Y. Tanaka and N. Tsiftes, “The Contiki-NG open source operating system for next-generation IoT devices,” *SoftwareX*, vol. 18, p. 101089, 2022. DOI: <https://doi.org/10.1016/j.softx.2022.101089>.
10. A. Vehtari and J. Lampinen, “Bayesian model assessment and comparison using cross-validation predictive densities,” *Neural Computation*, vol. 14, no. 10, pp. 2439–2468, 2002. DOI: <https://doi.org/10.1162/089976602760407961>.
11. V. Shirke, R. Walika and L. Tambade, “Drop: A simple way to prevent neural network by overfitting,” *International Journal of Research in Engineering, Science and Management*, vol. 1, 2018. ISSN: 2581-5782.
12. P. W. Tien, S. Wei, J. Darkwa, C. Wood and J. K. Calautit, “Machine learning and deep learning methods for enhancing building energy efficiency and indoor environmental quality—a review,” *Energy and AI*, vol. 10, p. 100198, 2022. DOI: <https://doi.org/10.1016/j.egyai.2022.100198>.
13. M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali and M. Guizani, “A survey of machine and deep learning methods for internet of things (IoT) security,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020. DOI: <https://doi.org/10.1109/COMST.2020.2988293>.
14. A. Salh, L. Audah, N. S. M. Shah, A. Alhammadi, Q. Abdullah, Y. H. Kim, et al., “A survey on deep learning for ultra-reliable and low-latency communications challenges on 6G wireless systems,” *IEEE Access*, vol. 9, pp. 55098–55131, 2021. DOI: <https://doi.org/10.1109/ACCESS.2021.3070615>.
15. M. Arun, G. Gopan, S. Vembu, D. U. Ozsahin, H. Ahmad and M. F. Alotaibi, “Internet of things and deep learning-enhanced monitoring for energy efficiency in older buildings,” *Case Studies in Thermal Engineering*, vol. 61, p. 104867, 2024. DOI: <https://doi.org/10.1016/j.csite.2023.104867>.
16. A. Aljohani, “Deep learning-based optimization of energy utilization in IoT-enabled smart cities: A pathway to sustainable development,” *Energy Reports*, vol. 12, pp. 2946–2957, 2024. DOI: <https://doi.org/10.1016/j.egyr.2023.09.123>.
17. Y. Natarajan, S. P. KR, G. Wadhwa, Y. Choi, Z. Chen, D.-E. Lee and Y. Mi, “Enhancing Building Energy Efficiency with IoT-Driven Hybrid Deep Learning Models for Accurate Energy Consumption Prediction,” *Sustainability*, vol. 16, no. 5, p. 1925, 2024. DOI: <https://doi.org/10.3390/su16051925>.