STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **15**, January 2026, pp 921–935.
Published online in International Academic Press (www.IAPress.org)

Optimized Data Offloading in IoT-Based Wireless Sensor Networks †

Aya Ouaarouss [1,*], Hajar Chabbout [2], Abderrahim Zannou [1], Jalal Isaad [1]

[1] *ERCI2A, FSTH, Abdelmalek Essaadi University, Tetouan, Morocco*
[2] *Algebra and its Applications, FSTH, Abdelmalek Essaadi University, Tetouan, Morocco*

Abstract The rapid growth of Internet of Things (IoT) data is driven by the massive increase in IoT devices, leading to an explosion in data generation. However, these devices often face challenges such as limited storage capacity and low energy reserves, making local data storage and processing costly, particularly in Wireless Sensor Networks (WSNs) where additional computational resources are scarce. To address these issues, offloading data to specialized aggregator nodes is a crucial solution. However, determining whether IoT devices should store data locally or offload it to aggregator nodes remains a significant challenge. In this paper, we propose an offloading mechanism that allows resource-constrained IoT devices to transfer their data to aggregator nodes for storage or processing. First, we define the resource constraints of IoT devices that guide the design of our proposed system. Using these constraints, we determine each device's memory fill time and battery life. Next, we apply k-means clustering to group the IoT devices into constrained and unconstrained groups based on memory fill time and battery life. In the second step, the constrained devices offload their data to the aggregator nodes. To enhance the control of the WSN, the aggregator nodes forward the collected data to the base station through a gateway. The base station is responsible for training and updating the LSTM model due to its higher computational capacity, while the gateway executes the lightweight inference of the LSTM model. The K-Nearest Neighbors (KNN) classifier also runs at the gateway, using the predicted values to classify IoT devices into constrained or unconstrained groups. Simulation results show that our approach achieves higher energy efficiency, with the slowest decrease in residual energy and the fewest sensor failures compared to the two reference methods from the literature. It also maintains greater available memory space, declining only from about 86% to 30% while the other methods drop far lower. Data offloading to the aggregator nodes remains stable between 14000 KB and 18000 KB per set (100 sensors), demonstrating consistent resource utilization at large scale.

Keywords Internet of Things (IoT), Wireless Sensor Networks (WSN), aggregator nodes, Data offloading, Energy efficiency, k-Means clustering, Long Short-Term Memory (LSTM),K-Nearest Neighbors (KNN), Memory fill time, Battery life.

DOI: 10.19139/soic-2310-5070-3041

1. Introduction

The Internet of Things refers to the vast ecosystem of interconnected physical devices such as sensors, actuators and smart objects that continuously monitor and interact with their environment [1],[2]. IoT devices collect physical phenomena (temperature, humidity, movement, etc.) which are converted to real-time digital data [3].In many cases, IoT devices run continuously and can be installed in massive quantities, generating vast amounts of highly heterogenous data at a high velocity[4]. It is continuous production of this type of data that provides the basis for applications of IoT that range from Smart Cities to Healthcare to Industrial Automation and Precision Agriculture [5]. As more sensors and actuators are deployed, the number of different types of measurement and how frequently those measurements are made increases exponentially producing a larger quantity of raw data. Efficiently storing,

†*This work was carried out with the support of CNRST under the "PhD-ASsociate Scholarship – PASS" program.*
*Correspondence to: Aya Ouaarouss (Email: aya.ouaarouss@etu.uae.ac.ma).ERCI2A, FSTH, Abdelmalek Essaadi University, Tetouan, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

transmitting and processing the produced data so that the resulting data can provide meaningful insights in a timely manner [6]. Energy management remains one of the most crucial challenges for Internet of Things deployments since a significant number of nodes tend to consume batteries with finite lifetimes and are often installed in remote or inaccessibly located regions [7, 8]. A consistent replacement and recharging of the battery can lead to an increase in maintenance costs and reduce the reliability of large networks [9, 10, 11]. Various studies point out that the energy consumption of wireless sensor networks is regulated by data transmission, not computation and thus communication protocols and offloading data strategies are significant in extending battery life [12],[13]. Approaches such as duty cycling, adaptive sampling and energy harvesting have been proposed to mitigate energy drain while preserving data quality [14, 15, 16]. More recent works explore the integration of machine learning and edge intelligence to predict energy usage and schedule transmissions more efficiently [17]. Despite these advances, sustaining long term operation in dense and dynamic IoT environments continues to require innovative designs that balance performance, scalability and energy efficiency [18, 19]. In order to overcome the growing energy and storage constraints of large scale IoT deployments, recent studies have explored the integration of high capacity devices directly inside the wireless sensor network [20, 21, 22, 23]. These devices, called aggregator nodes in our work, are special IoT nodes equipped with significantly larger memory and energy reserves. Their role is to act as powerful aggregator nodes that collect and temporarily store data from nearby low power sensors. They receive their data locally, thus reducing the communication path length (and therefore the number of communications over a long distance) that contribute to the consumption of the energy from the batteries of sensors. They preserve the full distributed nature of the network, they avoid the costs and complexities associated with connectivity at the same time as they take advantage of the possibility of aggregating data locally and managing scarce resources better. As such, they enable the scaling of a WSN in a manner that maintains low maintenance costs while conserving uninterrupted operation. Therefore, the use of high-capacity IoT nodes as aggregator nodes is a key component of increasing the operational reliability and lifetime of large-scale IoT systems.Based upon this principle, this research paper develops a novel, energy-efficient data offloading mechanism for IoT-based Wireless Sensor Networks. We start by characterizing the resource limitations of IoT devices, and consequently develop the expressions for the memory fill-time and the battery-lifetime of IoT devices. Based on these two metrics, a k-means clustering algorithm is applied to divide the IoT devices into constrained and unconstrained groups. In the second phase, constrained devices offload their data to the aggregator nodes. To improve system coordination, the aggregator nodes forward the collected data to the base station through a gateway. The base station performs the training and update of the LSTM model, while the gateway executes lightweight inference. The K-Nearest Neighbors classifier is also deployed at the gateway, using the predicted values to classify devices into the two groups. Simulation results demonstrate that the proposed approach yields superior energy efficiency, exhibiting the slowest residual energy decline and the lowest sensor failure rate compared with the two reference methods from the literature.

The rest of the article is organized as follows: in Section 2 the related works are presented, in Section 3 the problem definition is given, in Section 4 the proposed approach is presented, in Section 5 the results are discussed, and the last Section 6 concludes the paper and presents the future work.

2. Related work

The work [24] explores how to obtain high-quality information from large-scale sensor networks despite limited bandwidth, energy constraints, environmental noise and node failures. It concentrates on two dimensions of data quality: reliability, defined as the closeness of collected data to the true values, and sharing, meaning the strong correlation of data among neighbouring nodes. To address these issues, the authors propose an optimisation model that, while respecting energy and sharing constraints, selects the most reliable data sources for transmission in order to maximise overall information quality in homogeneous multi-source, multi-modal networks. The complexity of the methods used may be demanding for resource-constrained IoT devices, and scalability to larger networks remains to be addressed.

The authors of [25] aim to improve distributed computing in Software Defined Networks (SDN) using smart Internet of Things to surpass wireless sensor networks limitations, i.e., energy and routing efficiency. It provides an energy-efficient routing technique for massive IoT networks, rooted in a cloud-based SDN architecture. The technique relies on the use of mobile sinks and best data collection points in constructing effective transmission

routes. Optimization algorithms are used to improve network scalability, low-level routing, and load balancing. The main contributions include minimizing the energy consumed by cluster heads and better energy balancing in the SDN network. The research results show that the proposed model improves network stability, prolongs its lifetime, and optimizes the management of energy resources and data traffic. However, further research is needed based on Data processing complexity and how much energy will be consumed in the process for larger scale networks.

The authors of work [26] investigate how cloud services can be integrated with edge computing to improve data processing in IoT applications in many fields including critical ones. They show that centralized cloud computing may have inefficiencies due to communication delay, they provide an architecture of cloud edge orchestration through artificial intelligence to optimize the resources of the edge. State-of-the-art reviews of AI-based orchestration techniques are provided as well as the main research challenges to be explored in the future. Nevertheless, the authors do not empirically validate their frameworks and do not completely take into account the limitations of the resources of the aggregator node. Moreover, the complexity of AI could generate new security problems that would need to be investigated further.

The paper [27] examines the increasing demand for an efficient way to store and query data in IoTs, especially in the case of spatiotemporal joins that connect data from different sources created in the same location and time. The authors propose a novel method that divides the three-dimensional space-time in equal sized cells and maintain the information about the data from the IoT devices contained in each cell. When a spatiotemporal join is requested, the method identify the appropriate cells and retrieve only the required data thus minimizing the computational costs of the operation with respect to traditional approaches. The experimental results on a real IoT dataset demonstrate that the proposed method is faster than the other known methods to execute spatiotemporal joins. However, this study may not adequately consider the scalability problem of the solution in dynamic environment where the size of the data and the behavior of the IoT devices are changing over time.

The paper [28] present the integration between Internet of Things (IoT), Cloud Computing (CC) and Smart City (SC) technologies to develop a Smart City, where the inefficiencies of traditional urban systems can be addressed. The authors, therefore, analyze the most important technologies and a SC framework to enhance the architectural system, application design, network transmission and sensor integration. A major part of the work has been focused to solve the communication challenges in unreliable environments, particularly regarding the retransmission of data from sensors after transmission failures. The research introduces a data aggregation algorithm based on Markov chains to optimize retransmission processes. Experimental results demonstrate that the proposed system effectively facilitates information sharing and fusion across various sensing subsystems, overcoming previous issues of information silos and meeting the needs of modern smart cities. However, the approach may face limitations in scalability and adaptability in highly dynamic urban environments.

The paper [29] addresses the diverse networking needs of Industrial IoT applications, particularly those requiring timely updates, such as emergency alerts and surveillance systems. The ability of networks to reconfigure at high speeds is critical in terms of responding to events to continue delivering the latest packets. Most of the solutions available today use software-defined networking (SDN), and the openFlow protocol to manage packet latency and congestion, however, most are highly computationally intensive and are designed as offline solutions to overcome this limitation, authors propose an online algorithm using an SDN controller with a global view of the network. Specifically, the paper proposes a control-policy framework called TSNu, which reserves a specific time slot for each packet and resolves issues related to congestion. The TSNu control policy provides a beautiful solution to the problem of maximizing the total utility for scheduling, routing and admission control to improve network stability and admission of flows into the network. The experimental results provide significant performance improvement over current policy and support the claim that the proposed TSNu architecture is effective.

3. Problem definition

With the rapid growth of IoT device usage (smart sensors, connected appliances ...) there has been a significant amount of data created from an increasing number of sources. One of the main issues with this is processing and storing all of the data in an efficient manner. A typical method for storing data is to send it to a server located remotely. This is typically not a viable option for timely response in IoT devices. For example, In applications where time is critical and minor delays could lead to catastrophic results (driverless vehicles making decisions in

seconds, medical equipment that needs to monitor patient vitals in real-time, smart city network traffic management systems) sending data to servers to be processed is not a viable solution. The primary concern with using IoT devices is their power consumption since they have limited battery life. Sending large amounts of data over a long distance to a server consumes a lot of energy which reduces the operational lifetime of the devices and adds additional maintenance costs associated with frequent battery replacements/recharging.

Poor computational resources and storage capacity further limit the capability of IoT devices to conduct local data processing. Lacking sufficient computational resources, these devices cannot accomplish key tasks of data processing and analysis, therefore, these operations must be offloaded to nearby servers.

The solution to the listed issues requires using lower energy in data processing in real time and less latency, this has been adressed successfuly by the use of aggregator nodes to data sources for better storage and processing, this keeps the data at the edge, thus lowering transmission distance and helping reduce latency and energy use.

Memory and battery constraints present major challenges to IoT device performance and overall network performance and thus classify IoT devices into two categories: those with high-end features (memory and battery) and those that are likely to experience performance decline as a result of low-end features. Thus, the characterization of memory and battery constraints will allow for better understanding of the network and thus better development of resource management strategies that utilize this characterization. High-end and limited resources help develop an intelligent decision-making strategy to optimize data offloading and power conservation within IoT systems, contributing to greater reliability and longer product life cycles. Clustering occurs at the base station prior to deploying the IoT network, therefore, clustering is an offline process.

Predictive data managements techniques will enable data to be processed in an efficient way and allocate resources accordingly. IoT devices are often challenged regarding monitoring of memory fill time and battery life with fluctuating data generation rates under changing environmental conditions. Traditional management methods usually do not consider these temporal dependencies, resulting in inaccurate predictions and inefficient resource utilization. With the use of predictive techniques, such time-dependent metrics can be modeled effectively to classify IoT devices into distinct categories based on their foreseen performance.The clustering phase is performed at the base station , this is real time phase.

The integration of these techniques , clustering for device grouping and predictive techniques aims to enable an efficient energy consumption and data storage and processing for IoT networks.

4. Proposed Approach

4.1. IoT devices characteristics

Determining the capacity of an IoT depends on several factors. In this sub-section, we will discuss various dimensions that define the capacity of an IoT device, including data storage, data generation rate, power consumption, communication, processing, and measurement range.

Data storage capacity: An IoT device has local storage (e.g., flash memory), its capacity refers to the amount of data it can hold. This is typically determined by the device's memory specification (e.g., 512MB, 1GB, etc.).

Data generation rate (Throughput): The data generation rate is measured based on the device's sampling rate and the size of each data sample. Its formula is given as below:

$$\text{Throughput} = \text{Sampling rate} \times \text{Data size per sample} \tag{1}$$

Battery capacity (Power consumption): Battery-powered IoT devices have limited operational capacity based on battery life. To estimate the battery life, we use the following formula:

$$\text{Battery life} = \frac{\text{Battery capacity (mAh)}}{\text{Power consumption (mA)}} \tag{2}$$

If we consider that an IoT device consumes an average of 10mA and has a 2000mAh battery, its battery life as below:

$$\text{Battery life} = \frac{2000\,\text{mAh}}{10\,\text{mA}} = 200\,\text{hours} \tag{3}$$

Communication capacity (Network bandwidth): Communication capacity refers to the device's ability to transmit data over a network. This depends on the bandwidth (e.g., 1 Mbps, 10 Mbps) and the protocol overhead, which may reduce the effective data rate.

Processing capacity: Some IoT devices have onboard processing capabilities. The processing capacity can be assessed by the processor speed (in MHz or GHz) and the available RAM.

Coverage range: Coverage range is the maximum spatial distance or area over which an IoT device can reliably operate. The sensing range refers to the maximum distance from the sensor at which it can successfully sense the phenomenon, the communication range defines the maximum distance between two nodes at which they can exchange data using a wireless link without degradation of the signal quality. The performance characteristics of IoT devices are presented in Table 1. Each characteristic has been listed along with whether higher or lower values are preferable, and the reasoning behind this preference. Understanding these characteristics will help in developing and selecting appropriate sensors and actuators for a particular application (e.g., continuous monitoring, energy efficiency, precision).

Table 1. Optimal values for IoT device factors.

Factor	Optimal value	Explanation
Data storage capacity (S_i)	Larger is better	More storage reduces the need for frequent transmissions.
Data generation rate (G_i)	Smaller is better	Lower generation rate saves energy.
Battery capacity (B_i)	Larger is better (mAh)	Extends the device's operational time.
Power consumption (P_i)	Smaller is better (mA)	Increases battery life and reduces recharging.
Communication capacity (C_i)	Larger is better	Allows faster transmission, though may increase power usage.
Processing capacity (Pr_i)	Larger is better	Enables local computations
Range or coverage capacity (R_i)	Larger is better	Enhances versatility across environments and applications.

4.2. Memory fill and battery life

A. Memory fill time considering processing and communication capacity :

The memory fill time is an estimate of how long it will take for an IoT device to fill its available memory, depending on the current data generation rate. This value may be increased or delayed due to additional tasks such as data processing and/or communication that reduces the need to store data locally. For this, IoT devices with higher processing capacity can handle the data through processing or compressing before storage, hence, a reduced amount of data is being written to the memory. Another factor is that the communication capacity of the device contributes to offloading data to external devices, preventing filling up the memories rapidly.
Given the equation for memory fill time is as follows:

$$T_{memory} = \frac{S_i}{G_i \times (1 - Pr_i \times f_{Pr})} \tag{4}$$

Where:

- S_i is the memory storage capacity of IoT device i (in bytes),
- G_i is the data generation rate of device i (in bytes/second),
- Pr_i is the processing capacity factor, representing how effectively the IoT device reduces data through local processing,

- f_{Pr} is the processing efficiency factor, which quantifies how much data reduction or compression occurs before storing it in memory.

Thus, the term $(1 - Pr_i \times f_{Pr})$ captures the reduction in memory usage due to local data processing. An IoT device with no local processing capability will have $Pr_i = 0$, and the memory fill time would depend entirely on the raw data generation rate.

B. Battery life considering communication, processing, and coverage capacity:

The length of time an IoT device battery lasts is impacted by how much energy is used for each of the many things that it does such as communicate, process data and cover its physical area. All of those items require a certain amount of energy and therefore reduce the overall energy (battery) in the device. If the IoT device has a greater communication capacity then there is going to be increased energy use due to the fact that the device is sending out more data or communicating with other devices more often. Likewise if the IoT device has a greater processing capacity there is going to be increased energy consumption due to the fact that the IoT device is able to perform more calculations locally. Finally, an IoT device which has a greater range or capability of coverage will have greater energy needs to cover a wider physical area and/or gather data at longer distances.

We determine the equation for battery life as follows:

$$T_{battery} = \frac{B_i}{P_i + C_i \times f_C + Pr_i \times f_{Pr} + R_i \times f_R} \tag{5}$$

Where:

- B_i is the the battery capacity of IoT device i (in mAh),
- P_i is the base power consumption of the IoT device in standby mode (in mA),
- C_i is the communication capacity factor, which increases energy consumption based on communication frequency and bandwidth,
- f_C is the communication energy factor, representing the energy used during data transmission,
- Pr_i is the processing capacity factor, which accounts for the energy consumed by onboard processing,
- f_{Pr} is the processing energy factor, representing the energy used per unit of data processed,
- R_i is the range or coverage factor, representing how the IoT device's energy consumption increases with a wider measurement range,
- f_R is the range energy factor, which quantifies the additional energy used to monitor or cover larger areas.

The equation (5) above is the interplay between communication, processing and range activities, which is used to determine power consumption that's directly impacts the battery life of the IoT device. Managing these factors leads to the high performance that could be delivered by the batteries, especially in environments where energy resources are limited.

4.3. IoT devices normalisation and clustering

In this study, we are proposing an approach that will divide all IoT devices into 2 different categories, based on their *Memory Fill Time* (T_{memory}), and *Battery Life* ($T_{battery}$) characteristics by utilizing the *k-means* clustering model. The first category includes those IoT devices which have large values for T_{memory} and high $T_{battery}$, whereas the second category is comprised of those IoT devices which have small values for Tmemory and small values for Tbatttery. Before using our k-means model, we will apply a min-max normalization method to ensure that all features have an equal weight in clustering. This new methodology will also enable us to classify the different features of an IoT device much better which will allow us to better manage our resources and better understand how IoT devices are working. **A. Normalization using Min-Max scaling**

To ensure that T_{memory} and $T_{battery}$ are on the same scale, we begin by applying min-max normalization. This method adjusts both features to fall within the range [0,1], ensuring that no single feature disproportionately influences the clustering process. The min-max normalization for a given feature X is expressed as:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{6}$$

Where:

- X is the original value of either T_{memory} or $T_{battery}$,
- X_{min} and X_{max} are the minimum and maximum values of the feature.
- X' is the normalized value.

By performing this normalization step, we ensure that both features have equal importance when applying the k-means algorithm.

B. K-means clustering algorithm

We employ the *k-means* algorithm to partition the IoT devices into two clusters based on their T_{memory} and $T_{battery}$ values. The algorithm minimizes within-cluster variance by iteratively adjusting cluster centroids. In this study, we set $k = 2$ to classify the IoT devices into two distinct groups: one with high T_{memory} and $T_{battery}$, and the other with the inverse values.

4.4. Steps of the K-means algorithm

1. Initialization: We choose $k = 2$, as our goal is to classify IoT devices into two groups. Two initial centroids are randomly selected from the normalized dataset, representing the center of each cluster.

2. Assignment step: We compute the Euclidean distance between each data point and the centroids. The Euclidean distance between a data point i and a centroid c is given by:

$$d = \sqrt{(T'_{memory,i} - c_{T_{memory}})^2 + (T'_{battery,i} - c_{T_{battery}})^2} \tag{7}$$

Where:

- $T'_{memory,i}$ and $T'_{battery,i}$ are the normalized values of data point i,
- $c_{T_{memory}}$ and $c_{T_{battery}}$ are the coordinates of the centroid.

Each data point is then assigned to the cluster whose centroid is closest.

3. Update step: We update the centroids by calculating the mean of the T_{memory} and $T_{battery}$ values for all points assigned to each cluster. The new centroids c_k for cluster k are computed as:

$$c_{T_{memory}} = \frac{1}{N_k} \sum_{i \in C_k} T'_{memory,i} \tag{8}$$

$$c_{T_{battery}} = \frac{1}{N_k} \sum_{i \in C_k} T'_{battery,i} \tag{9}$$

Where:

- N_k is the number of data points in cluster k,
- C_k is the set of data points in cluster k.

4. Repeat: We repeat the assignment and update steps until the centroids no longer change significantly, indicating that the clusters have converged.

5. Group characteristics: Once convergence is achieved, we obtain two clusters:

- **Group 1 (Unconstrained IoT devices):** IoT devices characterized by high T_{memory} and high $T_{battery}$.
- **Group 2 (Constrained IoT devices):** IoT devices exhibiting the inverse characteristics, meaning at least one of T_{memory} or $T_{battery}$ is low. However, it is important to note that Group 2 is defined relative to Group 1 and does not necessarily imply that both features are low, instead, it captures the opposite behavior of the IoT devices compared to those in Group 1.

In this study, we proposed a way to classify IoT devices using the two most important performance measures *Memory Fill Time* and *Battery Life* using min max normalization and K-Means Clustering Algorithm so as to be able to cluster the IoT devices in two groups, thereby being able to find the IoT devices that have high and low performances in both memory and battery life. The advantage of this is not only a systematic way to cluster the IoT devices, but also a good insight to optimize IoT networks where there are limited resources.

4.5. Offloading mechanism for constrained IoT devices to aggregator nodes

In this part, we propose an offloading mechanism in which IoT devices classified in Group 2 characterized by constrained *Memory Fill Time* (T_{memory}) and *Battery Life* ($T_{battery}$) , and offload their data to aggregator nodes. The aggregator node, equipped with substantially greater computational power, storage capacity, and energy resources, manages both data storage and processing for the constrained IoT devices. By offloading these tasks, it reduces the strain on the limited resources of Group 2 IoT devices, and it is a strategic step toward maximizing energy consumption and improving the overall efficiency of the IoT system.

4.5.1. Offloading process

For IoT devices in Group 2, which have low memory and short battery life, minimizing local data storage and on-device computation must be kept low in order to preserve energy. According to that, we recommend the following data offloading approach.

1. **Data collection and minimal local processing**: IoT devices in group 2 collect raw data, as environmental conditions, temperature, and motion. In order to extend the battery life basic pre-processing such as noise reduction, or light compression is performed, to shrink the data to be transmitted.
2. **Data transmission to the aggregator nodes**: Using energy efficient wireless communication protocols such as LoRa, Zigbee, or Wifi, the pre-processed data is then transmitted to a nearby aggregator node. The aggregator nodes are placed to minimize transmission energy costs, in order to allow the preservation of battery power to group 2 IoT devices while continuously offloading data.
3. **Key functions of the aggregator nodes:**
   - *Efficient data storage*: the aggregator nodes tend to have much larger storage capacity, which allows group 2 IoT devices to function without storage limitation.
   - *Handling complex computations*: The aggregator nodes are responsible for computationally intensive workloads like data aggregation and feature extraction and thus spare the IoT devices from overtaxed computing resources.
   - *Providing real-time feedback*: After processing, the aggregator nodes are able to send signals to the IoT devices with control commands or insights, which increases energy efficiency and improves overall system efficiency.

4.5.2. Key improvements of the offloading mechanism

The Group 2 IoT devices transfer data to the aggregator nodes for processing within the offload mechanism offering various advantages:

Energy savings:

- *Reduced on-device processing*: When aggregator nodes process the device's task (i.e., data processing), it is no longer required by the IoT device, this lowers the amount of computation that the IoT device has to perform.
- *Lower communication overhead*: The amount of energy an aggregator node uses to communicate with other nodes is significantly lower than if the IoT device was to communicate directly with another node that may be farther away, thus reducing the amount of power consumed by the IoT device.

Optimized data storage:

- *Extended local memory usage*: Group 2 IoT devices can regularly offload data to the aggregator node, avoiding the risk of memory overflow and enabling continuous operation.
- *Data management*: Advanced data management techniques (e.g., compression, deduplication of storage) maximize storage and decrease redundancies in data management within the aggregator node.

Enhanced system efficiency:

- *Scalability*:aggregator node computing can manage data from multiple IoT devices, which allows to the system to scale without overwhelming IoT device resources.
- *Real-time Decision Making*: aggregator nodes have the ability to process data in a manner that reduces latency and allows faster feedback mechanisms to the IoT devices.

Local analytics and reduced latency: Offloading to nearby aggregator nodes improves processing speed and response time for applicable real-time mechanisms. Localized processing reduces reliance on cloud infrastructure and strength the IoT device networks as needed.

4.6. Predicting T_{memory} and $T_{battery}$ using LSTM and classifying with KNN

In this sub-section, we outline our approach to predicting the values of *Memory Fill Time (T_{memory})* and *Battery Life ($T_{battery}$)* for IoT devices using a *Long Short-Term Memory* network in a multi-output regression setting. Following the prediction, we employ the *K-Nearest Neighbors* algorithm to classify each device into either *Group 1* (high T_{memory} and $T_{battery}$) or *Group 2* (at least one of T_{memory} or $T_{battery}$ is low) based on the predicted values.

4.6.1. Motivation for using LSTM

LSTM is a new type of Recurrent Neural Network (RNN) that was developed with the intent of capturing long term sequential relationships in data. The majority of IoT device parameters have some temporal dependency, T_{memory} and $T_{battery}$ represent two examples of this, as many times the rate of change of the amount of data generated by an IoT device will affect memory and the amount of power consumed by the device will also change over time. The ability to predict future values for these types of parameters allows the user to better understand how their IoT devices will behave over time and can provide a level of accuracy that would be difficult to obtain through other means.

4.6.2. LSTM for multi-output prediction

We define the prediction of T_{memory} and $T_{battery}$ as a *multi-output regression problem*. The objective is to develop a single LSTM model capable of predicting both outputs simultaneously, while accounting for the time-dependent patterns present in the device data.

Steps for LSTM implementation

- *Data preparation:* The Time-Series Data is Collected from IoT Devices: Available Storage Capacity (Storage), Rate of Data Generation (Generation Rate), Base Power Consumption (Base Power), Level of Communication Activity/Capacity (Communication), Local Processing Capacity (Processing), Sensing Coverage Range of IoT Device (Sensing), and the Processing Efficiency Coefficient, Communication Efficiency Coefficient, and Coverage Efficiency Coefficient.

- *LSTM network architecture:* We build an LSTM model with a sequence of inputs from devices. The model contains multiple LSTM layers followed by a dense layer in order to produce output for each of the two targets simultaneously:

$$\hat{T}_{memory}, \hat{T}_{battery} = \text{LSTM}(X) \tag{10}$$

  Where X represents the input features as a sequence, and $\hat{T}_{memory}$ and $\hat{T}_{battery}$ represent the predictions for memory fill time and battery life, respectively.

- *Training process:* We train the LSTM model using MSE loss, which computes a loss value for both predictions.

$$\begin{aligned}\text{Loss} = \frac{1}{n}\sum_{i=1}^{n}((T_{memory,i} - \hat{T}_{memory,i})^2 \\ + (T_{battery,i} - \hat{T}_{battery,i})^2)\end{aligned} \tag{11}$$

  The model is trained on historical device data to learn the temporal dependencies between input features and the target values.

- *Prediction:* Once trained, the LSTM model is used to predict T_{memory} and $T_{battery}$ for new device data, providing real-time estimates of both metrics for each device.

4.6.3. Classification using K-Nearest neighbors

Having obtained the predicted values of T_{memory} and $T_{battery}$ with LSTM, we now classify each device into *Group 1* or *Group 2* by using these predicted values. To this end, a simple yet effective classification algorithm, called *K-Nearest Neighbors*, is employed, where the label of a given data point is determined by the majority class of its nearest neighbors.

Steps for KNN classification

- *Feature selection:* The predicted values $\hat{T}_{memory}$ and $\hat{T}_{battery}$ are used as input features for the KNN classifier. Each device is represented by the pair $(\hat{T}_{memory}, \hat{T}_{battery})$.

- *Labeling the training Data:* The training dataset for KNN consists of historical devices, where each device is labeled as either *Group 1* (high T_{memory} and $T_{battery}$) or *Group 2* (at least one of T_{memory} or $T_{battery}$ is low).

- *KNN algorithm:* For a new IoT device with predicted values $\hat{T}_{memory}$ and $\hat{T}_{battery}$, we compute the Euclidean distance to all labeled devices in the training dataset. The KNN algorithm then selects the *k-nearest neighbors* (with k typically chosen through cross-validation) and assigns the class based on the majority label of the neighbors.

- *Classification decision:* If the majority of the nearest neighbors are from Group 1, the new IoT device is classified into Group 1. Otherwise, it is classified into Group 2.

5. Simulation and evaluation

5.1. Simulation configuration of IoT sensors for aggregator node data offloading

To evaluate the proposed offloading strategy, we modeled a wireless sensor network (WSN) composed of 1000 IoT sensors and 50 aggregator nodes using the ns-3 simulation environment [30]. Each sensor node was parameterized with the key factors required by the analytical models of Section IV, covering both storage and energy constraints as well as the efficiency coefficients involved in the formulas of T_{memory} and T_{battery}.

During a three-hour experiment, each sensor offloaded approximately 140 KB to 180 KB. Table 2 summarizes the configuration adopted in the simulation.

Table 2. Configuration of IoT sensors for the ns-3 simulation.

Parameter	Value
Number of sensors	1000
Data generation rate	20 bytes/s
Storage capacity	5 MB
Initial battery capacity	540 abstract energy units
Base power consumption	0.03 units/s
Transmission power consumption	0.02 units/s
Communication capacity	50 kbps
Communication protocol	LoRa
Processing capacity	16 MHz MCU
Sensing/coverage range	10 m
Simulation period	3 hours
Processing efficiency factor (f_{Pr})	0.25
Communication energy factor (f_C)	0.15
Range/coverage energy factor (f_R)	0.10

5.2. Hyperparameter Settings for LSTM and KNN

To predict the memory–fill time (T_{memory}) and battery life (T_{battery}) of the IoT sensors, a LSTM was implemented, followed by a KNN classifier to assign each device to either the constrained or unconstrained group defined in Section IV. The LSTM captures temporal dependencies in the time–series data produced by the ns-3 simulation, while KNN performs a final classification based on the predicted values of T_{memory} and T_{battery}.

The dataset generated by the three-hour simulation contains 216 000 time–stamped records with nine representative input features (e.g., data generation rate, communication frequency, base power consumption, transmission power, processing load, ambient temperature, humidity, coverage range and residual battery level). All features are normalized using min–max scaling to ensure numerical stability.Table 3 summarizes the hyperparameters adopted for both the LSTM model and the KNN classifier.

This configuration allows the LSTM to capture the temporal behaviour of memory and battery dynamics, while the KNN classifier accurately distinguishes between constrained and unconstrained IoT devices based on the predicted values.

5.3. Results and discussion

To investigate the effectiveness of our approach, we have made a comparison among the outcomes from our methodology against the approaches in [24] and [25]. The IoT device residual energy, device failures, and average available memory space were the three critical performance metrics.The two baseline methods were selected because they represent state–of–the–art resource management strategies with comparable scope and assumptions to our scenario.

In the Figure 1, we compare our approach with two other approachs Liu et al. and Udayaprasad et al. in terms of average residual energy over elapsed times (T1 to T10) of 1000 sensors. Our approach shows a more gradual decrease in residual energy, indicating better energy efficiency and suggesting a longer network lifetime compared

Table 3. Configuration of the LSTM model and KNN classifier.

Parameter	Value
Number of time-stamped samples	216 000
Number of input features	9
Normalization method	Min–max scaling
LSTM layers	2 stacked hidden layers
Hidden units per layer	64 (first) / 32 (second)
Dropout rate	0.2
Output layer	2 linear neurons ($\hat{T}_{\text{memory}}$ and $\hat{T}_{\text{battery}}$)
Loss function	Mean Squared Error (MSE)
Optimizer	Adam (learning rate = 0.001)
Batch size	32
Training epochs (max)	50
Early stopping	Patience = 10 epochs
Data split	70% training / 15% validation / 15% test
KNN number of neighbors (k)	5
KNN distance metric	Euclidean distance

to the others. By T10, our approach retains the highest residual energy, highlighting enhanced energy management. Udayaprasad et al.'s approach performs moderately, with a faster decline than ours but still better than Liu et al., which shows the steepest decline. This pattern suggests that our approach is better suited for energy-constrained applications, where maintaining sensors longevity is crucial

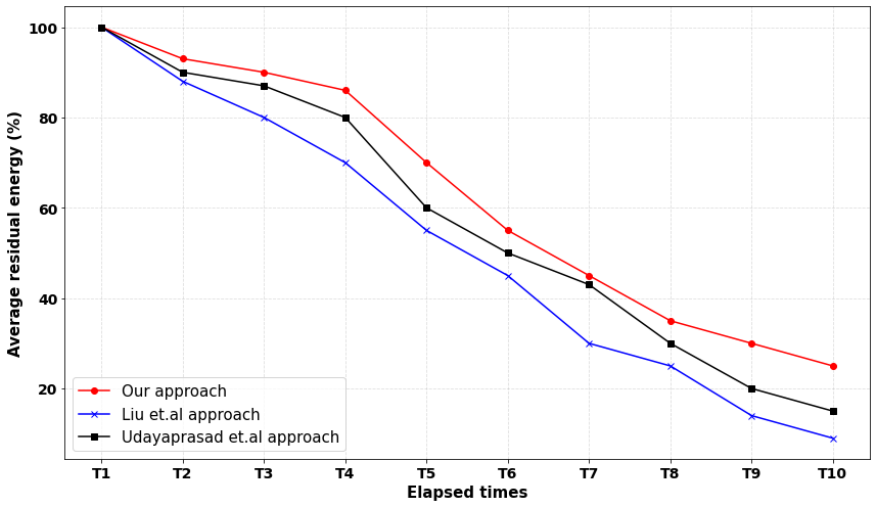

Figure 1. Comparison of average residual energy among different approaches over elapsed time.

The number of failed devices over time (from T1 to T10) was compared among the three approaches shown in Figure 2. The number of failed IoT devices over time indicates longer lived and reliable IoT devices when using our approach, resulted in fewer failed IoT devices over time compared to both of the other two approaches. A rapid rise in failed IoT devices using the Liu et al. method occurred after T3 and reached the highest failure rate by T10, demonstrating less efficient energy and resource management. Udayaprasad et al. method performed moderately between the two extremes, there were more failures with the Udayaprasad et al. method than with our method, but fewer failures than with the Liu et al. method. As a result, our approach is superior to the two other approaches for maintaining the functionality of an IoT device over time, which is important for maintaining the operational lifetime of the network.

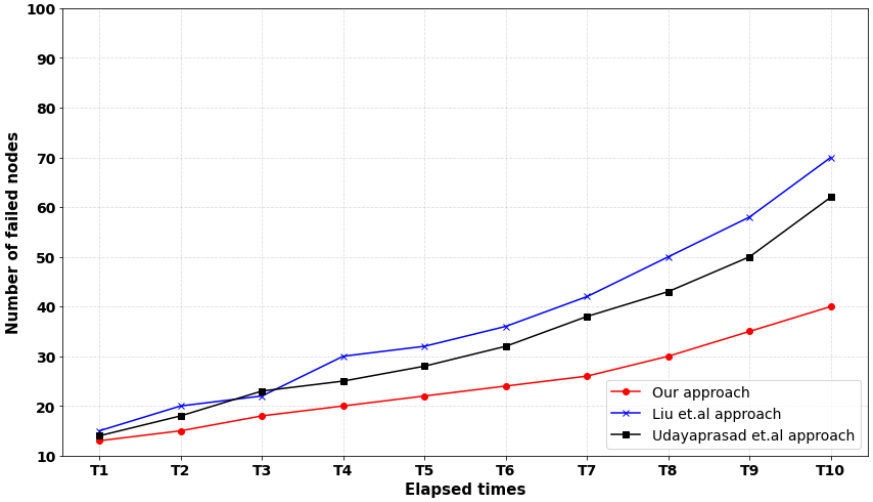

Figure 2. Comparison of the number of failed devices among different approaches over elapsed Time.

To analyze how well data are managed, we examine the variation in average available space (%), over time (Figure 3) among three different approaches: our approach, Liu et al. approach, and Udayaprasad et al. approach. We see that our approach began with the largest amount of available space (about 86%), and decreased gradually until about 30% of available space at T10, indicating good resource efficiency. Conversely, the Liu et al. method had a steeper decrease in available space, and stabilized at about 15%. Finally, the Udayaprasad et al. method began with less available space, and ended with less available space than either of the first two methods. These results indicate that our approach is more suitable for use in applications where available space is a priority.

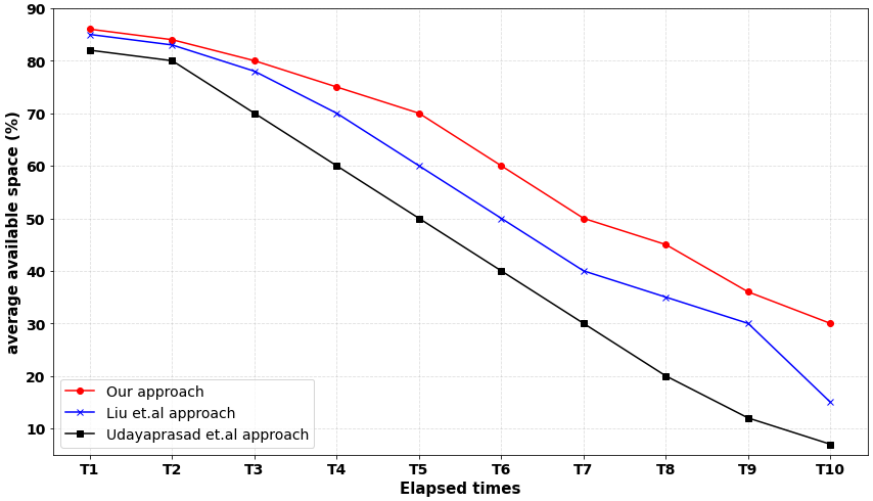

Figure 3. Comparison of average available space among different approaches over elapsed time.

We evaluate the average data offloaded to aggregator nodes of our approach, measured in kilobytes (KB), across ten different sets as illustrated in Figure 4, where each set represents 100 sensors. Most sets show a consistent offloading range between 14000 KB and 18000 KB. However, it appears the greater part of the created data was successfully transmitted to the aggregator nodes. Set 7 demonstrates the greatest volume of offload at 17999 KB, therefore, it represents a highly effective use of resources as well as an ideal balance of communication. On the

other hand, Set 5 is the least amount of offload achieved (approximately 13800 KB) suggests there may be slightly less than optimal operating conditions for the subset of these sensors. Overall, the offload behavior is relatively consistent across all sets and provides a solid demonstration of the reliability of the proposed method's ability to perform efficiently with respect to the transfer of large amounts of data in IoT sensor networks.

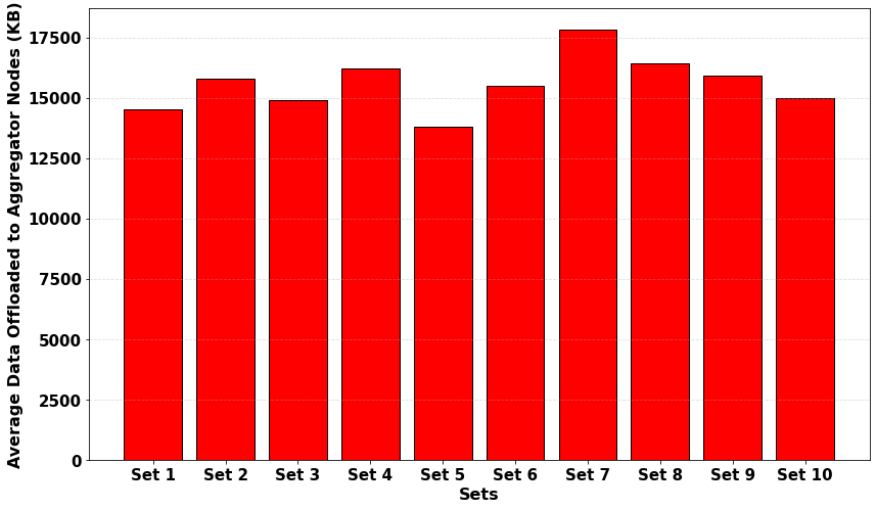

Figure 4. Data offloaded to aggregator nodes across different sets of IoT sensors (measured in KB).

6. Conclusion and future work

In conclusion, this study addresses the challenges of managing data and energy consumption in resource-constrained IoT devices.By creating an innovative offload procedure, we enable the efficient transfer of data from IoT devices to aggregator nodes. Consequently, this enables the enhanced processing and storage capability of aggregator nodes. We have integrated the use of k-means clustering to group IoT devices by their memory fill time and battery life, as well as the application of a predictive model using LSTM and KNN to create new and advanced strategies to manage these devices in order to maximize the efficiency of the energy used. Results from simulations show that there has been a significant decrease in failure rates of IoT devices and an increase in available storage capacity. These results confirm that the methodological approach presented here can be effectively utilized in practical applications. In addition, the research provides a contribution to the development of more reliable and sustainable IoT networks. Additionally, the research will provide the necessary foundational work for continued advancements in ad-hoc networks.

In the future, we intend to investigate the collection of energy-efficient data in Wireless Sensor Networks utilizing Unmanned Aerial Vehicles (UAV). Specifically, we aim to utilize the mobility of UAVs and the intelligent trajectory planning to optimize energy consumption during the collection of large amounts of data.

REFERENCES

1. S. Wijethilaka and M. Liyanage, "Survey on Network Slicing for Internet of Things Realization in 5G Networks," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 957–994, Secondquarter 2021. doi: 10.1109/COMST.2021.3067807.
2. M. Lombardi, F. Pascale and D. Santaniello, "Internet of Things: A General Overview between Architectures, Protocols and Applications," *Information*, vol. 12, p. 87, 2021. doi: 10.3390/info12020087.
3. M. Mazhar Rathore, A. Paul, W. H. Hong, H. Seo, I. Awan and S. Saeed, "Exploiting IoT and Big Data Analytics: Defining Smart Digital City Using Real-Time Urban Data," *Sustainable Cities and Society*, vol. 40, pp. 600–610, 2018. doi: 10.1016/j.scs.2017.12.022.

4. M. Ahmad Jan, M. Adil, B. Brik, S. Harous and S. Abbas, "Making Sense of Big Data in Intelligent Transportation Systems: Current Trends, Challenges and Future Directions," *ACM Computing Surveys*, vol. 57, no. 8, August 2025. doi: 10.1145/3716371.
5. R. K. Singh, R. Berkvens and M. Weyn, "AgriFusion: An Architecture for IoT and Emerging Technologies Based on a Precision Agriculture Survey," *IEEE Access*, vol. 9, pp. 136253–136283, 2021. doi: 10.1109/ACCESS.2021.3116814.
6. S. M. Khan, I. Shafi, W. H. Butt, I. de la Torre Diez, M. A. López Flores, J. C. Galán and I. Ashraf, "A Systematic Review of Disaster Management Systems: Approaches, Challenges, and Future Directions," *Land*, vol. 12, no. 8, p. 1514, 2023. doi: 10.3390/land12081514.
7. P. Mishra and G. Singh, "Energy Management Systems in Sustainable Smart Cities Based on the Internet of Energy: A Technical Review," *Energies*, vol. 16, p. 6903, 2023. doi: 10.3390/en16196903.
8. T. Sanislav, G. D. Mois, S. Zeadally and S. C. Folea, "Energy Harvesting Techniques for Internet of Things (IoT)," *IEEE Access*, vol. 9, pp. 39530–39549, 2021. doi: 10.1109/ACCESS.2021.3064066.
9. M. Eskandarpour and H. Soleimani, "Enhancing Lifetime and Reliability in WSNs: Complementary of Dual-Battery Systems Energy Management Strategy," *International Journal of Distributed Sensor Networks*, vol. 2025, no. 1, p. 5870686, 2025. doi: 10.1155/dsn/5870686.
10. M. Muzammal Islam, et al., "Improving Reliability and Stability of the Power Systems: A Comprehensive Review on the Role of Energy Storage Systems to Enhance Flexibility," *IEEE Access*, vol. 12, pp. 152738–152765, 2024. doi: 10.1109/ACCESS.2024.3476959.
11. A. Marouan, M. Badrani, N. Kannouf, A. Zannou and A. Chetouani, "Blockchain-based e-voting system in a university," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 3, pp. 1915–1923, 2024. doi: 10.11591/ijeecs.v34.i3.pp1915-1923.
12. K. Jain, A. Kumar and A. Singh, "Data transmission reduction techniques for improving network lifetime in wireless sensor networks: An up-to-date survey from 2017 to 2022," *Transactions on Emerging Telecommunications Technologies*, vol. 34, no. 1, p. e4674, 2023. doi: 10.1002/ett.4674.
13. P. Singh and R. Vir, "Enhanced energy-aware routing protocol with mobile sink optimization for wireless sensor networks," *Computer Networks*, vol. 261, p. 111100, 2025. doi: 10.1016/j.comnet.2025.111100.
14. D. K. Sah, S. Srivastava, R. Kumar, et al., "An energy efficient coverage aware algorithm in energy harvesting wireless sensor networks," *Wireless Networks*, vol. 29, pp. 1175–1195, 2023. doi: 10.1007/s11276-022-03125-3.
15. M. N. Khan, et al., "Energy-Efficient Dynamic and Adaptive State-Based Scheduling (EDASS) Scheme for Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 22, no. 12, pp. 12386–12403, June 15, 2022. doi: 10.1109/JSEN.2022.3174050.
16. A. Zannou, A. Boulaalam and E.H. Nfaoui, "Data Flow Optimization in the Internet of Things," *Statistics, Optimization & Information Computing*, vol. 10, no. 1, pp. 93–106, 2022. doi: 10.19139/soic-2310-5070-1166.
17. Y. Gong, H. Yao, J. Wang, M. Li and S. Guo, "Edge Intelligence-Driven Joint Offloading and Resource Allocation for Future 6G Industrial Internet of Things," *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 6, pp. 5644–5655, Nov.–Dec. 2024. doi: 10.1109/TNSE.2022.3141728.
18. R. Mustafa, N. I. Sarkar, M. Mohaghegh and S. Pervez, "A Cross-Layer Secure and Energy-Efficient Framework for the Internet of Things: A Comprehensive Survey," *Sensors*, vol. 24, no. 22, p. 7209, Nov. 11, 2024. doi: 10.3390/s24227209.
19. C. Kanzouai, S. Bouarourou, A. Zannou, A. Boulaalam and E.H. Nfaoui, "Enhancing IoT Scalability and Interoperability Through Ontology Alignment and FedProx," *Future Internet*, vol. 17, no. 140, 2025. doi: 10.3390/fi17040140.
20. A. Khalifeh, K. A. Darabkh, A. M. Khasawneh, I. Alqaisieh, M. Salameh, A. AlAbdala, S. Alrubaye, A. Alassaf, S. Al-HajAli, R. Al-Wardat, et al., "Wireless Sensor Networks for Smart Cities: Network Design, Implementation and Performance Evaluation," *Electronics*, vol. 10, p. 218, 2021. doi: 10.3390/electronics10020218.
21. K. K. Almuzaini, S. Joshi, S. Ojo, et al., "Optimization of the operational state's routing for mobile wireless sensor networks," *Wireless Networks*, vol. 30, pp. 5247–5261, 2024. doi: 10.1007/s11276-023-03246-3.
22. A. A. Qaffas, "Applying an Improved Squirrel Search Algorithm (ISSA) for Clustering and Low-Energy Routing in Wireless Sensor Networks (WSNs)," *Mobile Networks and Applications*, vol. 29, pp. 1753–1768, 2024. doi: 10.1007/s11036-023-02219-2.
23. Z. Hai-yu, "An In-depth Analysis of Uneven Clustering Techniques in Wireless Sensor Networks," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 3, 2023. doi: 10.14569/IJACSA.2023.0140381.
24. H. Liu, D. Cui, Q. Ma, *et al.*, "High-Quality and Energy-Efficient Sensory Data Collection for IoT Systems," *Arab. J. Sci. Eng.*, vol. 50, pp. 7245–7260, 2025, doi: 10.1007/s13369-024-09364-0.
25. P. K. Udayaprasad *et al.*, "Energy Efficient Optimized Routing Technique With Distributed SDN-AI to Large Scale I-IoT Networks," *IEEE Access*, vol. 12, pp. 2742–2759, 2024, doi: 10.1109/ACCESS.2023.3346679.
26. Y. Wu, "Cloud-Edge Orchestration for the Internet of Things: Architecture and AI-Powered Data Processing," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12792-12805, 15 Aug. 2021, doi: 10.1109/JIOT.2020.3014845. Keywords: Cloud computing; Servers; Computer architecture; Internet of Things; Edge computing; Data processing; Medical services; Artificial intelligence (AI); cloud computing; edge computing; Internet of Things (IoT); offloading.
27. K. Y. Lee, M. Seo, R. Lee, M. Park, and S. -H. Lee, "Efficient Processing of Spatio-Temporal Joins on IoT Data," *IEEE Access*, vol. 8, pp. 108371-108386, 2020, doi: 10.1109/ACCESS.2020.3001214. Keywords: Internet of Things; Indexes; Partitioning algorithms; Three-dimensional displays; Time measurement; Query processing; Big Data; Spatio-temporal join; spatio-temporal index; Internet of Things; IoT data.
28. D. Jiang, "The construction of smart city information system based on the Internet of Things and cloud computing," *Computer Communications*, vol. 150, pp. 158-166, 2020. doi: 10.1016/j.comcom.2019.10.035. https://www.sciencedirect.com/science/article/pii/S0140366419312034.
29. V. Balasubramanian, M. Aloqaily, and M. Reisslein, "An SDN architecture for time sensitive industrial IoT," *Computer Networks*, vol. 186, p. 107739, 2021. doi: 10.1016/j.comnet.2020.107739.
30. G.F. Riley and T.R. Henderson, "The ns-3 Network Simulator," *Modeling and Tools for Network Simulation*, Springer, 2010, pp. 15–34, doi: 10.1007/978-3-642-12331-3-2.