OPEN ACCESS
*International Journal of Modern Achievement in Science, Engineering and Technology (IJSET)*
ISSN 3023-459X
DOI: 10.63053/ijset.55

# Data Compression Algorithms for Improving Real-Time Monitoring and Automation in IoT-Enabled Smart Homes

**Ali Oveysikian[1]***

1.PhD student, Department of computer science and engineering, Tarbiat Modares university, Tehran, Iran

## ARTICLE INFO

***Keywords:***
*Data Compression, Internet of Things, Smart home, Data transmission, Energy Consumption*

## ABSTRACT

The rapid proliferation of Internet of Things (IoT)-enabled devices has revolutionized modern smart homes, offering advanced automation, real-time monitoring, and enhanced user convenience. However, this growth has brought forth challenges, particularly concerning the energy consumption of these devices. Smart sensors, as fundamental components of IoT ecosystems, continuously generate vast amounts of data, requiring efficient transmission and processing. The energy-intensive nature of data communication in IoT devices highlights the need for innovative approaches to optimize their energy efficiency without compromising performance. This paper aims to evaluate the impact of data compression on the energy consumption and latency associated with data transmission in smart sensors within smart homes. To achieve this, the performance of various compression algorithms in compressing data generated by sensor nodes is evaluated. This evaluation is conducted with the aim of reducing data volume, improving transmission efficiency, and lowering the energy consumption of communication systems. The experimental results demonstrate that utilizing data compression techniques can significantly contribute to reducing energy consumption. By extending this process to all sensor nodes in smart home systems, a more substantial reduction in energy consumption can be anticipated. Such optimizations pave the way for more sustainable IoT ecosystems, balancing technological advancements with environmental concerns.

## Introduction

In recent years, the role of microcontroller-based devices has expanded dramatically, becoming an integral part of modern human societies. These devices are now widely used in various domains, including smart homes, autonomous vehicles, industrial automation, healthcare, and even wearable technologies. IoT devices typically utilize sensors to measure environmental or operational variables, such as temperature, humidity, motion, or pressure, and then process the collected data through embedded processors. A significant number of these devices are equipped with network connectivity or communication capabilities, which places them within the category of Internet of Things (IoT) devices. IoT devices typically used sensors to measure environmental or operational variables, such as temperature, humidity, motion, or pressure, and then process the collected data through embedded processors. This data is often transmitted to other devices or cloud servers for further analysis, storage, or real-time decision-making. The ability to connect and communicate with other systems allows these devices to be part of larger, integrated networks, enabling seamless automation, data sharing, and remote control. As a result, microcontroller-based IoT devices are playing a crucial role in enhancing efficiency, safety, and convenience in both personal and industrial settings, while also contributing to the growing data-driven ecosystem.

The use of these devices also brings significant challenges. For instance, the continuous 24/7 operation of sensors in various environments like smart homes requires optimized management of energy consumption and bandwidth usage. Additionally, these challenges may negatively impact the lifespan of the equipment, making regular maintenance and replacement necessary. To address these issues, various approaches such as data compression, edge, fog, and cloud computing, along with the optimization of data transmission algorithms, have been explored. Each of these methods can contribute to reducing resource strain and enhancing the overall efficiency of the system.

Data compression significantly impacts energy consumption and the life cycle of IoT devices by optimizing data transmission and extending battery life. However, one of the primary challenges in these networks is the high energy consumption for data transmission, which can account for up to 60% of the total device energy (de Oliveira Júnior et al., 2023). In this way, (Krishnamurthi et al., 2021) emphasize that effective data processing, including techniques like data denoising and aggregation, can significantly reduce the energy required for transmission. By compressing data before it is sent, energy consumption can be minimized, directly contributing to the sustainability of IoT systems. This notion is further supported by (Azar et al., 2019) who highlight the importance of energy-efficient data compression in edge machine learning. Their findings demonstrate that reduced data processing requirements lead to lower energy usage, thereby extending device lifespans. Moreover, the integration of cloud, fog, and edge computing technologies can enhance data processing efficiency. By offloading computationally intensive tasks, as discussed by (Ren et al., 2019), the local energy consumption of IoT devices can be mitigated. This model not only facilitates energy savings but also prolongs the operational life of devices by minimizing wear on components. Similarly, some researchers explore how energy-efficient offloading strategies in smart IoT systems can optimize energy usage, reinforcing the correlation between effective data management and energy conservation (Chen et al., 2022; Fu et al., 2021). Considering these papers, it can be concluded that there is a vast research space for using compression methods in IoT devices. With the expansion and growing demand for smart homes, selecting the best data management approach, particularly in terms of energy optimization and device efficiency, is gaining more attention than ever before. This focus not only leads to reduced operational costs but also contributes to extending device lifespans and enhancing the sustainability of smart systems. Therefore, exploring and developing efficient and innovative compression techniques could be a significant step toward improving the quality of service and the effectiveness of IoT devices in smart environments.

However, while data compression offers numerous benefits, it may introduce computational overhead, potentially affecting real-time processing capabilities in certain applications, which must be carefully managed to maintain system efficiency. By focusing on these foundational aspects, this paper lays the groundwork for a comprehensive exploration of how data management strategies can drive advancements in smart home technology and foster a more sustainable IoT ecosystem.

The rest of this paper is organized as follow. It begins by addressing foundational concepts in data compression, illustrating how different techniques can enhance data management within smart home environments. Subsequently, in section three, the IoT's integration into smart home systems is explored, underscoring its pivotal role in enabling seamless interconnectivity among devices. The literature review then synthesizes relevant studies, offering a broad view of current advancements in section four. The proposed scheme in section five serves as a novel contribution, integrating these insights to present a more efficient and responsive smart home system. The article concludes by consolidating the findings and providing thoughtful insights for future work in smart home technology and IoT development.

## 1. Data Compression

The Data compression significantly reduces the volume of data transferred over telecommunication infrastructure, thereby creating more capacity for IoT-based services. For example, consider a smart grid with 20 million smart meters. If each energy consumption record is 5 KB in size and data is sampled every 10 minutes, the total volume of consumption data over a year will reach approximately 20,000 TB. Additionally, in a smart home with 30 different sensors collecting data on temperature, humidity, motion, air quality, lighting, and energy consumption, if each sensor transmits a 400-byte data sample every 5 minutes, the total data generated over a year would be 1.26 GB. By applying data compression techniques, this volume can be significantly reduced. The performance of data compression methods can be evaluated based on various indicators, such as processing speed, computational complexity, time consumption, and memory consumption (Wen et al., 2018). Among these indicators, the Compression Ratio (CR) is recognized as one of the most critical metrics, as it reflects the extent of data volume reduction while preserving essential information or with minimal loss. A higher compression ratio indicates more efficient compression. The CR is shown in Eq. 1.

$$CR = Original\ Data\ Size / Compressed\ Data\ Size \tag{1}$$

The quality of reconstructed data is also crucial; the data must remain accurate and usable after decompression, as poor reconstruction quality can lead to significant information loss or create deviations in analysis. Additionally, the speed and precision of compression methods can vary significantly depending on the data type and intended application. Some methods may use less memory but have slower speeds, while others operate quickly at the cost of higher resource consumption. Overall, selecting an appropriate compression method requires a comprehensive assessment of speed, accuracy, compression ratio, and data reconstruction quality to ensure optimal performance for each application. Compression Gain (CG) refers to the reduction in the size or volume of data after compression, as described in Eq. 2.

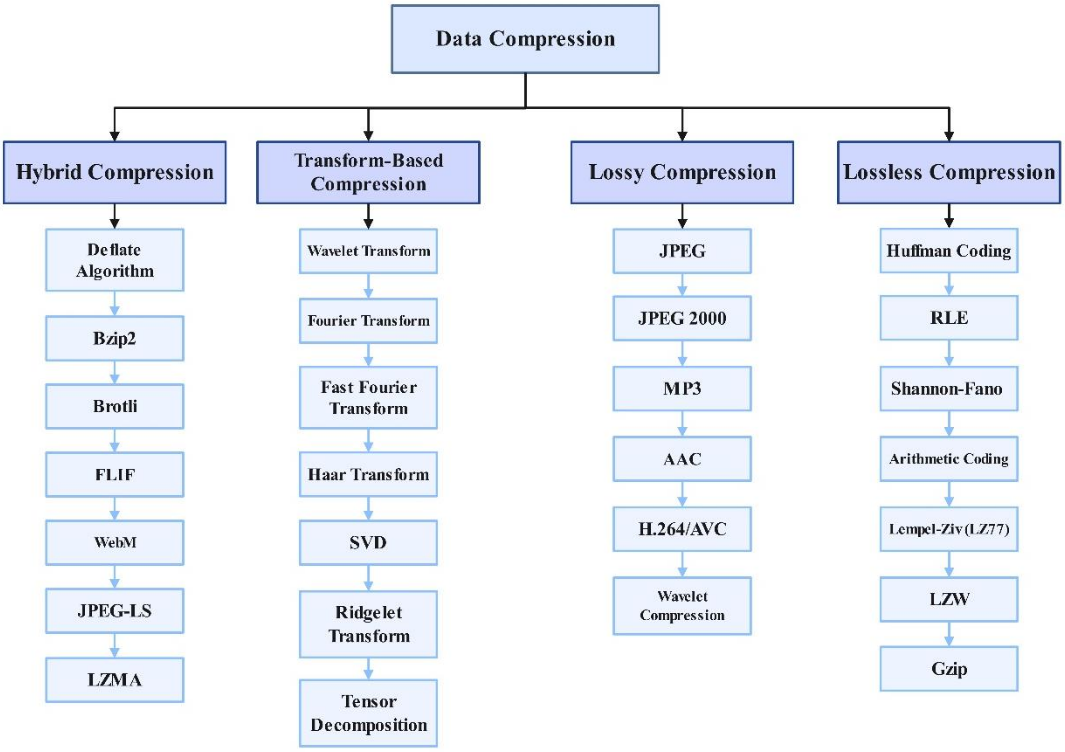
$$CG = (uncompressed\ size\ -\ compressed\ size) / (uncompressed\ size)\ \times 100 \tag{2}$$

This metric is important because it directly reflects how well a compression method minimizes data storage requirements. A higher CG value indicates that the compression technique is highly effective in reducing the amount of data, making it especially beneficial for applications where storage space or transmission bandwidth is limited. Effective compression gain not only reduces storage costs but also optimizes data transmission efficiency, leading to faster processing times and lower network congestion. In real-time systems and IoT environments, a higher CG can be critical, as it allows for the handling of larger data sets within constrained resources.

Data compression methods can generally be classified from various perspectives. One of the most practical classifications is based on compression quality. In this regard, compression methods can be categorized into two groups: lossless compression and lossy compression. In lossless compression, all original data can be fully restored without any alteration, making it ideal for data requiring high accuracy, such as text and medical data. On the other hand, lossy compression reduces data by removing some information, but the result remains acceptable for human perception. This type of compression is commonly used in multimedia applications like images, videos, and audio, where a slight quality loss is permissible. Given the importance of data within the smart home ecosystem, it is essential for data to be compressed at an adequate speed and in a lossless manner. Long-term decisions and behavioral analyses, such as creating energy consumption scenarios, rely heavily on this data.

In this context, employing efficient compression methods capable of preserving all essential data features is crucial. This data, often collected from numerous high-frequency sensors, requires precise compression to enhance transmission and storage efficiency, ultimately increasing equipment lifespan and reducing operational costs. In the following, some of the most practical lossless compression methods are discussed in detail. These methods include algorithms such as Run-Length Encoding(RLE), Huffman Coding, dictionary-based algorithms like LZ77 and LZW, as well as Arithmetic Coding. Subsequently, data compression methods are categorized in Fig. 1.

*Figure 1: Data compression methods*




Each of these algorithms has its own advantages and disadvantages, depending on the type and structure of the input data, and they demonstrate varied performance across different scenarios.

## *3 Run-Length Encoding (RLE)*

Run-Length Encoding is a lossless compression method. In lossless compression, the original data can be perfectly reconstructed from the compressed data without any loss of information. RLE achieves this by representing consecutive runs of identical values in the data as a single value and its count. When decompressed, the original data is reconstructed by replicating these values according to their counts. RLE is particularly effective when there are long runs of identical values in the data. However, its efficiency depends on the characteristics of the data being compressed. While RLE is a simple and fast compression method, it may not provide the same level of compression as more advanced algorithms in certain scenarios. Nonetheless, its simplicity and lossless nature make it suitable for various applications, especially when real-time compression or minimal computational resources are considerations.

## *4 Huffman coding*

Huffman coding, so named for its creator David A. Huffman, is a commonly used lossless data compression technique. Huffman coding is useful in many data compression techniques, including those that compress images and videos. It works especially well for text files. The first step in using the technique is to generate a table of probabilities, which can be done statically or dynamically. The encoder and decoder are both aware of the same fixed probability table when using the static approach. With the help of Huffman coding, a compact representation of the input is produced while maintaining its information. The resulting compressed representation of the data is varied in length, with shorter codes for more common symbols and longer codes for less frequent ones.

## *5 LZ77*

The basic idea behind LZ77 is to replace repeated occurrences of data with references to a single copy. Instead of explicitly storing duplicate data, LZ77 uses a sliding window to search for matching strings of characters within a certain range. When a match is found, the algorithm represents it by a pair of values: a length and an offset. The length specifies how many characters are repeated, and the offset indicates the distance back in the input stream to the start of the repeated sequence. LZ77 is a dictionary-based compression algorithm, meaning it maintains a dictionary or sliding window of recent data and exploits redundancy in the input by referencing previous occurrences of similar data. This approach is particularly effective for compressing repetitive or redundant information, making LZ77 suitable for a wide range of applications, including file compression and transmission of data over networks. Variants and improvements to the original LZ77 algorithm, such as LZ78 and DEFLATE (used in the popular ZIP file format), have been developed to address certain limitations and enhance compression efficiency. Despite being over four decades old, LZ77 remains an important and influential algorithm in the field of data compression.

### *6 LZ78*

Like LZ77, LZ78 is widely used in a variety of compression applications and provides an efficient means of reducing data size while preserving information. The main innovation introduced by LZ78 is the use of dynamic dictionaries to represent repeated data sets. Instead of a fixed size scrolling window, LZ78 creates a dictionary on the fly when it encounters a new string. The algorithm encrypts the input data by replacing repeated substrings with references to dictionary entries. One advantage of LZ78 is its ability to handle variable-length phrases efficiently, as it dynamically adapts the dictionary to the content of the input data. This makes it well-suited for compressing a wide range of data types. Despite its effectiveness, LZ78 is not as widely used as its predecessor, LZ77, in practical applications. However, the concepts and principles introduced by LZ78 have influenced other compression algorithms, and variations of the basic LZ78 idea are present in some modern compression techniques.

### *7 LZW*

The dictionary-based compression technique known as Lempel-Ziv-Welch, or LZW, is a popular data compression algorithm. Since LZW employs variable-length codes, the lengths of the various codes in the dictionary may change. More common patterns are given shorter codes, which helps to improve compression efficiency. LZW reads input data in streams or fixed-size blocks for compression. It keeps a dictionary up to date with the codes and sequences that go with them. It searches the input for recurring patterns and substitutes shorter dictionary codes for them. A stream of codes is used to represent the compressed output. When patterns or sequences are repeated, these codes can be substantially more compact than the original data. LZW remains a notable and influential compression algorithm, and its principles have inspired other compression techniques and standards. Despite its age, LZW is still relevant and can be found in various applications where efficient data compression is essential.

### *8 Arithmetic coding*

Arithmetic coding is a form of entropy encoding used in data compression. It's a variable-length encoding algorithm that represents entire messages with a single, real-valued number within the interval [0, 1). Developed by IBM researchers Jorma Rissanen and Robert M. Gray in the late 1970s, arithmetic coding is known for its efficiency and ability to achieve near-optimal compression rates. Arithmetic coding operates on individual symbols or characters, treating the entire input message as a sequence of symbols. Each symbol is assigned a unique probability based on its frequency of occurrence in the message. The encoding process relies on probability distributions for each symbol. These distributions are determined based on the frequency or probability of each symbol in the given message. The more frequent a symbol, the higher its probability. Arithmetic coding maps each symbol to a subinterval within the [0, 1) interval. The entire [0, 1) range represents the cumulative probability of the entire message. Each symbol's subinterval is proportional to its probability. Arithmetic coding has been used in various applications, including image and video compression standards (e.g., JPEG, MPEG), as well as in some lossless compression algorithms. Despite its theoretical elegance and efficiency, arithmetic coding may be computationally intensive, and practical implementations often use approximations for efficiency.

### *9 SAX*

Symbolic Aggregate Approximation (SAX) is a technique used in time series data analysis to approximate and represent the patterns within a time series with a reduced set of symbols. Developed by Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi, SAX is particularly useful for reducing the dimensionality of time series data while preserving important characteristics. SAX addresses the challenge of dimensionality by transforming the original time series into a symbolic representation with a significantly reduced set of symbols. This transformation allows for the efficient analysis of patterns and trends. Before applying SAX, the time series is typically preprocessed using a technique called Piecewise Aggregate Approximation (PAA). PAA involves dividing the time series into segments and representing each segment with its average value. This reduces the original time series to a sequence of average values, making it more amenable to symbolic representation. SAX has been applied in various domains, including anomaly detection, classification, and clustering of time series data. Its ability to represent time series data with a reduced set of symbols makes it particularly useful for identifying patterns and trends in large datasets.

## 2. Smart Home and IoT

In recent years, due to increasing demand and the subsequent need to improve the quality of life for human societies, industries and various fields have undergone significant changes. According to researchers in the field of technology and the Internet of Things, smart homes represent a transformative shift in how technology interacts with domestic environments. Moreover, with the growing penetration of IoT devices, it is now possible to establish the necessary conditions for automation, energy management, and enhanced security (Waheb A. Jabbar, 2023).

In smart homes, IoT devices are used for monitoring and automating household activities. These devices continuously monitor the environment and measure data related to its conditions. This data is then sent to a processing system, often a microcontroller. After processing this data, homeowners can check the conditions of their homes in real time. Additionally, in the event of abnormalities or problems, notifications are sent to users via various platforms such as mobile applications or text messages. In the smart home ecosystem, devices and central systems are connected via various communication networks such as Wi-Fi, Zigbee, and Bluetooth, and the collected information is sent to central servers or cloud processing systems for analysis and automation.

These nodes can collect various types of information, including temperature, humidity, light intensity, air quality, vibration levels, and the presence of individuals in the home (Popoola et al., 2024). Subsequently, they are utilized through computational algorithms and artificial intelligence to optimize energy consumption, enhance safety, and improve user comfort. One of the primary advantages of IoT-enabled smart homes is their ability to provide quick responses to environmental changes and make real-time autonomous decisions. These features turn the smart home into a dynamic ecosystem that can continuously adapt to the needs of its users. For example, in a smart home, if air quality drops, the ventilation system automatically activates. Additionally, if motion is detected during unusual hours, the security system is activated, and an alert is sent to the users.

These capabilities significantly improve the safety and efficiency of smart homes. Furthermore, these systems, leveraging machine learning algorithms, can analyze energy consumption patterns and provide recommendations for cost reduction (Rani et al., 2024; Varadarajan et al., 2024). In continuation, Fig. 2, illustrates the application domains of a smart home.

*Figure 2: Application domains of a smart home*


Data compression is also one of the most critical challenges and, at the same time, key solutions in smart homes. This compression reduces the volume of data and energy consumption during data transmission, thus improving the system's responsiveness and quality. In an IoT-enabled smart home, data processing can occur at the edge (Edge Processing) or in the cloud. Edge processing reduces latency and accelerates immediate reactions, while cloud processing enables the analysis of more extensive data and the use of complex models to enhance smart home performance. Utilizing data compression algorithms in IoT-enabled smart homes is vital as it facilitates faster and more efficient data transmission to servers or other devices. These processes play a pivotal role in enhancing the monitoring and automation capabilities of smart home system (Chakraborty et al., 2023; Chiarot & Silvestri, 2023).

## 3. Background and Related Work

Various studies highlight that implementing effective compression algorithms can lead to substantial energy savings, which is crucial for battery-operated devices. For instance, the use of the LZ78 algorithm in microcontroller-based systems can save considerable energy during data transmission (Piątkowski et al., 2024). Additionally, adaptive data compression schemes can reduce power consumption by an average of 40%, enhancing device longevity by up to 50% (Al-Kadhim & Al-Raweshidy, 2021). Furthermore, data compression can improve the Age of Information (AoI) and energy efficiency (EE) in cellular IoT networks, with reductions in AoI by up to 82% when optimizing compression ratios (Hu et al., 2023). Overall, these findings underscore the importance of tailored compression strategies in enhancing the performance and sustainability of IoT devices. The life cycle of IoT devices is closely tied to energy consumption patterns, particularly in battery-powered devices. (Sharaff & Sinha, 2021) argue that efficient data management through compression can reduce battery replacements, enhancing device sustainability.

The authors in (Kheir El Dine et al., 2024) investigates energy optimization techniques in NB-IoT technology, focusing on the impact of data compression and extending data transmission intervals on reducing energy consumption. NB-IoT, a key infrastructure within LPWAN networks, is recognized for its low power consumption, wide range, and high scalability in Internet of Things (IoT) applications. The study highlights that implementing data compression techniques and modifying device configurations can reduce energy consumption across various scenarios. Specifically, in areas with limited coverage, data compression significantly decreases energy usage. Simulations conducted using nRF9160 sensors compared three scenarios. The results demonstrate that extending data transmission intervals and compressing data packets in poor coverage areas play a crucial role in reducing energy consumption and extending battery life in NB-IoT devices. Moreover, choosing appropriate configurations and reducing the number of transmitted packets can further enhance device efficiency. (de Oliveira Júnior et al., 2023) examines data compression algorithms in LoRa networks and evaluates their impact on compression rates, processing time, energy consumption, and battery life of IoT devices. Given the hardware and energy constraints of IoT end devices, data transmission accounts for the highest energy consumption, comprising up to 60% of the total energy used. To mitigate this energy consumption, the paper adapts classical algorithms such as Huffman, Arithmetic, LZ77, LZ78, and LZW to IoT devices with ESP32 architecture and LoRa communication. Two real-world datasets were used for experimentation: temperature monitoring data for concrete in large buildings and GPS data. Results show that the LZW algorithm provides the best performance with a compression rate of 69% for temperature data and an energy consumption reduction of up to 22%. Furthermore, dictionary-based algorithms like LZW and LZ78 consume more memory but are more efficient at compressing repetitive data. The study also highlights that data compression methods not only reduce energy consumption but also significantly decrease the volume of transmitted data. This reduction contributes to extending the battery life of devices and improving the overall performance of LoRa networks. Additionally, an analytical model for estimating energy consumption is presented, which can assist in designing more efficient IoT systems.

Despite providing valuable insights into data compression in LoRa networks, this paper has several research gaps. Firstly, the compression algorithms focus only on two specific datasets, leaving diverse data types unexplored. Secondly, the impact of these algorithms on network latency and hardware resources such as CPU and RAM has not been analyzed. Moreover, the study does not investigate modern or hybrid algorithms, such as those based on deep learning techniques. The performance of the algorithms under real-world LoRa network conditions, such as noisy environments, has not been thoroughly evaluated, and no comparison has been made with other LPWAN protocols like NB-IoT and Sigfox. Additionally, the security of compressed data and the cost-benefit trade-offs of the compression algorithms have not been thoroughly examined. Finally, a comprehensive decision-making model for selecting the most suitable algorithm under varying conditions is missing.

The authors in (Sudha et al., 2024) investigate use of data compression algorithms to improve energy efficiency in ZigBee-based Wireless Sensor Networks (WSNs) for habitat monitoring. The proposed system collects environmental data such as temperature, humidity, and images through sensor nodes and compresses the data using algorithms like Huffman and Modified Huffman before transmission. Simulations conducted using MATLAB demonstrate that these methods can reduce energy consumption and increase network lifetime. The study also highlights how compression algorithms, by reducing the volume of transmitted data, optimize energy resources and enhance network efficiency. Experimental results show that the Modified Huffman algorithm offers better compression rates and ensures higher PSNR compared to classical methods. Despite its positive outcomes, this paper has several research gaps. Firstly, the study predominantly focuses on image data, leaving other types of data, such as audio or complex signals, unexplored. Secondly, the impact of these algorithms on network latency or processing time under real-world conditions has not been analyzed. Furthermore, comparisons with modern or hybrid algorithms, such as machine learning-based methods, are absent. The study heavily relies on simulations and lacks real-world testing in complex habitat environments. Additionally, the security of compressed data during transmission and a cost-benefit analysis of the methods have not been addressed. Lastly, a comprehensive model for selecting the best compression method under different scenarios has not been proposed. (Hwang et al., 2023) introduces a novel lossless compression technique called Bit Depth Compression (BDC), which dynamically determines the size of data packets based on the bit depth level of sensor data. The method is designed to minimize storage space waste and optimize network bandwidth usage. BDC employs forecasting algorithms like delta coding and RNN to compress time-series data and then reduces wasted space by splitting data into sub-packets. Experiments demonstrated that this approach achieves up to 247% improvement in compression ratio and reduces energy consumption by up to 34%, compared to other compression techniques such as Sprintz and TSXor. The primary application of this method is for transmitting sensor data from end devices to edge nodes or cloud servers without any data loss, maintaining high accuracy for AI-driven analysis. While the BDC method delivers remarkable improvements in compression and energy efficiency, the paper does not address the impact of more complex or rapidly changing data patterns in real-time scenarios. Additionally, computational limitations in low-power devices, which may increase with the complexity of forecasting algorithms, remain unexplored. Furthermore, the paper does not compare BDC's performance with newer deep learning techniques for time-series compression. There is a need for further studies on applying this method at a larger scale and in real-world environments.

As highlighted in a wide range of studies, research in the field of data compression has paid relatively little attention to energy consumption and the delay introduced by compression techniques. However, in IoT ecosystems, managing energy consumption and minimizing delay play a critical role in enhancing overall system performance. Moreover, most existing studies focus on computational architectures at higher levels of the network, such as centralized servers. These architectures not only entail high implementation and maintenance costs but also require longer cycles to send data, process it, and then deliver responses. Additionally, these server-dependent architectures face limitations, such as increased latency in low-bandwidth or noisy network conditions. In contrast, edge computing architectures can significantly reduce latency and improve the responsiveness to environmental changes.

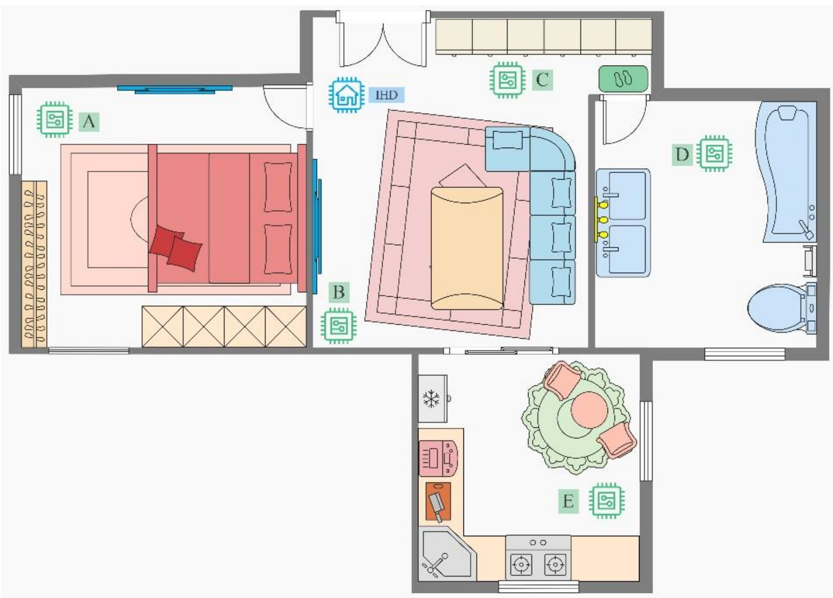
However, the application of data compression algorithms at the edge level requires optimizations to manage limited resources, such as computational power and energy consumption. Therefore, conducting a study that not only evaluates data compression algorithms but also examines their impact on energy consumption, latency, and feasibility for deployment at lower network levels (e.g., edge computing) holds significant value. This research aims to provide optimized solutions that enhance the efficiency of IoT-enabled smart homes while addressing the shortcomings of previous studies, thereby contributing to the advancement of this field.

## 4. Proposed Scheme

In general, the proposed scheme consists of two main parts: hardware, including sensors and a data aggregator node (IHD), and the data compression mechanism. In Fig. 3, the architecture of the proposed scheme in a candidate smart home is illustrated. This scheme is implemented in such a way as to manage the home fully intelligently using a set of measurement nodes and communication systems.

In this smart home, five candidate measurement nodes, labeled A to E respectively, can measure various parameters such as temperature, humidity, light intensity, sound level, air quality, and vibration level. These nodes are placed in the bedroom, living room, hallways, bathroom, and kitchen, respectively. The precise selection of these locations aims to monitor key environmental conditions from all critical areas of the home and to track any changes in real-time. The purpose of distributing these sensors is to enable real-time monitoring of the smart home conditions, improve energy efficiency, and enhance security.

*Figure 3: Smart home architecture*




This system, leveraging a wireless network, gathers data from all parts of the house and transmits it to the aggregation node. The hardware designed for this purpose is introduced and reviewed in the following sections. In each node, the data is processed immediately upon measurement, and if any error or abnormality is detected, necessary alerts are sent both at the node level and through the internal display to the user. This approach helps reduce computational load and energy consumption, as only essential data is sent to the aggregation node. These data are then efficiently compressed using lossless compression algorithms to ensure that data transmission occurs with minimal volume and maximum speed. Finally, the compressed data is transmitted wirelessly to the data aggregation node, represented by IHD. After the aggregation process, the data is sent to cloud servers for storage and further processing. This cloud system enables data analysis and the execution of advanced algorithms, which can lead to improved accuracy and efficiency of the smart home system. With this architecture, users can enjoy enhanced safety and comfort while also reducing the energy costs of their home.

## 13 Hardware

As mentioned, the proposed scheme consists of two hardware layers. The first layer comprises sensors and measurement boards for various variables in a smart home. These sensors measure data such as temperature, humidity, light intensity, air quality, sound levels, and energy consumption. For instance, temperature and humidity sensors can be used to enhance comfort and automatically adjust heating and cooling systems, while air quality sensors provide information on pollution levels or harmful gases in the environment. The second layer is the in-home display, which acts as a bridge between consumers and power grid. This display not only allows users to monitor their energy consumption but can also provide alerts and notifications related to consumption optimization or network emergency status. Thus, users can make more informed decisions to better manage their energy usage based on this information. The proposed hardware design is shown in Fig. 4.

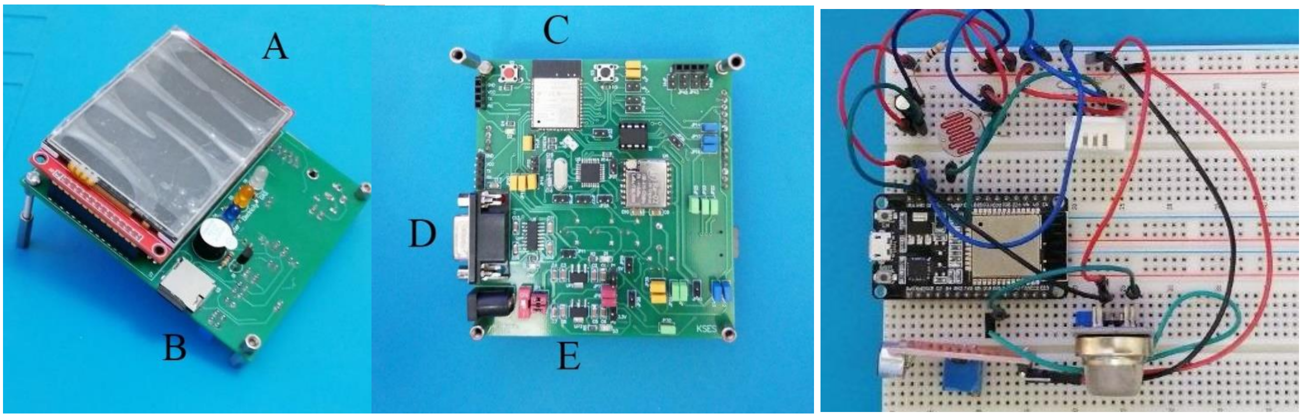



*Figure 4: Hardware part of the proposed scheme*

In Figure 4(left side), section A represents the IHD node display, which is a 3.2-inch touch screen used to display essential information about the smart home system. This touch display allows interaction with the user, providing quick access to system settings and real-time information, such as sensor status, energy consumption, and system alerts. Below the screen, several LEDs are located, enabling the user to monitor the operation of meters and other devices in the smart home. These LEDs can also serve as error or alert indicators. Section B is dedicated to an SD card slot for long-term data storage. This feature allows users to analyze stored data for extended periods, such as energy consumption patterns or sensor behavior, enabling more detailed insights and planning. Section C includes the ESP32, which serves as the main processor of the board. The ESP32 is a powerful microcontroller module equipped with Dual-Core Tensilica Xtensa LX6 processors, running at up to 240 MHz. It supports built-in Wi-Fi and Bluetooth, making it highly suitable for wireless communication. The ESP32 features multiple GPIOs, ADC, and PWM pins, allowing seamless integration with a wide range of sensors and devices. Its internal memory includes 520 KB of SRAM, along with support for external flash, making it ideal for advanced IoT applications. Section D serves as the serial output of the board, enabling communication with a computer or other devices through protocols like UART or USB to Serial. This functionality is essential for data transfer, debugging, or updating the board’s firmware. Finally, Section E comprises the power supply circuit, powered by a 5V, 2A power source. This circuit uses high-quality components to ensure stable voltage and current delivery, ensuring the reliable performance of the board and connected devices.

Additionally, on the right side of Figure 4, the sensor hardware node is depicted, designed for measuring various environmental variables. This node includes sensors for measuring temperature, humidity, sound levels, air quality, and light intensity. Moreover, it is capable of detecting vibration levels and movements within the environment, making it useful for security and environmental analysis. The sensor node is designed to transmit the collected data in real-time to the IHD node or the central processor. After initial compression and processing, these data are utilized for further analysis and automated decision-making. Furthermore, this node features a low-power design to extend battery life, a critical aspect in IoT-based applications.

## 14 DC methods

As previously discussed, there are various methods available for data compression, each offering unique advantages and limitations depending on the type of data and the system's objectives. In this research, to achieve a comprehensive evaluation, we assess the performance of seven different compression methods, including RLE, Huffman coding, LZ77, LZ78, LZW, Arithmetic and SAX techniques, with respect to several criteria: compression ratio, computation time required for each method, and energy consumed during the compression process.

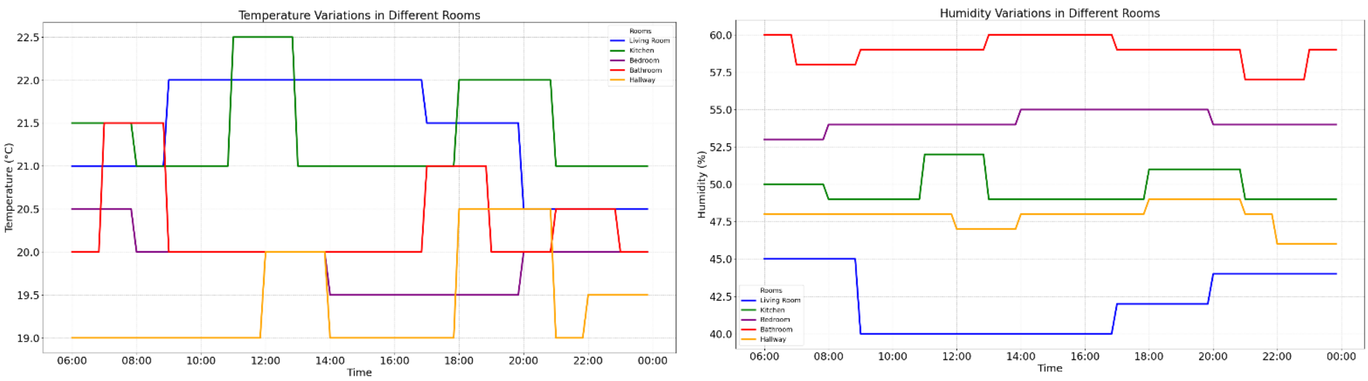
This evaluation enables us to examine and analyze the distinctions among these methods, helping to identify the most suitable compression algorithm for specific applications and environments, such as smart homes and Internet of Things (IoT) systems.

## 15 Experimental result

Considering the characteristics of a smart home and the need for comprehensive environmental data, deploying various sensors with a wide range of applications is essential. For the purposes of this study, five key points in a smart home have been selected, namely the kitchen, bedroom, living room, hallway, and bathroom, to serve as candidate locations for data collection. In each of these locations, six critical environmental variables(temperature, humidity, air quality, sound level, light intensity and vibration data)are measured.

These variables are chosen due to their relevance in providing valuable insights into the indoor conditions of a smart home, enhancing comfort, safety, and energy efficiency. The data obtained from these five candidate points are shown in Fig. 5 to Fig. 7 for each respective variable, offering insights that support decision-making in managing energy and indoor conditions within a smart home.

*Figure 5: Temperature and humidity data of the 5 candidate points*



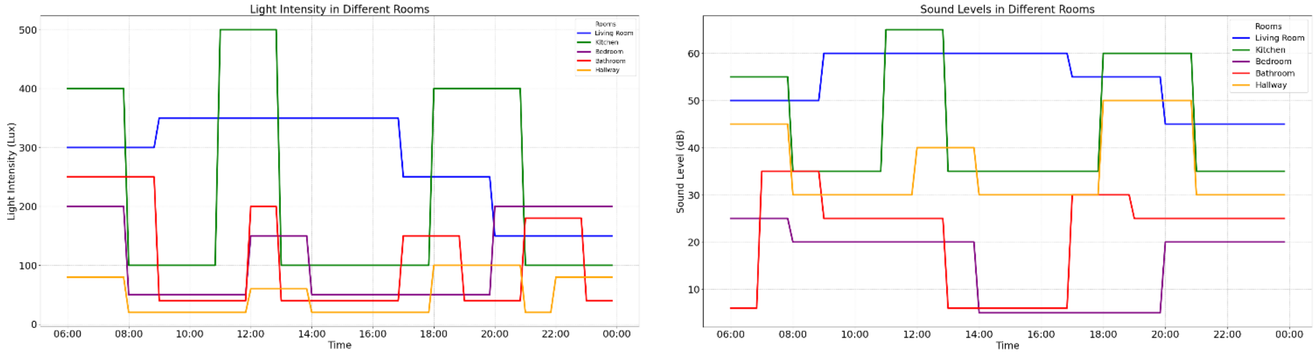
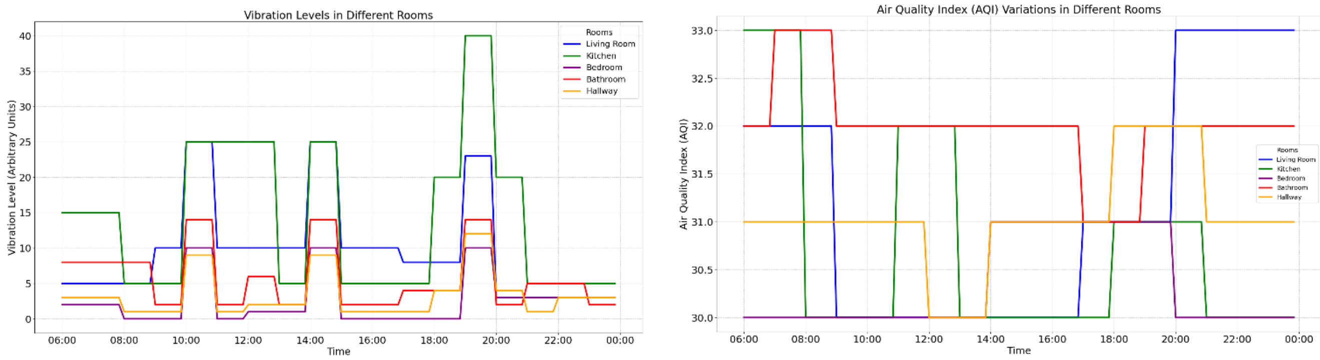



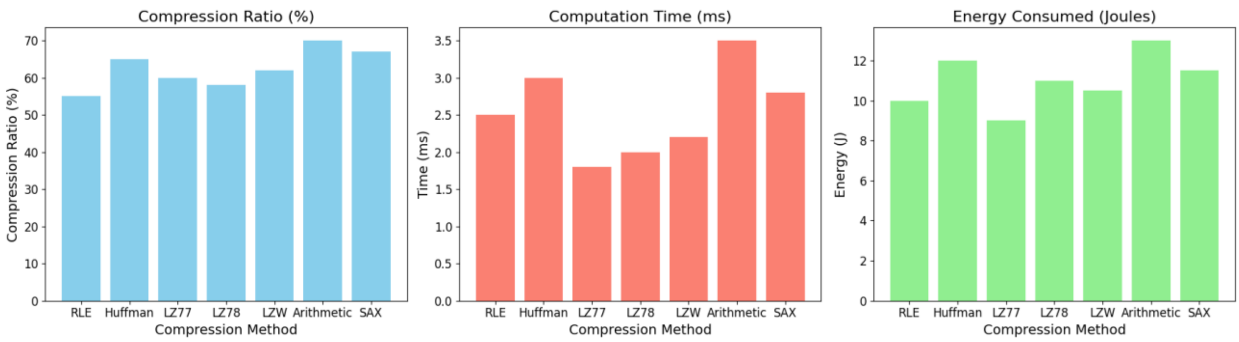

*Figure 6: Light and sound data of the 5 candidate points*

*Figure 7: Vibration and air quality data of the 5 candidate points*




In this study, the impact of various data compression methods on three key metrics, compression ratio, computation time, and energy consumption was analyzed. These metrics are critical for optimizing the performance of IoT systems, where resource constraints and the need for efficient data transmission are paramount. As shown in the Fig. 8, compression algorithms such as LZW and Arithmetic Coding achieve higher compression ratios compared to simpler methods like RLE. Specifically, the LZW algorithm, achieving a compression ratio of approximately 70%, demonstrates the best performance, indicating its suitability for applications where data reduction is a priority.

*Figure 8: Performance comparison of data compression methods*




In terms of computation time, more complex algorithms like Arithmetic and SAX require longer processing times due to their advanced mathematical structures and iterative encoding processes. In contrast, simpler methods like LZ77 and RLE demand less processing time, making them ideal for real-time applications with limited computational resources. However, results indicate that the additional computation time required by complex algorithms does not significantly affect the overall data transmission delay, especially when compared to the benefits of reduced data size. This finding suggests that even computationally intensive algorithms may be viable for scenarios where slight delays can be tolerated.

Regarding energy consumption, algorithms that provide higher compression ratios, such as Arithmetic and SAX, consume more energy during the encoding process. This is primarily due to the increased number of calculations required for their operations. However, this additional energy consumption is justified by the reduced volume of transmitted data, which significantly lowers the energy required for data transmission—a major factor in IoT systems.

Notably, the results demonstrate that the LZW algorithm achieves a good compromise between compression ratio and energy consumption, balancing computational efficiency with energy savings. This makes it an excellent choice for IoT applications where devices are often battery-powered and energy efficiency is critical. Furthermore, the study highlights the trade-offs inherent in selecting a compression algorithm. While simpler algorithms like RLE excel in low-latency scenarios, their lower compression ratios may lead to increased data transmission volumes and higher long-term energy costs.

Conversely, advanced algorithms like Arithmetic Coding optimize data storage and transmission efficiency but may introduce additional computational overhead. The balance between these factors is highly dependent on the specific requirements of the application, such as the size of the dataset, the frequency of data transmission, and the available processing power of the device.

In conclusion, the findings underscore the importance of selecting the right compression method based on the use case. The LZW algorithm, with its high compression ratio, moderate computational complexity, and reasonable energy usage, stands out as a versatile option for IoT applications. By adopting appropriate compression techniques, IoT systems can achieve significant improvements in data handling efficiency, reduce network congestion, and extend the operational lifetime of battery-powered devices. This study provides a foundation for further research into compression methods tailored to the unique challenges of IoT ecosystems.

## 5. Conclusion

The findings of this research demonstrate that employing data compression algorithms can significantly reduce the volume of transmitted data while optimizing energy consumption. Despite a slight increase in computation time for some algorithms, its impact on overall network latency is negligible. Overall, this study highlights the importance of data compression in IoT networks and shows that selecting an appropriate algorithm based on application requirements (e.g., data volume reduction, energy efficiency, or delay minimization) can positively impact the system's overall performance. The LZW algorithm, due to its balanced performance in terms of compression ratio, processing time, and energy consumption, is recommended as an optimal solution. While the literature presents a compelling case for the benefits of data compression in enhancing energy efficiency and extending device life cycles, several challenges remain. For instance, the implementation of lightweight encryption methods for data security, can potentially conflict with energy optimization strategies. The need for balancing security measures with energy consumption poses an ongoing challenge for IoT systems. Additionally, the exploration of big data management frameworks, suggests that the integration of advanced algorithms for machine learning can further optimize energy usage. Yet, the direct implications of such technologies on data compression and energy consumption in IoT devices require further investigation.

## Acknowledgment

The author of this article expresses gratitude to ChatGPT as a highly useful tool for grammar revision.

## References

1. Al-Kadhim, H. M., & Al-Raweshidy, H. S. (2021). Energy Efficient Data Compression in Cloud Based IoT. *IEEE Sensors Journal*, *21*(10), 12212–12219. https://doi.org/10.1109/JSEN.2021.3064611
2. Azar, J., Makhoul, A., Barhamgi, M., Couturier, R., An, R. C., & Couturier, R. (2019). energy efficient IoT data compression approach for edge machine learning. *Future Generation Computer Systems*, *96*, 168–175. https://doi.org/10.1016/j.future.2019.02.005ï
3. Chakraborty, A., Islam, M., Shahriyar, F., Islam, S., Zaman, H. U., & Hasan, M. (2023). Smart Home System: A Comprehensive Review. In *Journal of Electrical and Computer Engineering* (Vol. 2023). Hindawi Limited. https://doi.org/10.1155/2023/7616683
4. Chen, H., Chen, J., Lu, Z., & Wang, R. (2022). CMIC: an efficient quality score compressor with random access functionality. *BMC Bioinformatics*, *23*(1). https://doi.org/10.1186/s12859-022-04837-1
5. Chiarot, G., & Silvestri, C. (2023). Time Series Compression Survey. *ACM Computing Surveys*, *55*(10). https://doi.org/10.1145/3560814
6. de Oliveira Júnior, J. A., de Camargo, E. T., & Oyamada, M. S. (2023). Data Compression in LoRa Networks: A Compromise between Performance and Energy Consumption. *Journal of Internet Services and Applications*, *14*(1), 95–106. https://doi.org/10.5753/jisa.2023.3000
7. Fu, Y., Yang, X., Yang, P., Wong, A. K. Y., Shi, Z., Wang, H., & Quek, T. Q. S. (2021). Energy-efficient offloading and resource allocation for mobile edge computing enabled mission-critical internet-of-things systems. *Eurasip Journal on Wireless Communications and Networking*, *2021*(1). https://doi.org/10.1186/s13638-021-01905-7
8. Hu, H., Dong, Y., Jiang, Y., Chen, Q., & Zhang, J. (2023). On the Age of Information and Energy Efficiency in Cellular IoT Networks With Data Compression. *IEEE Internet of Things Journal*, *10*(6), 5226–5239. https://doi.org/10.1109/JIOT.2022.3222343
9. Hwang, S. H., Kim, K. M., Kim, S., & Kwak, J. W. (2023). Lossless Data Compression for Time-Series Sensor Data Based on Dynamic Bit Packing. *Sensors (Basel, Switzerland)*, *23*(20). https://doi.org/10.3390/s23208575
10. Kheir El Dine, M., Al Haj Hassan, H., Nasser, A., Zaki, C., Moawad, A., & Mansour, A. (2024). Reducing Energy Consumption in NB-IoT by Compressing Data and Aggregating Transmission. *Lecture Notes in Electrical Engineering*, *1110 LNEE*, 181–188. https://doi.org/10.1007/978-3-031-48121-5_26
11. Krishnamurthi, R., Gopinathan, D., & Nayyar, A. (2021). A comprehensive overview of fog data processing and analytics for healthcare 4.0. In *Signals and Communication Technology* (pp. 103–129). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-030-46197-3_5
12. Piątkowski, D., Puślecki, T., & Walkowiak, K. (2024). Study of the Impact of Data Compression on the Energy Consumption Required for Data Transmission in a Microcontroller-Based System. *Sensors*, *24*(1). https://doi.org/10.3390/s24010224

13. Popoola, O., Rodrigues, M., Marchang, J., Shenfield, A., Ikpehai, A., & Popoola, J. (2024). A critical literature review of security and privacy in smart home healthcare schemes adopting IoT & blockchain: Problems, challenges and solutions. *Blockchain: Research and Applications*, *5*(2). https://doi.org/10.1016/j.bcra.2023.100178
14. Rani, K. P., Sreedevi, P., Veeranjaneyulu, P., Kanth, M. R., Allam, S., & Mohanty, J. R. (2024). Smart Home Automation Using AI and IoT with High Security. *Proceedings of 2nd International Conference on Advancements in Smart, Secure and Intelligent Computing, ASSIC 2024*. https://doi.org/10.1109/ASSIC60049.2024.10508006
15. Ren, J., Ruan, Y., & Yu, G. (2019). Data Transmission in Mobile Edge Networks: Whether and Where to Compress? *IEEE Communications Letters*, *23*(3), 490–493. https://doi.org/10.1109/LCOMM.2019.2894415
16. Sharaff, A., & Sinha, G. R. (2021). Data Science and Its Applications. In *Data Science and Its Applications*. Chapman and Hall/CRC. https://doi.org/10.1201/9781003102380
17. Sudha, G., Archana, M., Sharmila, S., Nandhini, K., Saranya, S., & Sankari, S. (2024). Analysis of Energy Efficiency Improvement Using Data Compression Algorithms for Habitat Monitoring in WSNs Using Zigbee. *2024 International Conference on Communication, Computing and Internet of Things, IC3IoT 2024 - Proceedings*. https://doi.org/10.1109/IC3IoT60841.2024.10550202
18. Varadarajan, M. N., Viji, C., Rajkumar, N., & Mohanraj, A. (2024). Integration of Ai and Iot for Smart Home Automation. *SSRG International Journal of Electronics and Communication Engineering*, *11*(5), 37–43. https://doi.org/10.14445/23488549/IJECE-V11I5P104
19. Waheb A. Jabbar, M. H. A. N. S. S. A. and S. K. M. (2023). *Design and Implementation of IoT Based Automation System for Smart Home*. Institute of Electrical and Electronics Engineers.
20. Wen, L., Zhou, K., Yang, S., & Li, L. (2018). Compression of smart meter big data: A survey. In *Renewable and Sustainable Energy Reviews* (Vol. 91, pp. 59–69). Elsevier Ltd. https://doi.org/10.1016/j.rser.2018.03.088