Ministry of Higher Education
& Scientific Research
University of Babylon
College of Information Technology
Department of Information Networks

Energy-Aware Prediction and Processing Approaches for Minimizing Communication Cost in IoTs Networks

A Dissertation Submitted to the Council of College of Information Technology-University of Babylon in a Partial Fulfilment of the Requirements for the Degree of Doctorate of Philosophy in Information Technology \ Information Networks

By
Ahmed Mohammed Hussein Al-Gazaly

Supervised By
Prof. Dr. Ali Kadhum Idrees
Prof. Dr. Raphaël Couturier

2023 A. D. 1445 A. H.

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

﴿ يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ ﴾

صدقَ اللهُ العليُّ العظيم

سورة المجادلة: الاية (11)

SUPERVISOR'S CERTIFICATION

I certify that the dissertation entitled "Energy-Aware Prediction and Processing Approaches for Minimizing Communication Cost in IoTs Networks" was prepared under my supervision at the Department of Information Network / College of Information Technology/ University of Babylon as partial fulfillment of the requirements of the degree of Doctorate of Philosophy in Computer Sciences.

Signature:

Name: Prof. Dr. Ali Kadhum Idrees

Date: / / 2023

THE HEAD OF THE DEPARTMENT CERTIFICATION

I in view of the available recommendations, I forward the dissertation entitled "Energy-Aware Prediction and Processing Approaches for Minimizing Communication Cost in IoTs Networks" for debate by the examination committee.

Signature:

Name: Prof. Dr. Saad Talib Hasson

Head of Dept.

Date: / / 2023

DECLARATION

I hereby declare that this dissertation, submitted to the University of Babylon as fulfillment of requirements for the degree of Doctorate of Philosophy in Computer Sciences has not been submitted as an exercise for a similar degree at any other university. I also certify that the work described here is entirely my own except for excerpts and summaries whose sources are appropriately cited in the references.

Signature:

Name: Ahmed Mohammed Hussein Al-Gazaly

Date: September 2023

DEDICATION

THIS DISSERTATION IS DEDICATED TO THE LOVING MEMORY OF MY FATHER

MOHAMMED HUSSEIN JASIM AL-GAZALY

YOU HAVE SUCCESSFULLY MADE ME THE PERSON I AM BECOMING.

I WILL ALWAYS BE REMEMBERED YOU, AND I WILL NEVER FORGET YOU DAD.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my God, ***Allah Almighty***, for giving me endless graces. My deep sense of gratitude to the beacon of science, to the master of creatures, to the greatest Prophet, ***Mohammed*** (Peace be upon Him and His Family).

I take this opportunity to express my sincere gratitude and greatest appreciation to my supervisor ***Dr. Ali Kadhum Idrees*** for his continuous support for my Ph.D. study, his patience, motivation, enthusiasm, and immense knowledge. The words are inadequate and I can't say thank you enough for his tremendous support and help. I feel motivated and encouraged every time I attend his meeting. His tireless guidance has helped me immensely in researching and writing this dissertation.

Also, I would like to show my gratitude to whom weave my happiness from strings woven from her heart to my dear ***Mother***. My thanks and appreciations also go to my ***Brother*** and my ***Sister***, for their encouragement, support and patience.

I wish to express my love and gratitude to my beloved ***Wife*** for her understanding and endless love throughout the duration of my study. I also need to thank my wonderful children, ***Sama***, ***Jana***, and ***Mohammed Al-Ameen***.

This dissertation would not have been possible without the support of my Ph.D. colleagues, and my friends, especially ***Assis. Prof. Dr. Ali Al-Quraby***, the list is almost endless, it will be impossible to mention all of their names here.

I would also like to express my thanks and gratitude to all those who contributed to making this dissertation possible, and foremost among them all the teaching and staff members at the College of Information Technology at the University of Babylon, headed by ***Prof. Dr. Hussein A. Lafta***.

Finally, yet importantly, I would like to thank my College of Sciences for Girls and especially the Computer Department for the opportunity to complete my studies.

ABSTRACT

In the modern world, it will be necessary to deploy a large number of sensor devices to sense everything around us in order to detect changes, risks, and hazards and to mitigate them. This increasing number of sensor devices represents an essential data provider in the Internet of Things (IoT). The devices generate and transmit a huge amount of data which requires a large amount of storage and high processing power to come real-time processing and speed up the network. It also leads to an increase in high energy consumption. Thus, it is important to remove redundant data to reduce the data transmission before sending it to the Gateway while maintaining a good level of data quality.

This research works on two levels: the first is the Sensor Node level (SN) and the second is the Fog Gateway level (GW). The SN level suggested four energy-efficient data prediction and processing approaches to reduce redundant data and save energy while maintaining a suitable quality of received data at the next level of the network. The first approach (Distributed Energy-efficient Data Reduction (DEDaR)) used AutoRegressive(AR) prediction and Huffman Compression approaches. The second approach (a distributed prediction-compression-based mechanism (DiPCoM)) used ARIMA prediction and LZW compression approaches. The third approach (Integrated Data Prediction and Compression Techniques (IDaPCoT)) used AR prediction and LZW compression approaches. The fourth approach (Energy-efficient Data Reduction based Prediction and Encoding (EDaRePE)) used ARIMA prediction and Huffman compression approaches. All the approaches at the SN level used adaptive piecewise constant approximation (APCA) and symbolic aggregate approximation (SAX) techniques to reduce the data.

At the FW level, we designed a new Two-Tier Energy-efficient Data Reduction Technique for IoT Networks (TEDaReT) to remove the duplicates

between the data of sensors before sending them to the cloud. The TEDaReT is utilized to eliminate duplicates between the data of sensors obtained from the sensor node level by identifying similarities between them, resulting in a reduction of data of sensors before sending them to the cloud.

A Python language-based custom simulator is utilized to evaluate proposed approaches through simulation experiments using real data collected from sensor nodes that are used at the Intel Berkeley Research Lab.

The results of the simulation demonstrate efficiency in the SN level, that is, the proposed approaches increased the percentage of data reduction by 93.14%, 99.71%, 99.72%, and 97.3% respectively comparison with other approaches, and in the number of sent readings the overhead reduction up to 93.44%, 96.05%, 96.54, and 93.9% respectively. In consumed energy reached to 0.0010%, 0.000209%, 0.00019%, and 0.000212% respectively, and in while maintaining the accuracy of sent data as high as 99.33, 99.73%, 99.79%, and 99.737% respectively. As a result, depending on the results that obtaining we can say the IDaPCoT approach is the best approach comparison between the four suggested approaches.

The results of the simulation demonstrate efficiency at the FG level, that is, the proposed approach reduces the number of redundant data of sensors the percentage reaching 26.081% in comparison with other approaches.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

a ----------	Number of alphabet (for instance, if the alphabet = (*w*, *x*, *y*, *z*), *a* = 4)
AD ----------	Actual Data
AMDR --------	Adaptive Method for Data Reduction
APCA ---------	Adaptive Piecewise Constant Approximation
AR ----------	AutoRegressive
ARIMA -------	AutoRegressive Integrate Moving Average
ATP ----------	Aggregation and Transmission Protocol
DDR-IoT------	Double-Layered Data Reduction for Internet of Things
DE ----------	Differential Encoding
DEDaR -------	Distributed Energy-Efficient data Reduction
DiPCoM ------	Distributed Prediction-Compression-based Mechanism
DPDR ---------	Data Prediction and Data Reduction
DPS ----------	Dual Prediction Scheme
DS ----------	Data Set
DTW ----------	Dynamic Time Warping
ED ----------	Euclidean Distance
EDaRePE ----	Energy-efficient Data Reduction based Prediction and Encoding
FG ----------	Fog Gateway
IDaPCoT -----	Integrated Data Prediction and Compression Techniques
IoT ----------	Internet of Thinks
LMS	Least-Mean-Square
LSTMs --------	Long Short-Term Memory Networks
LZW ----------	Lempel-Ziv-Welch
N ----------	Total number of sensor nodes
ODTE ---------	Online Data Tracking and Estimation
p ----------	The total number of temperature readings generated by sensor node
PD ----------	Prediction Data
PFF ----------	Prefix-Frequency Filtering
PPMC ---------	Pearson Product-Moment Correlation Coefficient

PSN ---------- Periodic Sensor Network

$\boldsymbol{S}$ ---------- Temperature readings series $S=\{s_1,s_2,....,s_{\rho-1},s_\rho\}$

$\boldsymbol{S^{AP}}$ ---------- APCA representation of S^w,$S^{AP}= S_1^{AP},...., S_m^{AP}$

SAX ---------- Symbolic Aggregate approXimation

SFDC --------- Structure Fidelity Data Collection

$\boldsymbol{S^W}$ ---------- Segment construction using SW(S, ε), $S_W= S_1^w...., S_1^w$

$\boldsymbol{S^x}$ ---------- Symbolic representation of S^{AP}, c_1^x, .. , c_w^x

TEDaReT ---- Two-Tier Energy-efficient Data Reduction Technique for IoT Networks

WSN ---------- Wireless Sensor Network

$\boldsymbol{E_{elec}}$ ---------- The power dissipation by the radio to operate the transmitter or receiver circuits, $E_{elec}=50\ nJ/bit$

$\boldsymbol{\beta}$ ---------- Break points, $\beta=\beta_1,...,\beta_{a-1}$

$\boldsymbol{\beta_{amp}}$ ---------- The amplifier of transmitter, $\beta_{amp}=100\ pJ/bit/m^2$

$\boldsymbol{\delta}$ ---------- Similarity Threshold

LIST OF PUBLICATIONS

- **PUBLISHED PAPERS IN JOURNALS**

1. **Ahmed Mohammed Hussein, Ali Kadhum Idrees, and Raphael Couturier (2022). Distributed Energy-efficient Data Reduction Approach based on Prediction and Compression to reduce data transmission in IoT networks, International Journal of Communication Systems, John Wiley & Sons Inc, SJR (Q2), Web of Science (Clarivate), Impact factor: 1.882.**
2. **Ahmed Mohammed Hussein, Ali Kadhum Idrees, and Raphael Couturier Energy-aware Prediction-based Data Reduction and Compression in IoT networks. Submitted** to the Journal of Supercomputing, Springer Nature Switzerland, **SJR (Q2), Web of Science (Clarivate), Impact factor: 2.557.**

- **PAPERS UNDERREIVEW:**

1. **Ahmed Mohammed Hussein, Ali Kadhum Idrees, and Raphael Couturier, Energy-efficient data prediction and reduction approaches for sensing based internet of things applications, Submitted** to the **Wireless Networks**, Springer Nature Switzerland, SJR (Q2), **Web of Science (Clarivate), Impact factor: 2.701.**
2. **Ahmed Mohammed Hussein, Ali Kadhum Idrees, and Raphael Couturier, Two-Tier Energy-efficient Data Reduction Technique for IoT Networks, Submitted** to **Ad Hoc Networks**, Elsevier, SJR (Q1), **Web of Science** (Clarivate), **Impact factor: 4.816**.

CHAPTER ONE

INTRODUCTION

CHAPTER ONE: INTRODUCTION

1.1 Introduction

The fundamental aspect of the Internet of Things (IoT) is to allow the communication of virtual and physical things with each other [1]. IoT systems include embedded intelligence, wireless sensor networks (WSNs), and cloud computing. Sensors, cameras, radio frequency identifiers (RFID), and other devices are used by IoT systems to gather environmental data [2]. These systems are capable of providing sophisticated services such as remote management, online analytics, and real-time remote monitoring. The IoT has used a range of remote monitoring applications, including healthcare, smart manufacturing, smart homes, smart cities, and smart agriculture, to improve productivity and decrease costs [3, 4].

In addition to being a major force in the Internet of Things (IoT), Wireless Sensor Networks (WSNs) play a significant role in people's daily lives across a variety of sectors. Wireless Sensor Networks are widely employed in harsh environments and wide-range applications [5, 6] such as multimedia, underwater, terrestrial, and underground [7]. More specifically, the WSNs are applied in real-world applications such as military, agriculture, earthquakes, glaciers [8, 9], environmental issues, industry [10, 11, 12], and healthcare [13].

The gathering of data in WSN may be either event-driven (such as the detection of forest fires [16]) or time-driven (such as the detection of gas or oil leaks [17]), depending on the needs of the application (such as habitat monitoring [18], logging temperature and humidity in the canopy of potato

plants for precision agriculture [19]). The time-driven data-gathering approach that is known as Periodic Sensor Networks is taken into consideration in this study (PSNs). In PSN, each sensor node is responsible for regularly sending the data that it has detected about the monitored region to the sink.

In the future, periodic sensor networks (PSNs) will be one of the most critical parts of the WSN, and they will play a key part in people's lives due to their extensive use in a variety of applications [3,4].

These types of networks have received a lot of attention from researchers in the past 4 years. The characteristics of these PSNs differentiate them from other ad-hoc wireless networks. However, due to the widespread distribution of a large number of sensor devices in these wide area locations, these applications' most prevalent concerns and challenges are distribution deployment, design, and energy consumption. Each sensor's lifespan depends on its ability to perform various activities such as sensing, calculation, processing, and data transfer. Moreover, data transmission consumes more energy than other activities. The data in a query-driven approach is captured, kept locally, and sent to the right module when a certain piece of knowledge is needed from many places. Time-driven sensor data also referred to as periodic sensor data, is primarily utilized for monitoring certain phenomena, such as earthquakes, healthcare, and melting glaciers [14].

Academics have paid a lot of attention to Periodic Sensor Networks (PSNs). One of the most important research questions in PSNs is how to gather and reduce the huge amounts of data regularly in a way that saves energy and then sends them to the sink to make the network lifespan longer. Due to the limited duration of the sensors' batteries, energy-efficient data transmission reduction approaches are necessary for energy optimization. In addition, the

sensor nodes in PSN capture data from the physical environment continuously and transmit it to the base station. This will lead to the rapid depletion of the sensor nodes' batteries. No matter how fast or slow the environment changes, the data is the same or duplicated in both cases, which increases the cost of transmission [15].

1.2 Dissertation Scope

This dissertation studies different energy-efficient data reduction methods to prolong the lifetime of PSN in which the sensor nodes collect data periodically, data reduction based on prediction and data compression, and then transmit the compressed data to the sink via a single hop path. More specifically, our attention is concentrated on data collection, data reduction, and selective forwarding in PSNs to improve the network lifetime.

Furthermore, this study focuses on the data reduction problem, in which saving energy is an essential condition. To address this problem, this dissertation suggests a two-level data reduction approach. The first level is in the sensor node and the second is in the gateway for reducing the data redundancy and reducing the power consumed while keeping good data accuracy in the gateway and sink.

1.3 Problem Statement

Sensor nodes are severely restricted by limited energy, computation power, and memory resources. One of the biggest challenges in WSN is the lifetime maximization of the battery. Minimizing the quantity of data transmitted to the base station, also known as the sink, is a crucial consideration. Because of the data is captured by sensor nodes exhibit both temporal and

spatial correlations, identifying and reducing redundant sensed data are the exact problems. The need energy-efficient approaches to combine the sensed data into high-quality data, which reduces the volume of data to be transmitted to a base station, resulting in the preservation of energy and bandwidth. Despite many data reduction algorithms for reducing sensed data and improving the network lifetime in WSNs being proposed, most of them ensure high accuracy of data for the sensing field with minimum energy consumption.

1.4 The Motivation of the Dissertation

One of the biggest challenges in WSNs is reducing the sensed data of the monitored area of interest in an energy-efficient manner while keeping an appropriate degree of data accuracy since the WSN is one of the largest data producers in the IoT. To extend the life of wireless sensor networks (WSNs), it is necessary to reduce the amount of energy each sensor node uses by eliminating unnecessary data before delivering it to the gateway.

Transmitting/receiving the data by sensor devices is an expensive process, while in-network computations are much less expensive from the energy consumption point of view and are sometimes ignored as insignificant [20, 7]. However, as shown in Figure 1.1[7], the computation requires much less energy than the data transmission/receiving. The energy required to transmit a 1 KB data message over a distance of 100 m, for instance, is nearly similar to the execution of nearly three million instructions on a normal microprocessor. As a result, any extra processing that lowers the data size even by one data bit would save energy [20].

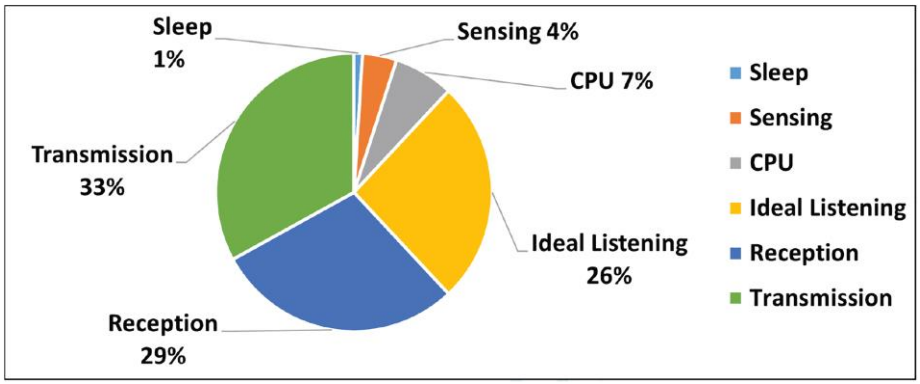


Figure 1.1: Energy Consumption Inside the Sensor Device

1.5 Aims of the Dissertation

1. The most critical resource in the sensor node of an IoT network that impacts the lifetime of the network is the energy provided by the battery.
2. Since the limited lifetime of the battery in the sensor node, it is difficult or impossible to replace (or recharge) it, especially in a remote or hostile environment. Since the WSN represents one of the big data contributors in the IoT, therefore, one big challenge in WSNs is to collect and reduce the sensed data of the monitored area of interest in energy-efficient way while maintaining a suitable level of data accuracy.
3. In order to improve the WSN lifetime, the principal idea is to exploit the advantage of the temporal and spatial data correlation among the sensor nodes to minimize energy consumption by removing the redundant sensed data before sending them to the FG.
4. Obviously, the elimination of redundant data from the sensor nodes is suitable if the received sensed data at the cloud node of the monitored area is not affected.

1.6 Main Objectives of this Dissertation

The following is a summary of the dissertation's primary aims:

1. Find out what problems the current algorithms for reducing data in WSNs have.
2. Obtaining and processing actual data from the sensor network installed in the Intel Berkeley Research facility.
3. Suggest a reduction of data approaches based on prediction and data compression that use less energy. Implementing this approach would result in a reduction in the quantity of data gathered and transmitted, thereby extending the longevity of the network.
4. A custom simulator based on the Python programming language and based on real observed data from sensor nodes. Different performance metrics, such as accuracy, energy consumption, network lifetime, etc., can be used to figure out the performance analysis.
5. Compare the proposed approaches with themselves and some recent research in the same field to show that the proposed approaches are better.

1.7 The Contributions of this Dissertation

The main contributions in our work concentrate on designing periodic energy-efficient in-network processing approaches that depend on data prediction, reduction, and compression approaches within two levels: Sensor nodes level and Gateway level.

1. In the first level (Sensor nodes), we design four energy-efficient approaches:

a. **DEDaR approach**: it executes the AutoRegressive Prediction (ARP) model to predict the data for the next period. It applied the data transmission reduction technique based on adaptive piecewise constant approximation (APCA), Symbolic Aggregate Approximation (SAX), and Huffman Encoding (HE).
b. **DiPCoM approach**: it uses an AutoRegressive Integrated Moving Average (ARIMA) model to predict the data for the next period, and it combines different data transmission reduction techniques based on APCA, Differential Encoding (DE), SAX, and LZW.
c. **IDaPCoT approach**: it uses an AR model to predict the data for the next period, and the data transmission reduction technique based on APCA, DE, SAX, and LZW
d. **EDaRePE approach**: it makes use of an ARIMA model to predict the data for the next period, and the data transmission reduction technique based on APCA, DE, SAX, and HE

2. In the second level (Fog Gateway):

Spatial correlated data reduction technique for energy conservation in the IoT network. We design an approach named **Two-Tier Energy-efficient Data Reduction Technique for IoT Networks (TEDaReT)** to reduce data before sending it to the cloud. The TEDaReT is executed to remove the duplicated sets of data received from the sensor node level to reduce the data sets based on finding the similarity between data sets.

1.8 Literature Review

WSN-based IoT data reduction has received much interest in recent years. Some of the most often used fundamental data reduction technologies include clustering, scheduling, compressive sensing, multi-channel multi-paths, data compression, data aggregation, and prediction. In this section, certain a number of those alternatives will be presented with details explanations and one will explain why they work.

In a previous study [21], the authors suggested combining the Gaussian process for robust prediction with a wavelet multiresolution transform. The wavelet domain handles data and allows the transform to capture geometric information and break down it into smaller signals or sub-bands. For each sub-band of the wavelet, the deconstructed signal is estimated using a Gaussian process, allowing the processing of Gaussian to pick up a much simpler signal. Using difficult time series generated by several types of sensors, [22] built a multi-dimensional attribute selection model and a prediction model of sensor data reactive. This methodology enhances the accuracy and consistency of IoT sensor data long-term prediction results when compared with existing data prediction models. The prediction model was set to the test with sensor data from Intel Berkeley Research Lab, which have an accuracy of more than 98%, and weather and water data from the Chicago Park District, which have an accuracy of 92%.

As a solution for reducing IoT data, the use of in-networking data filtering and fusion is proposed in [23]. The suggested method is split into two levels, each of which may be adjusted on a single or two tiers. The data change detection and the divergence of true observations from their predicted values are two methods that refer to the initial layer of the data filtering layer which

was introduced by the proposed method. The second layer uses a least square error criteria to merge the data in the same certain region at the same time domain for individual sensors.

To decrease the data size, an adaptive method called AMDR was presented. This method works on two levels: the first level is represented by the sensor node and the second level is represented by the gateway [24]. This method is used to reconstruct the data based on the threshold determined by the user in advance to maintain the accuracy of the data. In some cases, AMDR gets as good a result as 92% for data reduction while maintaining good forecast accuracy, according to real-world data. AMDR offers the realization of all energy-saving possibilities.

Some proposed prediction algorithms [25-28] examined the relationship between gathered data to create a model that compares historical and future values. Liazid et al. [26] recommended modifying the dual prediction scheme (DPS) mechanism. Rather than updating the historical data table, the new version applies a series of models for data prediction during earlier DPS algorithm runs. In reality, the improved model of prediction is generated in (SN) and sent to the gateway, or vice versa.

Russo et al. [27] proposed an unsupervised machine learning system based on self-organized maps to predict data from sensor nodes. They demonstrate a new predictive method that puts the sensor into hibernation to reduce data transmission based on a self-organizing map. In Karjee and Kleinsteuber [28], to track faulty data gathered at the sink, an online data tracking and estimation (ODTE) system is proposed. The data prediction system (DPS) and the distortion factor are the two basic systems used in ODTE (DF). DPS is used at the sensor level to decrease transmission by setting a limit, whereas DF

estimates the best data collected at the sink node. Compression and aggregation techniques are used in several of the suggested works to minimize data transmission. For periodic sensor applications based on clusters, the authors offered a structure fidelity data collection (SFDC) method in Wwu et al. [29] SFDC searches for spatial correlation between sensors using a distance function and temporal correlation using a similarity index. The authors then present a scheduling strategy to provide power to the sensor in wake/sleep nodes in clusters. Similarly, Dhimal and Sharma [30] used a similarity method to search the sensor nodes for spatial-temporal correlation to switch the associated nodes to sleep mode to reduce power in the network. The authors demonstrated that, when compared to alternative similarity metrics, PPMC provides the greatest results in terms of network energy conservation.

Previous studies [31, 32] proposed a modified k-nearest neighbor algorithm for data redundancy removal in the sensor node to save energy and extend the lifespan of the network. Then, they extend their work to include data redundancy elimination on the second level of the network (gateway). The received data vectors of sensor nodes are gathered into groups of similar data vectors and then one representative vector is sent for each group. In Idrees et al. [33] the authors introduced a new data reduction method for saving energy in the IoT network. They implemented two algorithms to remove the redundant data at both the sensor and aggregator levels. The divide and conquer method is implemented at the aggregator level while the clustering is used with the sensor nodes. The proposed work in Idrees and Al-Qurabat [34] presented an energy-aware data transmission reduction in fog computing-based IoT networks. The method is activated at both sensor nodes and the fog gateway. In the sensor devices, they implement combined grouping and easy encoding methods to remove unnecessary data before forwarding them to the fog gateway. In the fog

gateway, they proposed a clustering algorithm based on “dynamic time warping” (DTW) that combines with the simple encoding method to further remove the redundant data before transmitting them toward the cloud data center.

Wang et al. [35] suggested two phases of dual prediction to decrease the amount of data which are (DPP) which refers to the data prediction phase and (DRP) which refer to the data reduction phase which decreases the volume of data sent to the gateway and saves power in the network, while DPP processed on the gateway in synchronization with previous level (SN) to predicted non-transmitted data in the previous level. Jarwan et al. [36] produced two variant algorithms: data compression (DC) which use to minimize the traffic between the gateway and based station and dual prediction (DP) which utilize to decrease transform of data to the gateway, they proposed NN network and long short-term memory networks (LSTMs) for runs the prediction. AM-DR proposed by Fathy et al. [37] depended on merging two decouple LMS of filtering of the window that has the variant size to find the measurement for two-level (SN) and (GW), which makes (SN) send just the immediate value that considered irregular value.

Several data reduction algorithms based on clustering techniques have been proposed in the past few years [38–46]. The PFF strategy is used in the sensor and aggregator devices [38]. The Jaccard similarity is used by researchers in the sensor node to lower the redundancy of data and set similarity in the aggregator node to minimize duplicated sets of data. Harb et al. [39] described an approach called ATP that was implemented in the sensor device. It lowers the amount of data before transferring them to the base station. They

eliminate the redundancy of data at the sensor node, then use a variety of techniques to lessen spatially similar data in the gateway.

The authors in [40] suggested a de-redundancy algorithm that works at two levels. The first level produces a dual-metric distance, and to obtain clusters of similar nodes, the enhanced k-means method first identifies redundant nodes. In the second step, a Gaussian hybrid clustering classification technique is provided, which is used to build data similarity clusters for edge sensing data. The clustered data is randomly weighted in the third step to deduplicate the spatial correlation data. The EK-means strategy presented by Rida et al. [41] is a two-step approach. First, it uses a Euclidean distance-based data aggregation approach to reduce similar data collected at the sensor level. Then, it uses an improved k-means clustering algorithm to aggregate similar data sets generated by surrounding nodes into the same clusters, reducing the quantity of data transmitted to the sink even more.

Loganathan et al. [42] proposed a data reduction method based on aggregation and re-scheduling using clustering techniques to save energy in sensor networks. The cluster formation and cluster head selection are implemented in an energy-efficient way. This model contributes to reducing the transmitted data and increasing the lifetime of the network.

Alam et al. [43] proposed an in-network data-lowering approach at the cluster head using an error-aware data clustering scheme. This approach allows the user to select the suitable model that satisfies their requirements and the quality of data. The temporal data is clustered into groups, and the data redundancy is removed from each group. The random outliers are detected to introduce an error-aware data clustering that maintains the level of data error under a predefined threshold. For data gathering, a correlation clustering

strategy is applied, in which sensor nodes with similar data are grouped as a cluster. Then, using independent component analysis on cluster head IoT devices, an algorithm is developed to gather the data. The data-gathering phase is carried out on clusters with higher data correlation [44]. A previous study [45, 46] suggested an integrated divide and conquer with an improved K-means strategy for data gathering with power saving in WSNs. It gathers the data at two different levels: node and cluster head. At the sensor node, a divide and conquer technique is used to eliminate data redundancy from the gathered data before transferring it to the head of the cluster. The head of the cluster uses an improved K-means strategy to a group obtained data sets from IoT devices into groups of near-identical data sets and then sends the best representative set from each group to the sink.

The comprehensive resume of related literature mentioned above indicates many methods for data reduction using different techniques. Nevertheless, the proposed methods in existence cannot adequately eliminate redundant data while keeping a high level of data accuracy. Furthermore, some of these methods are more complicated and require high time and memory complexities, and they cannot be implemented inside the constrained resources sensor devices.

No.	Reference	Level	Approach	Objective
1	Jose Mejia et al. [21]	SNL	Combines a wavelet multiresolution transform with robust prediction using Gaussian process	present an algorithm for the prediction of time series, with which it is expected to reduce the energy consumption of a sensor network, by reducing the number of transmissions when reporting to the sink node only when the prediction of the sensed value differs in certain magnitude, to the actual sensed value

2	Cong Zhang et al. [22]	SNL	multi-dimensional feature selection model and a dynamic sensor-data prediction model.	Use the complex time series formed by various types of sensors to establish a multi-dimensional feature selection model and a dynamic sensor-data prediction model.
3	Ibrahim Kok et al. [103]	SNL	Propose a missing data prediction protocol called DeepMDP	The proposed protocol can work on resource-constrained IoT devices as well as fog and cloud servers. propose a missing data prediction protocol called DeepMDP for IoT systems with unreliable data sources, which can reduce the amount of data transmission and delay in the network significantly
4	Mohammad Mehrani et al. [102]	SNL	Adaptive Neuro Fuzzy Inference System (ANFIS) & Long Short Term Memory (LSTM)	Using forecasted sampling frequencies of the biosensors for controlling their energy expenditure, these forecasted values would also be used to forecast patient's status in the future
5	Azar J. et al. [83]	SNL & FGL	lossy compressor & supervised machine learning techniques	apply a fast error-bounded lossy compressor on the collected data prior to transmission, that is considered to be the greatest consumer of energy in an IoT device. In a second phase, rebuild the transmitted data on an edge node and process it using supervised machine learning techniques.
6	Khushboo Jain et al. [104]	SNL	Prediction Based Data Aggregation Technique	a prediction model based on Extended Cosine Regression (ECR) for Data Aggregation is proposed
7	Anoop Kumar et al. [101]	SNL	extended linear regression model	The purpose of the proposed model is to exempt the sensor nodes (SN) from sending huge volumes of data for a specific duration during which the BS will predict the future data values and thus minimize the energy utilization of WSN. The study suggested an extended linear regression model, which determines resemblance in shape of data curve of contiguous data periods.

8	Ghina Saad et al. [100]	SNL & FGL	Periodic Data Collection Model & a prediction model based on the long short-term memory (LSTM)	each sensor collects data for some periods of the round then it sends them to the next node then, it enters into sleep mode for the other periods of the round. Upon receiving the data from each sensor, the sink uses a prediction model based on the long short-term memory (LSTM) time series in order to expect sensor data during the sleeping mode
9	Waleed M. Ismael et al. [23]	SNL & FGL	Data Filtering Layer & Data Fusion Layer	The proposed approach consists of two layers that can be adapted at either a single tier or two tiers. The first layer of the proposed approach is the data filtering layer that is based on two techniques, namely data change detection and the deviation of real observations from their estimated values. The second layer is the data fusion layer. It is based on a minimum square error criterion and fuses the data of the same time domain for specific sensors deployed in a specific area.
10	Yasmin Fathy et al. [24]	SNL	Adaptive Method for Data Reduction (AMDR)	describe an Adaptive Method for Data Reduction (AMDR), a data reduction approach for reducing the overall data transmission and communication between sensor nodes in IoT networks such that fine-grained sensor readings can be used to reconstruct the original data within a user-defined accuracy boundary.
11	Wang, H. et al. [35]	SNL	DPDR approach	Proposes a reliable dual prediction data reduction approach for WSNs. This approach performs data reduction through two phases: the data reduction phase (DRP) and data prediction phase (DPP).
12	Jarwan A. et al. [36]	SNL & FGL	an Event Clustering Routing Protocol based on Consensus (ECRPC)	present both schemes in a two-tier data reduction framework. The DP scheme is used to reduce transmissions between cluster nodes and cluster heads, while the DC scheme is used to reduce traffic between cluster heads and sink nodes. For both schemes, various algorithms will be studied and compared in terms of accuracy, delay, and transmission reduction percentage. For the DP scheme, neural networks (NNs) and long short-

				term memory networks (LSTMs) are proposed to perform predictions. The training phase of the NNs and LSTMs is done online which is necessary in the DP scheme.
13	Fathy Y. et al. [37]	SNL	Combination of two decoupled Least-Mean-Square (LMS)	propose an Adaptive Method for Data Reduction (AM-DR). The method is based on a convex combination of two decoupled Least-Mean-Square (LMS) windowed filters with differing sizes for estimating the next measured values both at the source and the sink node such that sensor nodes have to transmit only their immediate sensed values that deviate significantly (with a pre-defined threshold) from the predicted values

1.9 Dissertation Organization

The rest of this dissertation is arranged as follows:

- **Chapter 2: Theoretical Background**

  This chapter presents the theoretical background of WSNs. An overview of PSNs, their applications, and challenges is made as a way to give a background to the PSNs and the challenges of PSNs. Also, this chapter illustrates some of the prediction techniques, reduction methods, and data compression approaches. Furthermore, present the metrics of data reduction such as network lifetime, data accuracy, energy consumption, and data transmissions. Moreover, present the similarity measurements. Finally, illustrate the energy consumption model.

- **Chapter 3: Proposed Data Reduction Approaches**

  This chapter illustrates the designing of periodic energy-efficient in-network processing approaches that depend on data prediction, reduction,

and compression approaches within two levels: Sensor nodes level and Gateway level. In the first level (Sensor nodes), we design four energy-efficient approaches: the **DEDaR approach, DiPCoM approach, IDaPCoT approach,** and **EDaRePE approach.** In the second level (Fog Gateway): we design an approach named **Two-Tier Energy-efficient Data Reduction Technique for IoT Networks (TEDaReT)** to reduce data before sending it to the cloud.

- **Chapter 4: Simulation Results and Discussion**

  The assessment of the performance as well as the results of the simulation are shown here in the form of graphs, and a discussion of the recommended approaches that were provided in Chapter 3 is included. The evaluate the performance is twofold: first, to evaluate the performance of the suggested approaches using actual sensor data and a variety of performance measures such as data reduction, number of sending readings, energy consumption, and data accuracy; and second, some performance metrics of further results are applied to assess the effectiveness of the proposed approaches, such as transmitted readings, energy consumption, and data loss percentage. Finally, the evaluation of the offered approaches in relation to other competing approaches that belong to the same area.

- **Chapter 5: Conclusion and Future Works**

  This chapter presents the conclusion for the dissertation which is drawn from the findings which came from conducting this research and recorded in the chapters. Finally, future works are also introduced in this chapter.

CHAPTER TWO

THEORETICAL BACKGROUND

CHAPTER TWO: THEORETICAL BACKGROUND

2.1 Introduction

Wireless Sensor Networks (WSNs) are widely employed in harsh environments and in wide-range applications [5, 6] such as multimedia, underwater, terrestrial, and underground [7]. More specifically, the WSNs are applied in real-world applications such as military, agriculture, earthquakes, glaciers [8, 9], environmental issues, industry [10, 11, 12], and healthcare [13]. However, due to the widespread distribution of a large number of sensor devices in these wide area locations, these applications' most prevalent concerns and challenges are distribution deployment, design, and energy consumption. Each sensor's lifespan depends on its ability to perform various activities such as sensing, calculation, processing, and data transfer.

Moreover, data transmission consumes more energy than other activities. The data in a query-driven approach is captured, kept locally, and sent to the right module when a certain piece of knowledge is needed from many places. Time-driven sensor data also referred to as periodic sensor data, is primarily utilized for monitoring certain phenomena, such as earthquakes, healthcare, and melting glaciers [14]. Academics have paid a lot of attention to Periodic Sensor Networks (PSNs).

This chapter describes the periodic sensor network. PSN application samples are reviewed first. Then, we introduce the challenges that face PSN. Next, we provide the techniques and performance measurements for Predictions, compressions, and data reduction. After that, we introduce the

network lifespan and time series data mining. And finally, we provide this work's energy model.

2.2 Overview of WSN

Recent rapid growth in one of the most significant applications in WSNs is environmental monitoring, which is highly recommended in the wake of certain climatic shifts and catastrophic disasters in nature throughout the globe. The uses for WSN may be broken down into three distinct classes. First, event-driven applications, where sensors report back to a central hub when something noteworthy has occurred, like a forest fire. The second form of application is query-driven, whereby the information collected by the sensor nodes is transmitted in reaction to a request from the sink, which in this instance is a storage facility. Third, this thesis focuses on time-driven (or periodic) types [47].

A wireless sensor network is said to be periodic if its sensor nodes deploy at regular intervals to gather data from the area of interest and then periodically report that data to a central location, or sink. PSNs, in contrast to other networks, are exceptionally well-suited to ensuring persistent data collection in the relevant area [48]. The periodic method is used by a number of PSNs applications (such as environmental monitoring [49] and phenomenon surveillance [50]) to keep tabs on recurring changes. Sensor nodes are used in extreme environments to collect data on the weather and hydrology on a regular basis. This data includes things like pressure, light, humidity, wind speed and direction, temperature, and so on.

PSN has two primary challenges: first, it must be able to offer an appropriate lifespan in order to fulfill the requirements of applications. Second,

as a result of the enormous quantity of data that is gathered by this network, managing the data is made more complex [51].

The implementation of PSN for agricultural applications is shown in Figure 2.1. A large number of sensor nodes—in the hundreds—have been randomly dispersed around a field to determine the levels of light, temperature, and humidity in the soil. Every five minutes, all of the sensor nodes in this system collect the same piece of data. Once each hour, the collected information is sent through multi-hop transmission to the final recipient for analysis [47].

Figure 2.1. Agricultural uses of PSNs [47].

2.3 The WSNs Applications

WSNs include many different types of sensors, such as infrared, visual, seismic, acoustic, thermal, magnetic, and radar. The sensors possess the capability to identify and communicate a diverse array of environmental variables such as pressure, motion, orientation, temperature, illumination, velocity, flooring composition, acoustic intensity, the existence or non-existence of particular objects, and the mechanical strain levels of linked entities [20]. As can be seen in Figure 2.2, this opens up a wide range of possible applications.

This variety of applications includes homeland security [20], monitoring the environment, monitoring the structural health of structures (such as bridges, tunnels, and buildings), tracking targets, monitoring pipelines (such as water, oil, and gas), monitoring active volcanoes, precision agriculture, health care, transportation, supply chain management, underground mining, human activity monitoring [7], monitoring seismic acceleration, intelligence gathering for defense, weather, and climate, and monitoring seismic acceleration.

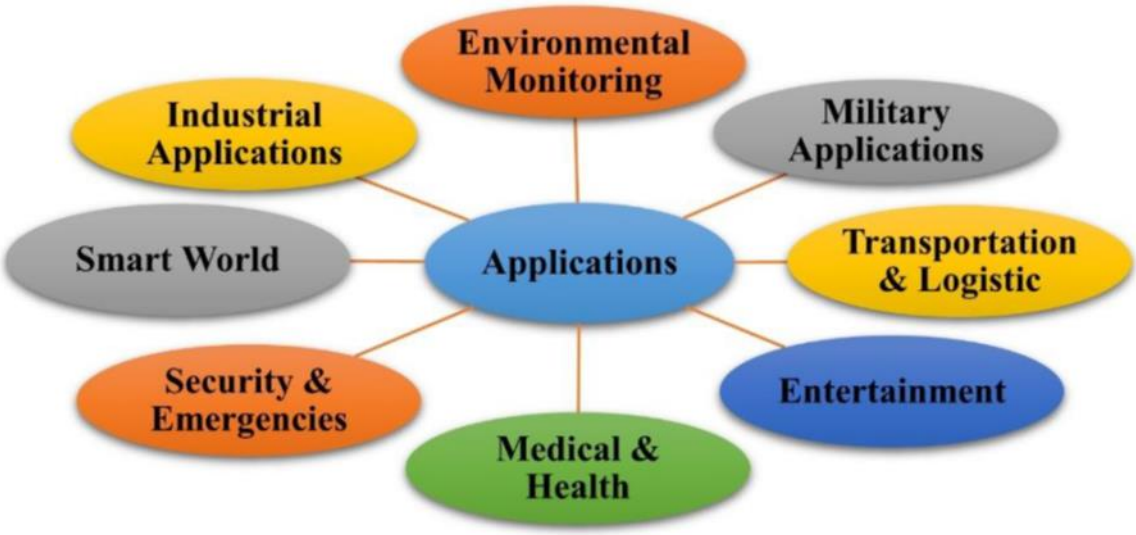


Figure 2.2. Applications taxonomy of WSN.

In the near future, it is expected that sensors will be embedded in everything to give them intelligence. Intelligent objects have the capability to acquire knowledge about their environment, engage in communication with other intelligent entities, and even establish connections with individuals. Consequently, the market for Wireless Sensor Networks (WSNs) has witnessed a significant expansion and paved the way for the emerging "Internet of Things (IoT)" [52].

The Internet of Things (IoT) is a network of devices that are capable of independent thought and are linked together by a means of electronic communication. Since sensor nodes form the backbone of the Internet of Things, WSNs are poised to play a pivotal role in this emerging field [53]. With the rise of the IoT, wireless sensor networks (WSNs) have found widespread usage in a variety of modern contexts. This encompasses a range of technologies, such as remote metering and smart water supply, useful agriculture, and smart farming, among others [54].

2.4 Challenges in PSNs

The PSN has the capability to gather data from its sensor node constituents, which are widely distributed in large areas, and subsequently transmit that data to the sink at regular intervals. This is one of the features that sets PSN apart from other networks. PSNs will be subject to several problems (some of which are shown in Figure 2.3), all of which will have an impact on their overall design, operational effectiveness, and operational performance. In the following sections, specific information on PSNs' most significant difficulties will be provided.

Figure 2.3: The PSNs Challenges.

2.4.1 Deployment

The initial challenge that Periodic Sensor Networks (PSNs) may encounter pertains to the installation of sensors with the objective of collecting data on a particular phenomenon within the designated area of concern. This is because sensors constitute the initial step in the network life cycle [20]. It is possible to divide the process of deploying sensor nodes in WSNs into two distinct types: the first of these is a deterministic deployment, while the other is a random deployment. Furthermore, while some applications may offer deterministic sensor placement, random placement is frequently utilized in the majority of scenarios due to its convenience in terms of both time and cost. In order for a PSN application to be successful, it is necessary for the sensors to cover the whole area and to have a link to the communication network [55].

2.4.2 Energy Consumption

Two of the challenges that are frequently associated with the design of PSNs: first is the deployment of the sensors in different, and second is limited power sources for the sensors that are representative of the core of PSN's work [17]. In general, the power source for sensor nodes is a battery that requires

recharging after a certain period of time or replaced since its energy storage capacity has been depleted. It may be difficult or impossible to replace or recharge the sensor batteries in some circumstances, such as when resources are costly or when sensor nodes operate in a hostile environment, harsh, or distant [56]. In these instances, none of the two choices is appropriate.

When its power supply runs out, though, it may be easily discarded. Sensor nodes with non-rechargeable batteries must be able to keep going until the job is complete or a new battery is installed. The duration of the work is based on the nature of the application, and PSN must have a long enough lifespan to meet the needs of the application [7]. Thus, energy efficiency is often the most critical task for a PSN.

2.4.3 Security

PSN also has the difficulty of ensuring the safety of data collected in non-military settings, and mission-critical infrastructure like airports and hospitals [56]. Although security is a concern in all networks, it is more pressing in sensor networks because of their mobility and limited resources. Conventional methods of protecting networks of sensors are impractical (i.e., they are wholly inadequate due to the specific features and application requirements for sensor networks) due to the restricted resources of sensor nodes (such as energy and memory) [57]. As sensor nodes are often placed in vast, uncontrolled regions, they are open to a wide variety of threats and attacks from humans and animals alike. Both the sensors' physical and data security, as well as the needs of the monitored application, are required to be in place, therefore, be taken into account throughout the system's design and construction [57].

2.4.4 Scalability/density

The density diffusion of sensors presents a barrier when it comes to scalability, notwithstanding the fact that the deployment of a large number of sensors may result in redundancy, it can enhance the fault tolerance of the network. It is very uncommon for there to be hundreds or even thousands of sensors dispersed around an area with the aim of monitoring a particular physical occurrence [20]. The necessity of this duplicative volume has been established in order to ensure the reliability of sensor data acquired through collaborative means. The presence of such enormous amounts of data will lead to the emergence of a number of issues within the PSNs, including but not limited to: an increase in the degree to which data is redundant; the network packets colliding with one another; required energy consumption has increased; and a more difficult task for those responsible for making decisions. Hence, in the process of building networking protocols, they need to be capable of dealing with a high number of sensors in an effective way [20]. As a result, the processes of data collecting and data reduction have received a significant deal of scrutiny in the field of PSN with the aim of diminishing the volume of generated data.

2.4.5 Routing

One of the additional issues that may be presented to networks of periodic sensors is the task of locating and maintaining pathways that connect sensor nodes and sinks. There are many limitations that are imposed on the design of routing protocols. These limitations are dependent on the capabilities of nodes (the limitations of wireless networks include a restricted range of transmission, limited energy capacity, and constrained processing and storage capabilities. Additionally, the network's characteristics, such as self-configurability, sensor locations, node identifications, fault tolerance, and topological changes, also

play a significant role). On other hand, numerous other routing methods for PSNs have been suggested in the research that has been done. The majority of these protocols make use of several established routing strategies, for instance, clustering, data reduction, and in the domain of network processing, in order to reduce the amount of power that they use. [58, 59] These techniques may help reduce the amount of power that is required.

2.4.6 Coverage

Keeping the best possible coverage of the target region is another significant issue for periodic sensor networks. After sensor nodes have been set up, we need to make sure they provide enough sensing coverage over the network's whole lifespan. Nonetheless, environmental monitoring and other periodic sensor network applications provide considerable leeway in terms of network coverage. Critical applications like industrial or military surveillance need comprehensive coverage of the target region. Hence, sensor scheduling algorithms have been shown to be an efficient methodology for periodic sensor networks when considering the issue of coverage [60, 61].

2.5 Data Collections

Data collection refers to the act of amassing sensed measurements in a methodical manner from a large number of sensor nodes with the purpose of finally transmitting them to the sink for further processing [20]. In contrast, in-network calculations are less expensive in terms of energy consumption and are frequently regarded minor and inconsequential [20, 7]. Data transfer is an expensive process in the sensor. On the other hand, the power required to do the computation is far lower than the power required to communicate the data. For instance, the amount of energy that must be used in order to transmit a data

packet with a size of 1 KB across a distance of 100 meters is about similar to the standard microprocessor's ability to carry out around 3 million individual instructions. Hence, any extra processing that will lower the quantity of the data, even if it's only one data bit, will be valuable in terms of finding ways to save energy. This high fluctuation in transmission and calculation demonstrates the relevance of using local data processing to reduce the energy volume within the sensors network that is used [7].

It is common for nearby sensors to produce data that is highly correlated with one another and redundant. [48] The reason for this is that sensors that are placed in close proximity to one another typically sense the same phenomena, which results in the generation of a large amount of duplicate data. The amount of bandwidth and energy that will be used as a direct consequence of this data replication will be excessive [62]. Moreover, the station will be faced with a significant challenge in vast sensor networks that managing the data generated [48].

It is a waste for each of the nodes to transmit their data straight to the sink because sensor nodes have a finite amount of energy. In addition, the decrease of energy consumption in each sensor node has become important in order to increase the lifespan of WSN [62]. Therefore, we need strategies for eliminating redundancy and consolidating data in order to produce a piece of information with high quality at the sensor nodes. This will result in a reduction in the number of packets that will be sent to the sink, which will ultimately lead to the conservation of both energy and bandwidth. Data reduction is one method that may be used [48] to accomplish this goal.

This dissertation focuses on the problem of reducing data, where saving energy is a very important goal. To address this problem, the dissertation proposes two-level data reduction strategy based on prediction and compression

that will be executed at the sensor device level and Gateway level to lower data redundancy and power consumption while maintaining high data accuracy. This section will provide some theoretical context for certain approaches.

2.6 Prediction Techniques

The term "prediction" is used to describe the process of making an educated guess about a future result or occurrence by analyzing available facts and information. Prediction is the process of utilizing past data to predict future outcomes based on patterns or trends detected via statistical or machine learning analysis [63].

Prediction may be made in many different industries, including business, economics, meteorology, sports, and medicine. Experts in several fields utilize historical data to make forecasts about the future, such as stock market analysts, meteorologists, and physicians who try to gauge a patient's risk of contracting a disease based on their medical history. As many circumstances, some of which cannot be predicted in advance, might affect the course of future events, predictability is not guaranteed. Yet, the capacity to create accurate forecasts may be a powerful resource for decision-making and planning, enabling companies and organizations to foresee future events and take preventative or proactive steps as needed [63].

2.6.1 The Autoregressive Prediction

The AutoRegressive Prediction (ARP) prediction technique is main aim is to predict the data in the next period in sensor nodes. Autoregressive statistical models predict following values by the history of previous values. An autoregressive model may attempt to stock market prices based on its previous performance for example:

1. Autoregressive models estimate future values based on historical values and are commonly employed in technical analysis to forecast future security prices.
2. The underlying assumption in autoregressive models is that the future will be similar to the past. As a result, they may be wrong under specific market scenarios, such as financial crises or rapid technological growth.

Because they work on the principle that prior values have an influence on present values, autoregressive models are helpful for analyzing nature, economics, and other time-varying systems. Autoregressive models use an accumulation of the variable's prior values, whereas multiple regression procedures use a linear combination of predictors to predict a variable [64].

The AR(1) autoregressive process's current value is determined by the value immediately preceding it, whereas an AR(2) process's current value is determined by the previous two values. For electronic noise, an AR(0) process with no term dependency is used. There are many various techniques to determine the coefficients suitable calculations used, such as the least squares method, in addition to these variations. Technical analysts use these concepts and methodologies to forecast security prices. However, because autoregressive models rely solely on historical data to predict future values, they implicitly presume that the fundamental forces that drove past prices will remain constant over time. If the underlying dynamics at stake are fluctuate, for example if a particular industry is undergoes rapid and unprecedented technological development, this might lead to unexpected and incorrect forecasts. To create forecasts, an autoregressive model uses a linear combination of the target's past values. Naturally, the regression is performed against the target. An AR(p) model is mathematically stated as [65]:

$$x_t = c + \sum_{i=1}^{p} \emptyset_i \, x_{t-i} + \varepsilon_t \qquad \textbf{(2.1)}$$

Where, p: is an order, c: is a constant, and epsilon: noise. The AR(p) model is extremely adaptable, and it may be used to describe a wide range of time series patterns. Autoregressive models are often used only on stationary time series.

2.6.2 The ARIMA Prediction

It is a statistical analysis approach known as Autoregressive Integrated Moving Average (ARIMA). In order to better understand the data set or forecast future changes, this model analyzes time series data. A statistical model known as an autoregressive one estimates future values using historical data. For example, an ARIMA model can try to predict sales income based on what happened in the past or the price of a stock based on what happened in the past. There are three main parts to the ARIMA method [66]:

- In terms of time series analysis, autoregression (AR) is the process of making predictions based on prior data. The idea behind an AR model is that the current value of a time series may be calculated by looking at its historical data.
- The time series is integrated (I) by differentiating it until it is stationary. When a time series is stationary, it has a constant mean and constant variance. To do a differentiation, one simply subtracts the value of a time series at a given instant from its value at an earlier instant.
- The term "moving average" (MA) is shorthand for a method that makes predictions based on the average of previous forecast failures. The value of a time series at any instant in time is assumed to be proportional to the total of all the mistakes in the preceding predictions in an MA model.

Time series may be modeled using ARIMA. This model is linear because future values must be linear functions of past data. Over the last few decades, academics have focused heavily on linear models due to their ease of grasp and application. Demand is typically forecast using time series forecasting methods. The examined prediction accuracy under an autoregressive moving average assumption and utilized historical data to evaluate seasonal variation in demand [67]. To improve the predictability of their model, Miller and Williams [68] utilized a multiplicative method to identify the seasonal components. Hyndman [69] extended Miller and Williams' [68] work by usi ng alternate correlations between seasonality and trend in the context of the theory of seasonal ARIMA. If the order of the seasonal adjustment is large or the diagnostics fall short of proving that the time series is stationary after the seasonal adjustment, the standard ARIMA approach becomes costly.

Furthermore, it is frequently impossible to create a model. In these cases, the traditional ARIMA model static parameters are seen as the principal obstacle to forecasting the rapidly varying seasonal demand. The traditional ARIMA method also has the problem that it needs a lot of observations to find the best model for a set of data [70]. ARIMA model (p, d, and q) is the name given to an ARIMA model in which:

- p stands for the autoregressive amount,
- d for the differences amount,
- q for the moving averages amount.

According to equation (2), and the autoregressive model, Y_t is a linear function of the preceding values:

$$\boldsymbol{Y_t = \alpha_1 Y_{t-1} + \varepsilon_t} \qquad \textbf{(2.2)}$$

A random shock ($\boldsymbol{\varepsilon}$) and a linear mixture of prior observations have been combined to create each observation. The self-regression coefficient, $\boldsymbol{\alpha}1$ is shown in equation (2). The integrated approach considers how multiple processes interact cumulatively to affect how a time series behaves [71]. For instance, the status of stocks is constantly shifting according to demand and supply. Nonetheless, the overall influence of the fleeting changes throughout the course of time between inventories largely determines the average level of stocks. The long-term level of the series will remain constant even if short-term stock prices vary dramatically around this average number. The class of integrated processes includes time series that are determined by the cumulative impact of an action [71].

For a process seen over a long period of time, a series' behavior might be chaotic even when the variations between observations can be extremely tiny or even revolve around a fixed value. A crucial characteristic for an integrated process from the perspective of statistical analysis of time series is the stationary nature of the series of differences. Non-stationary series are represented by integrated processes. It is assumed that there is a constant difference between any two sequential values of Y while doing an order one differentiation [72]. Equation (3) defines an integrated process:

$$\boldsymbol{Y_t = Y_{t-1} + \mathcal{E}_t} \qquad \textbf{(2.3)}$$

A random disturbance of white noise is εt.

A moving average process' current value is created by linearly combining the present disturbance and one or more preceding disturbances [69]. The moving average order reveals how many earlier periods were taken into account

while calculating the present value. The following is a definition of a moving average provided by equation (4):

$$Y_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} \qquad (2.4)$$

2.7 Reduction Techniques

When discussing time series analysis, the term "reduction" most often refers to the process of simplifying or lowering the dimensionality of the time series data while maintaining as high a level of data accuracy as is practicable. Methods such as aggregation, sampling, and filtering are examples of approaches that may be used to accomplish this goal. The practice of aggregating data over a predetermined period of time is a typical method that may be used to reduce the dimensionality of time series data. For instance, rather of utilizing data on a minute-by-minute basis, one might aggregate the data to hourly or daily intervals rather than using minute-by-minute data. This would make the data more digestible and simpler to evaluate [73].

Filtering is yet another method that may be used to alleviate the difficulties associated with time series data. This requires performing mathematical operations on the data in order to filter out noise or undesirable variability in the data while maintaining the integrity of the data's essential characteristics. For instance, you may apply a moving average filter to uncover underlying patterns in the data while also smoothing out oscillations in the data [74].

In the process of lowering the dimensionality of time series data, sampling is another method that is often used [75]. This entails picking a subset of the data points to utilize for analysis, which may lead to a reduction in the total quantity of data while still preserving a significant amount of information. Nevertheless, care must be taken to ensure that the sample is representative of

the underlying population and that essential aspects of the data are not lost in the process of sampling [76]. In addition, care must be made to ensure that the sample is representative of the underlying population.

Generally, the objective of time series reduction is to simplify the data without omitting any significant information and while preserving data accuracy with high quality [77]. This is done with the intention of making the data easier to work with for the purposes of analysis and modeling. There are some approaches that are used to reduce the data, and we introduce some of these:

2.7.1 Adaptive Piecewise Constant Approximation (APCA)

The APCA segments of time series can be obtaining from separate time series into sets of varying value segments (with a limited reconstruction error) by using APCA with different lengths depending on the data, with the purpose of reducing individual reconstruction mistakes. In more technical terms, | R (S^{AP}) – S |< ε, R (S^{AP}) is the reinitialized function, and ε is an error threshold. One of the basic components of data analysis is data sorting. It enhances the search and combines the sequences more effectively [34].

As a result, sorting the detected humidity values in descending order improves the efficiency of APCA by grouping the same (or nearly identical) signals together. Long segments represent low-activity data regions, whereas short segments represent high-activity data regions [34]. The (S^{AP}) is given as follows:

$$S_j^{AP} = \{(dm_1, dr_1), ..., (dm_m, dr_m)\}, dr_0 = 0 \qquad (2.5)$$

S_j^{AP} is given by a pair of numbers (dm_j, dr_j), where a mean value of humidity measurements is dm_j in the j^{th} segment, which is given [39]:

$$\mathbf{dm_j} = \frac{\left(\sum_{k=dr_{j-1}}^{dr_j} S^k\right)}{\mathbf{dr_j - dr_{j-1}}} \qquad \mathbf{(2.6)}$$

Where right endpoint of the j^{th} segment is dr_j.

2.7.2 The SAX Representation

SAX refers to a method of representation of data of time series. By using symbols set to refer to a time series of length p into q, where the length of a time series is q_p. To begin, the time series is normalized, converting the standard deviation to unity and the mean to zero [78]. This process ensures that input data are transformed to output series with a mean approximating 0, and a standard deviation approximating 1. This normalization is required so that the data reduction algorithm can focus on structural similarities and differences rather than amplitude. The following formula is used to calculate this normalization:

$$\boldsymbol{\mu} = \frac{\sum_{i=1}^{p}(S_i)}{p} \qquad \mathbf{(2.7)}$$

$$\boldsymbol{\sigma} = \sqrt{\frac{\sum_{i=1}^{p}(Si-\mu)^2}{p-1}} \qquad \mathbf{(2.8)}$$

$$S_i' = \frac{S_i - \mu}{\sigma} \qquad \mathbf{(2.9)}$$

Where σ, μ *and* S_i' are the standard deviation, mean, and normalized data readings.

The SAX is regarded as a pioneer in lowering time series dimensionality and numerosity. It is divided into two parts: the Adaptive Piecewise Constant Approximation (APCA) transformation and the numerical data transformation into a set of symbols. A finite alphabet determines the value of each symbol [42].

The following steps are used to convert the APCA considering S^P is the subset of the original time series S into the SAX (S^X):

1- Divide the series of reading values into w parts.

2- For each component of the readings, calculate mean.

3- From N-letter in alphabet, draw mean value quantized into symbols.

The APCA representation is used in the first two steps. In step 3, employment of quantization (*N-1*) breakpoints to divide the Gaussian distribution's domain into proportionate regions equal *a*. A list of sorted numbers $\beta_1, \dots, \beta_{a-1}$ known as Breakpoints. The region from β_i to β_{i+1}=1/*a* under a N (0, 1) Gaussian curve, where β_a and β_0 refer to ∞ and −∞ respectively. By searching a table of statistical of them can find Breakpoints. For example, by searching in the table of breakpoints, it is possible to find *a* value ranging from 3 to 10 [38], as shown in Table 2.1.

Table 2.1: Breakpoint's values

α		3	4	5	6	7	8	9	10
β_i	β_1	0.43	-0.67	-0.84	-0.97	-1.07	-1.15	-1.22	-1.28
	β_2	0.43	0	-0.25	-0.43	-0.57	-0.67	-0.76	-0.84
	β_3		0.67	0.25	0	-0.18	-0.32	-0.43	-0.52
	β_4			0.84	0.43	0.18	0	-0.14	-0.25
	β_5				0.97	0.57	0.32	0.14	0
	β_6					1.07	0.67	0.43	0.25
	β_7						1.15	0.76	0.52
	β_8							1.22	0.84
	β_9								1.28

The APCA coefficients can be quantified as follows once the breakpoints have been defined. Every APCA value is converted to "a" if it is larger than or equal to the smallest breakpoint. The other values are less than the second smallest breakpoint are changed to "b," and so on.

2.7.3 The Differential Encoding

Differential encoding is also a method that can be used to time series data in order to describe the variations that occur in the data over the time period. Instead of storing the starting value of time series data, differential encoding includes encoding the difference between the most recent data point and the one before it [79].

Take for instance the case where we have a time series of temperature values collected over the course of any length of time. We can save time by recording the differences between successive temperature readings rather than encoding each individual temperature measurement. If the current measurement

is higher than the prior one by two degrees, we will interpret this as a positive value. If the temperature is lower than the previous measurement by one degree, we would represent this as a negative number in our encoding [80].

When working with time series data, differential encoding can be helpful because it can help to reduce the overall size of the data while still preserving the important information about the changes in the data over time. This helps differential encoding achieve its potential as a useful tool for working with time series data. Since even minute shifts in the data will only produce negligible variations in the encoded data, this technique may also be useful for lowering the impact that noise has on the data [81].

Nonetheless, employing differential encoding in time series data does come with certain possible downsides, just as using any other encoding approach does. For instance, if there are significant changes to the data over the course of time, the encoding may produce higher values, which may result in an increase in the total amount of the data. Also, as is the case with all compression strategies, there is a possibility that some information may be lost during the encoding process. This may have an effect on the precision of any analysis or modeling that is carried out on the data [82].

2.8 Data Compression

When discussing time series, the term "data compression" refers to the process of decreasing the number of observations needed to accurately describe a time series while maintaining a level of quality and accuracy that is comparable to the original data. This is relevant in a wide variety of applications because storage space may be at a premium or because it may be preferable to transfer data across a network with a lower bandwidth. Using lossless

compression methods, such as run-length encoding or Huffman coding, is one strategy that is used often in the process of data compression in time series. When applied to time series data that has repeated patterns or sequences of values that are near together, these approaches have the potential to be very successful in compressing such data [83].

Using methods for data reduction, such as down-sampling or smoothing, is an additional method that may be used to compress the data in time series. The amount of data points in a time series may be decreased by the use of down-sampling by selecting one point from the series just once every nth time, where n is an integer. Applying a filter to the time series as part of the smoothing process helps get rid of noise and lowers the amount of high-frequency components [84].

Wavelet compression and principal component analysis are two examples of more sophisticated approaches for data compression in time series (PCA). PCA involves identifying the principal components of the time series and representing it using a smaller set of variables, whereas wavelet compression involves the use of wavelet transforms to identify and remove redundant or irrelevant information from the time series. Wavelet transforms are used to compress the time series [85].

The particular application and the properties of the time series data both play a role in determining which data compression method should be used. While trying to get the highest possible compression performance, it is often necessary to apply a variety of various approaches in conjunction with one another. Data compression may be broken down into two primary categories, lossless compression and lossy compression [86]:

1. **Lossless compression** is a sort of data compression technology that reduces the amount of the data without sacrificing any of the information contained within it. The data that has been compressed may be decompressed back to its original form without any information being lost in the process. The Huffman coding method, the run-length encoding technique, and the Lempel-Ziv-Welch (LZW) compression method are all examples of lossless compression techniques.
2. **Lossy compression** is a sort of data compression technology that decreases the amount of the data by deleting part of the information that is deemed to be of lower importance or to be redundant. The data that has been decompressed is not identical to the data that was originally stored, but it is often sufficiently similar to be suitable for the majority of applications. JPEG and MP3 are two examples of lossy compression methods. JPEG is used for compressing pictures, while MP3 is used for compressing music.

The two different methods of compression each have their own set of benefits and drawbacks. When it is essential to save all of the original data, such as when compressing text or program files, lossless compression is the best option. This kind of compression may be found on modern computers. When the aim is to minimize the amount of the data as much as possible, such as when compressing audio or video files, lossy compression is advantageous since it achieves this goal to a greater extent. Lossy compression, on the other hand, might cause a reduction in the original quality of the data, therefore it might not be suitable for all applications [87].

2.8.1 Huffman Encoding

Huffman Encoding is a binary tree-based technique that leverages the frequency (or probability) property of symbols. It entails the following three steps [40]: probability calculation & symbol ordering, transformation of a binary tree, and assigning symbols with codes. In frequency calculation & symbol ordering, the number of each symbol in the entire data is counted, then the count is divided by the total number of letters in the data to get the "Frequency" of each symbol. One of the advantages of Huffman Encoding relies on the fact that it is a probability-based method. The most common symbols — the once with the highest frequencies — are often represented with fewer bits than least common symbols [40]. For example, we have the frequencies as has been shown if Figure 2.1 for the following data with five different symbols: A B C D E.:

Figure 2.4. Example of probabilities of five symbols A, B, C, D, E[40].

The binary tree transformation can be achieved as follows:

1. Select the two nodes with the shortest sum of frequencies from the collection. Then, merge the two nodes with a root includes the frequency equal to the sum of frequencies of these two selected nodes.

2. Reintroduce the new tree to the collection.

3. Repeat this process until the tree that encompasses all of the input frequencies.

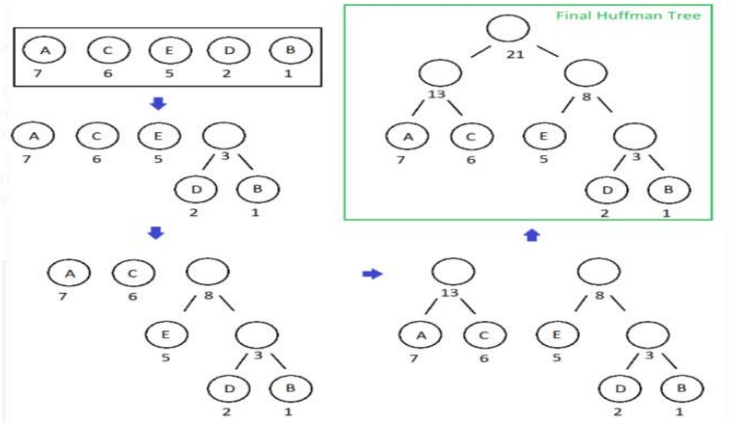


Figure 2.5. Binary Tree Transformation [40]

To assign the codes to symbols, the only thing that need to be done is to assign 1 for each time go right child and 0 for each time go left child. Then, the binary tree is obtained that known as Huffman Tree [40]. Finally, using Huffman Encoding, the symbols and their codes can be retrieved.

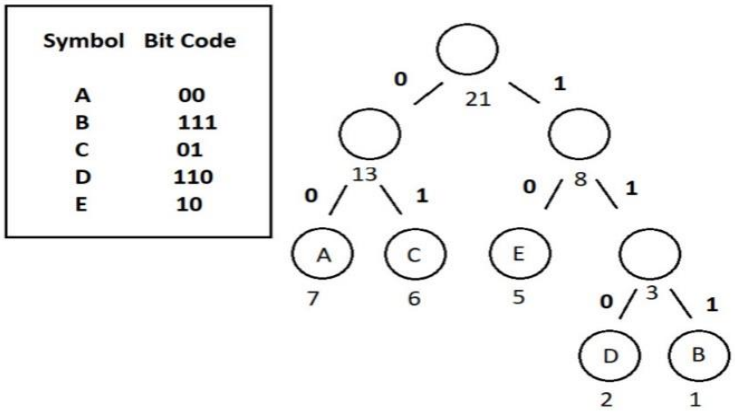


Figure 2.6. Assigning Codes to Symbols [40]

Even if the difference between compressed and non-compressed data is only 21 characters, which this difference can be seen as significant.

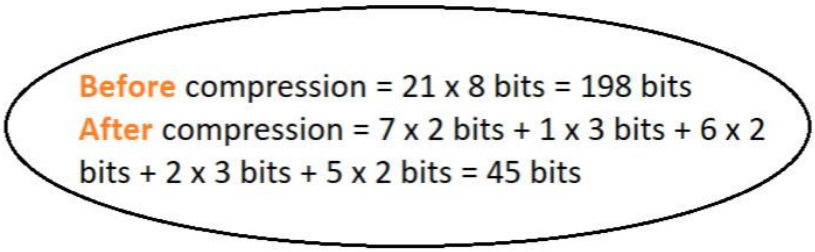


Figure 2.7. Number of bits calculation before and after compression [40]

2.8.2 LZW Compression

The most popular compression method is the LZW algorithm because it is a lossless technique; no data will be lost after compression. The technique is easy to use in hardware and has very high throughput potential. The idea relies on recurrent patterns to conserve data space. Due to its adaptability and simplicity, the method most commonly used for general-purpose data compression is LZW. Several PC utilities can "double the capacity of the hard disk" based on this method. Reading a set of symbols, stringing them together into codes, and finally altering the strings is how LZW compression operates. Since the codes take up less space than the words they replace, compression is possible. LZW has the following characteristics [86]:

1. A code table is used by LZW compression, with 4096 entries being a popular choice. Single bytes from the input file are always allocated the code values 0-255 in the code table.
2. Only the first 256 items of the code table are presented when encoding starts; the remaining entries are blanked. Codes 256 through 4095 are used to represent byte sequences during compression.
3. As the encoding process progresses, LZW identifies repeating sequences in the data and adds them to the code table.

4. Each code is retrieved from the compressed file and then used to accomplish decoding by passing it through the code table to see which character (or characters) it belongs to.

For example, consider the ASCII code. Normally, 8 binary bits are used to store each character, giving the data enough for up to 256 different symbols. By using 9 to 12 bits each character, this technique aims to expand the library. The new, distinctive symbols are combinations of previously occurring symbols in the string. Particularly with short, varied strings, it may not always compress properly [86].

However, this approach can both compress and uncompressed data, making it useful for compressing redundant data. Additionally, the new dictionary need not be saved with the data as well. The compression technique works by keeping a dictionary that corresponds the longest words encountered with a list of code values when the input data is processed. The input file is compressed because the words are swapped out for their matching codes. As a result, the algorithm becomes more effective as the volume of lengthy, repeating words grows in the input data.

2.9 Data Reduction Metrics

The challenge of building efficient algorithms in order to achieve data reduction is not a simple task, but rather a complex one. There are a number of significant performance elements that may be used in the calculation of algorithm efficiency [48], which are as follows:

2.9.1 Network Lifetime

"Network Lifetime" refers to the duration for which the network can remain operational or useful. A WSN is composed of a large number of small, low-power wireless sensor nodes that collect and transmit data to a central base station. The network lifetime in WSNs is an important factor that determines the efficiency and effectiveness of the network. The network lifetime in WSNs depends on several factors, including the number of sensor nodes, their energy consumption, the communication protocol used, and the power management strategies employed by the nodes. In general, the energy consumption of the sensor nodes is a critical factor that determines the network lifetime [88].

To maximize the network lifetime in WSNs, various power management strategies can be employed. For example, nodes can be put to sleep when they are not needed, or their transmission power can be reduced to conserve energy. Also, efficient routing protocols can be employed to minimize the energy consumption of the nodes during data transmission. Moreover, energy harvesting techniques, such as solar, thermal or vibration energy harvesting, can also be used to extend the network lifetime [89].

The network lifetime in WSNs is a crucial factor that needs to be carefully considered during the design and deployment of the network. The goal is to maximize the network lifetime while maintaining the required level of performance and reliability. A longer network lifetime can lead to cost savings and increased efficiency in various applications, including environmental monitoring, industrial automation, and healthcare [84].

2.9.2 Data accuracy

Data accuracy is a critical factor in sensor networks because the data collected by sensor nodes forms the basis for decision-making in various

applications. The accuracy of the sensor data is important because it directly impacts the quality of the decisions made based on that data. Several factors can affect the accuracy of sensor data in a sensor node. These factors include sensor calibration, environmental conditions, sensor drift, and sensor noise. Sensor calibration is critical to ensure accurate and reliable data collection. Environmental factors, including humidity, temperature, and light can also affect the accuracy of the data collected by the sensor node. Sensor drift refers to the gradual change in the sensor's output over time, which can result in inaccurate data collection. Sensor noise is another factor that can affect the accuracy of the data collected by the sensor node, and it can be caused by electronic or environmental factors [90].

To ensure the accuracy of the data collected by a sensor node, it is essential to calibrate the sensors regularly and to ensure that the sensors are operating within their specified environmental conditions. Also, it is important to employ noise filtering techniques to eliminate or reduce the effect of sensor noise on the data collected by the sensor node. The accuracy of the data collected by a sensor node can also be improved by using multiple sensors to measure the same variable and then combining the data from these sensors to obtain a more accurate measurement. This technique is known as sensor fusion and can significantly improve the accuracy of the data collected by a sensor node [48].

In summary, the accuracy of data collected by a sensor node is crucial for the performance of sensor networks. It is essential to take into consideration the factors that affect the accuracy of the data and to employ appropriate calibration, filtering, and fusion techniques to ensure accurate and reliable data collection [48].

2.9.3 Energy Consumption

Energy consumption is a critical factor in sensor networks because most sensor nodes operate on battery power, and their lifetime depends on how efficiently they use the available energy. Energy consumption in a sensor node is influenced by several factors, including the hardware design, software design, communication protocol, sensing mode, and environmental conditions. The hardware design of a sensor node has a significant impact on its energy consumption. Power-hungry components, such as the processor and radio, consume more energy than other components, such as the sensors. Therefore, the hardware design should be optimized to minimize power consumption. For example, low-power microcontrollers and radio modules can be used to reduce energy consumption [91].

The software design of a sensor node also plays a critical role in energy consumption. Efficient algorithms and data processing techniques can minimize energy consumption by reducing the time and frequency of processing and communication. The software design should also consider power management strategies, such as putting the sensor node to sleep when it is not required. Communication protocols used in a sensor network can also impact the energy consumption of a sensor node. The communication protocols should be designed to minimize the number of transmissions and the time spent in communication, as they consume significant energy. For example, routing protocols that minimize the number of hops required to transmit data can significantly reduce energy consumption [92].

The sensing mode of a sensor node can also impact its energy consumption. The sensing mode should be optimized to minimize the energy consumption while still providing the required accuracy and precision. For example, duty cycling can be used to minimize the time that the sensor is active

and reduce energy consumption. Environmental conditions such as temperature, humidity, and light can also affect the energy consumption of a sensor node. Extreme conditions can cause the sensors and other components to consume more energy, leading to a shorter lifetime. Therefore, it is essential to design the sensor node to operate within the specified environmental conditions [93].

In summary, energy consumption is a critical factor in sensor networks, and it is essential to optimize the hardware and software design, communication protocol, sensing mode, and environmental conditions to minimize energy consumption and extend the lifetime of the sensor node.

2.9.4 Data Transmissions

The number of data transmissions in a sensor node depends on the sensing mode, data processing, and communication protocol used in the network. Generally, the more data a sensor node sends, the higher its energy consumption and the shorter its battery life. The sensing mode used in a sensor node can affect the number of data transmissions. In continuous sensing mode, the sensor node generates data continuously, leading to a high number of data transmissions. In event-based sensing mode, the sensor node only generates data when a specific event occurs, leading to a lower number of data transmissions. Therefore, the event-based sensing mode is more energy-efficient than the continuous sensing mode [94].

Data processing also affects the number of data transmissions. Efficient algorithms and data processing techniques can reduce the amount of data generated and the frequency of data transmissions. For example, data aggregation can reduce the number of data transmissions by combining data from multiple sensor nodes before transmitting it to the base station. The

communication protocol used in the network can also affect the number of data transmissions [83].

The protocol should be designed to minimize the number of transmissions and the amount of data sent. For example, routing protocols that minimize the number of hops required to transmit data can significantly reduce the number of data transmissions. It is also essential to consider the required level of data accuracy and the application requirements when determining the number of data transmissions. The sensor node should only send data that is necessary for the application, and the data should be transmitted at a frequency that meets the application requirements [83].

In summary, the number of data transmissions in a sensor node is influenced by the sensing mode, data processing, and communication protocol used in the network. It is essential to optimize these factors to minimize the number of data transmissions and conserve energy in the sensor node.

2.10 Similarity Measures

While performing activities related to data mining and the study of time series, using similarity measures is very crucial and vital [95]. A connection between two items and a scalar number is one way to construct a measure of similarity. This relationship may also be thought of as a scale. For the purpose of mapping the similarities, the intervals [-1, 1] or [0, 1] are used, with 1 being the highest level of proximity [96].

There are four primary categories that may be used to categorize the many similarity metrics of time series that can be found in the research literature. The basic form of the series is used to compare the distances between points. Edit-based metrics determine the total number of actions required to transform one time series into another. These actions might include adding, removing, or

editing data. With feature-based distances, the characteristics of the time series are extracted first, and then any distance function may be used to compare the extracted attributes. In conclusion, structure-based similarity metrics compare different series by first extracting higher level structures from them and then comparing those structures to one another [96].

This dissertation may use various standard measurement techniques in order to determine whether or not there is a full or partial match between two time series. These methods include the Euclidean distance and the cosine angle that exists between the two vectors.

2.10.1 Euclidean Distance Similarity

Euclidean distance is a measure of the distance between two points in Euclidean space. It is the length of the straight line segment connecting the two points. In two-dimensional space, the Euclidean distance between two points (x_1, y_1) and (x_2, y_2) is given by the formula [96]:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \qquad \textbf{(2.10)}$$

In three-dimensional space, the Euclidean distance between two points (x_1, y_1, z_1) and (x_2, y_2, z_2) is given by the formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \qquad \textbf{(2.11)}$$

The Euclidean distance can also be extended to higher-dimensional spaces in a similar way. The Euclidean distance is a fundamental concept in many

fields, including mathematics, physics, and computer science. It is commonly used in machine learning and data analysis to measure the similarity between data points in a high-dimensional space [96].

2.11 Energy Consumption Model

Typically, a network of sensors will be made up of sensor nodes, each of which is powered by a minuscule battery that only provides a limited amount of Jule. Therefore, one of the things to follow in wireless sensor networks is in order to minimize the energy consumption of individual sensor nodes, which is considered to be one of the important issues that must be resolved over the lifetime of the network. This is one of the things that is considered to be one of the things to follow in wireless sensor networks. As a result of the fact that radio communication is the most significant contributor to the total amount of energy that is used by WSNs, it is imperative that the sensor nodes use as little power as possible while transmitting data [97].

Two distinct methods exist for reducing the amount of electricity transferred via protocols. The first advantage is that sensor nodes are able to convey data to relatively close destinations, such as base stations (also known as sinks) or cluster heads. After then, the cluster head will utilize its capacity to transport the gathered data across a great distance in order to keep the energy levels of the lesser nodes stable. The second approach is to lessen the amount of data, or bits, that are transferred via wireless networks. Our procedures are predicated on the goal of cutting expenses (in terms of energy) by identifying and consolidating superfluous data in the interest of preserving the authenticity and precision of the information [97].

We employed the very same energy consumption model that was described in [98] in order to determine the amount of energy that our procedures required.

This model suggests that the power dissipated by the radio will be E_{elec}=50 *nJ/bit* while operating the circuits of the transmitter or receiver, and β_{amp}=100 $pJ/bit/m^2$ when operating the amplifier of the transmitter. The radios have the ability to adjust the power and may increase the minimum power necessary to reach the intended receivers. In addition, the radios have the ability to be switched off to prevent the receipt of inadvertent broadcasts.

The *m*- bits message is for the cost of transmission, and d is for the distance, Equation 2.12 show this.

$$E_{TX}(m, d) = E_{elec} * m + \beta_{amp} * m * d^2 \quad \textbf{(2.12)}$$

The energy consumption required for reception m − bits is calculated as in Equation 2.13.

$$E_{RX}(m, d) = E_{elec} * m \quad \textbf{(2.13)}$$

The amount of energy that must be supplied to the sensor node in order to record *m*- bits is given by Equation 2.14.

$$E_{CX}(m,) = E_{TX}(m, d) / 7 \quad \textbf{(2.14)}$$

Since receiving data is likewise a costly activity (in terms of energy), the total amount of data that is received and sent should be kept to a minimum. In the simulations of our experiment, the length of the data reading m is assumed to be equal to 64 bits. In the first level, which corresponds to the level of the sensor nodes, and Each sensor node will have accumulated data readings at the end of each period. When a message containing m bits is being sent, an

additional 64 bits are appended to it. This adds up to the frequency of data reading m. The following formula (2.15) is used to determine how long each data packet will be when it is sent:

$$DataPacket\ Length = (number\ of\ readings\ in\ the\ data\ set \times 2) \times 64 bits \quad \textbf{(2.15)}$$

As a result, the length of a packet is equal to the number of readings in the sensed dataset multiplied by 64 bits to account for their frequencies. Hence, the term "energy consumption" refers to the aggregate amount of energy dissipated at individual sensor nodes while gathering and transmitting data readings. This quantity is mathematically expressed in equation 2.16.

$$E_{Sensor}(m,) = E_{TX}(m,d) + E_{CX}(m,d) \quad \textbf{(2.16)}$$

At the Gateway level, also known as the CH level, the cluster head is the node that is responsible for receiving the sensed data sets from all of the member sensor nodes that are part of the same cluster throughout each period. Therefore, the amount of power that is consumed at the Gateway level is equal to the amount of energy that is consumed by the data sets that are received at the cluster head from its members as well as the amount of power that is consumed by the data that is transmitted to the sink and is computed using equation 2.17.

$$E_{\mathrm{rd}}(m,) = E_{TX}(m,d) + E_{RX}(m,d) \quad \textbf{(2.17)}$$

2.12 Conclusion

An introduction to Periodic Sensor Networks (PSNs) has been provided in this chapter for those interested in learning more. PSNs, as opposed to event- or query-based Sensor Networks, send their data to be processed at the sink on a periodic basis. It has been shown that it may be used in a variety of application domains, including environmental monitoring, industrial monitoring, undersea monitoring, and medical monitoring. The primary challenges associated with PSNs have been outlined. The issue of time series in PSNs as well as the energy-efficient data reduction techniques that seek to minimize the amount of energy spent by wireless sensor nodes have been demonstrated. The goal of these mechanisms is to reduce the amount of energy that is consumed by the wireless sensor nodes. Many different performance criteria are broken down here so that an evaluation of the effectiveness of the suggested data reduction strategies may be carried out. This chapter focuses on the aspects of machine learning that are related to time series. These aspects include prediction techniques, data representation techniques, similarity measures, and data compression techniques. These aspects are all taken into consideration when designing an energy-efficient data reduction protocol for PSNs. In addition to that, an explanation has been provided for the energy consumption model.

CHAPTER THREE

PROPOSED DATA REDUCTION APPROACHES

CHAPTER THREE: PROPOSED DATA REDUCTION

3.1 Introduction

One of the most difficult problems that has to be solved in PSN is determining the maximum lifespan of the battery. Whereas, in sensor nodes, radio communication, which includes the transmission and receiving of messages, is the aspect that has the most impact on the amount of energy that PSNs use.

In this dissertation, we presented novel approaches for the decentralized implementation of energy-efficient data reduction. The prediction and data compression methodologies were used throughout the development of the suggested approaches for data reduction. Figure 3.1 the work of the dissertation with the techniques used.

The goal of these approaches is to minimize the amount of data that is sensed on two different levels: sensor nodes (SN) level and Fog Gateway (FG) level. This is done to extend the network's lifespan in PSNs. The work that has to be done at the sensor node stage involves deleting duplicated measurements at each period so that the amount of sensed data that is sent to the Gateway may be reduced. While on the other hand, the reduction that occurs during the second stage (Gateway) functions as a filtering mechanism by enabling the Gateway to recognize and then eliminate sets of data that are redundant and produced by neighboring nodes. Before being sent to the Cloud must reduce the overall number of sets. The following sections provide a full overview of possible solutions to the problem of managing the amount of energy that is being used.

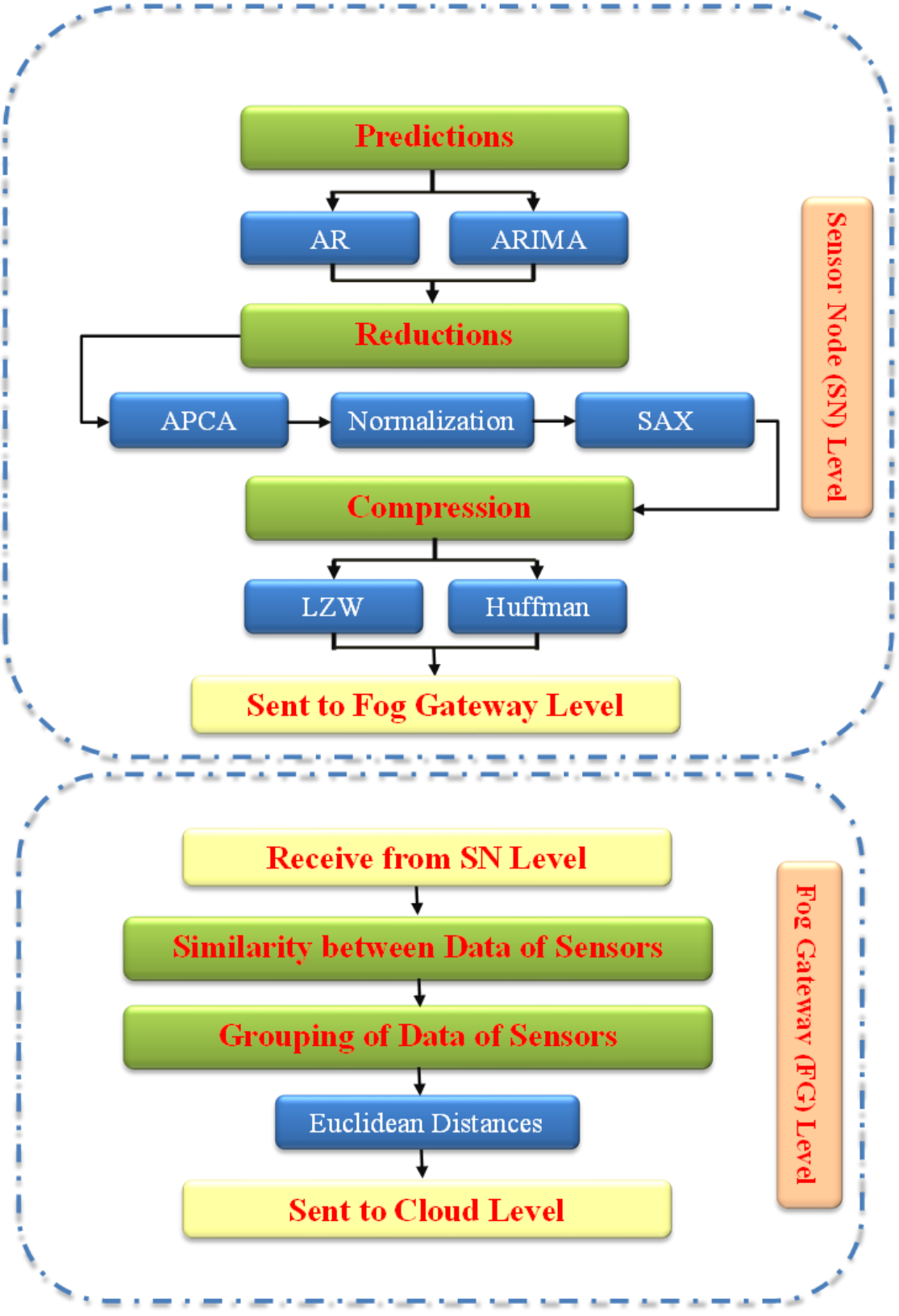


Figure 3.1. The Design of Our Work

3.2 Sensor Nodes Level

Minimizing energy consumption in each sensor node is an essential challenge in WSN that has to be addressed during the lifetime of the network's existence. At the sensor node level, we developed four different approaches. At this stage, the recommended algorithms are implemented on every sensor node. In the following sections, we'll go into further depth about the proposed approaches.

3.2.1 Proposed DEDAR Approach

A distributed energy-efficient data reduction (DEDaR) approach based on prediction and compression is proposed to decrease data transmission in IoT networks. The DEDaR Approach is implemented on each sensor device. It is regarded as an effective method to save power and reduce the amount of transferred data and thus extending the lifetime of the network while maintaining the accuracy of received data at the gateway. Figure 3.2 depicts the flowchart of the proposed DEDaR approach.

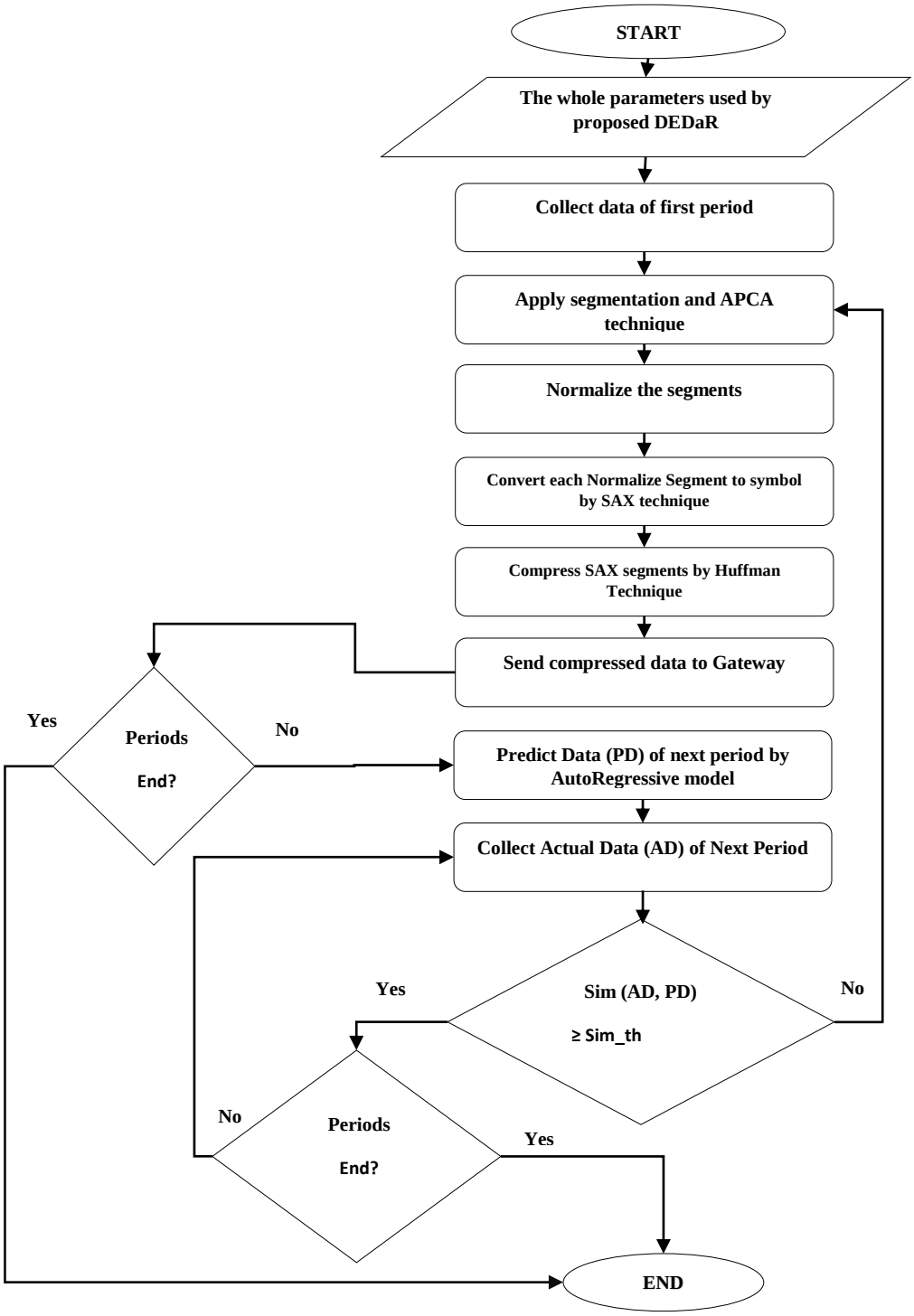


Figure 3.2. Flowchart of proposed DEDaR approach

3.2.1.1 Collection of IoT sensor data

The data collection is periodic, where the new reading is collected from sensor node i at slot times. The node i then creates a new data vector (i.e. time series vector) from the captured readings $R = [r1, r2,..., yT]$ at each period, where T is the total amount of readings in each period. The IoT sensor node will capture the (same or very similar) measurements, especially when is very short or when there is no change in the monitored area for a long time. This will allow the IoT sensor node to send a large volume of data to the Gateway at each period.

3.2.1.2 APCA technique

In many cases, the sequence of temperature measurements is too extensive to be evaluated, and as a result, it is required to approximate the data. The purpose of data approximation is to reduce the number of sensor readings while preserving the essential structure and properties of the signal. DEDaR approach employs a simple and efficient representation approach called APCA. At this step, the DEDaR approach changes the series of temperature readings from its initial form to the APCA representation (as mentioned in section 2.7.1), to reduce the dimensionality of the series, which refers to the number of observed measurements. This series is broken up into many equal parts by APCA. After that, it determines the mean value for each individual part. Algorithm 3.1 outlines the procedure that must be followed to convert the initial temperature measurement series into an APCA representation.

In the beginning, the first period will be sent to the Gateway. The data of this period and the other periods that need to be sent to the Gateway will be processed to remove the redundancy at the sensor node's level before sending

them to the Gateway. Therefore, the dimensionality reduction based on segmentation and APCA is to segment the data into m segments of data readings with different lengths depending on their activities, where the long segments represent the low activities and the short segments represent the high activities.

Algorithm 3.1: Dimensionality Reduction based Segmentation and APCA

Input *: SoD (set of data reading of a period)*

Output *: $SoG = \{SoG_1, ..., SoG_m\}$ (set of m segments of readings), $SegLen$ (set of lengths of segments)*

Process:

1. $l \leftarrow 0$
2. $\overrightarrow{set\_bfr} \leftarrow \emptyset$
3. $m \leftarrow 0$
4. $SoD \leftarrow Sorting\ (SoD)$ // Sorting of readings
5. **for** $j \leftarrow 1$ **to** len(SoD) **do**
6. $set\_bfr_l \leftarrow SoD_j$
7. $l \leftarrow l + 1$
8. $Sm \leftarrow 0$
9. $Count \leftarrow 0$
10. $flag \leftarrow 0$
11. **for** $i \leftarrow 1$ **to** $len(\overrightarrow{set\_bfr})$ **do**
12. $Sm \leftarrow Sm + set\_bfr_i$
13. $Count \leftarrow Count + 1$
14. **end for**
15. $SEG_\mu \leftarrow Sm / Count$
16. $i \leftarrow 0$
17. **while** *(flag = 0)* **and** $(i < len(\overrightarrow{set\_bfr})$ **do**
18. **if** $(set\_bfr_i - SEG_\mu) \geq e\_max$ **then**
19. $flag \leftarrow 1$
20. $i \leftarrow i-1$
21. **else**
22. $i \leftarrow i+1$
23. **end if**
24. **end while**
25. **if** *(flag = 1)* **then**
26. $l \leftarrow 0$
27. $m \leftarrow m+1$
28. $Count \leftarrow 0$
29. $Sm \leftarrow 0$
30. **for** $k \leftarrow 1$ **to** i **do**
31. $Sm \leftarrow Sm + set\_bfr_j$
32. $Count \leftarrow Count + 1$
33. **end for**
34. $SoG_m \leftarrow Sm / Count$
35. $SegLen_m \leftarrow len(\overrightarrow{set\_bfr})$
36. $\overrightarrow{set\_bfr} \leftarrow \emptyset$
37. **end if**
38. **end for**
39. **return** $SoG,\ SegLen$

3.2.1.3 SAX Representation

This approach converts from a numerical form of time series to a sequential series of discrete symbols using pre-defined mapping rules. Symbolic techniques are more noise resistant. SAX is the most extensively used symbolic representation approach for time series data mining. A large reduction in dimensions ensures this representation method as well as a reduced bounding property, improving the efficiency of the algorithm. It is necessary to transform time series, which contain data readings from sensor nodes, into suitable forms for further analysis. To deal with time series, it is recommended to use two different representation techniques: normalization and symbolic representation.

After applying the APCA method, which divides the data into a set of segments, the normalization method was used on the data of each segment as shown in Algorithm 3.2. Then, the SAX method was used to convert the data into symbols by using a breakpoint value whose alphabet range is 3–10 symbols as shown in Table 2.1. Note that the number of characters used determines the breakpoint values. In our method, the number of symbols that are used is 10. Every APCA value will be converted to a suitable representation using the SAX algorithm. Algorithm 3.3 demonstrates the SAX algorithm.

Algorithm 3.2: Normalization Data of Segments

Input: *SoG: APCA Segments, SegLen: Length of APCA Segments*
Output: *SoN: Set of Normalize Segments.*
Process:
1. $sum_{\mu} \leftarrow 0$
2. ***for*** $j \leftarrow 1$ *to SegLen* ***do***
3. $sum_{\mu} \leftarrow sum_{\mu} + SoG_j$
4. ***end for***
5. $\mu \leftarrow \frac{sum_{\mu}}{SegLen}$ *// find mean values*
6. $Sum_{\sigma} \leftarrow 0$
7. ***for*** $j \leftarrow 1$ *to SegLen* ***do***
8. $Sum_{\sigma} \leftarrow Sum_{\sigma} + |SoG_j - \mu|$
9. ***end for***
10. $\sigma \leftarrow \sqrt{\frac{Sum_{\sigma}}{SegLen}}$ *// find the standard deviation*
11. ***for*** $j \leftarrow 1$ *to SegLen* ***do***
12. $SoN_j \leftarrow \frac{SoG_j - \mu}{\sigma}$
13. ***end for***
14. ***Return*** *SoN*

Algorithm 3.3: SAX Representation

Input: *SoN : Set of Normalize Segments, SegLen: length of Segments, B (breakpoint values of table 2.1)*
Output: SAX_{sm}*: Symbol Segments.*
Process:
1. ***alphabet*** ← *{'a' ,'b' ,'c' ,'d' ,'e' ,'f ,'g' ,'h' ,'i' ,'j' }*
2. *char* ← *10 # The no. of Symbol that represents the values*
3. ***for*** *i* ← *0 to SegLen* ***do***
4. ***for*** *j* ← *0 to len(B)* ***do***
5. ***if*** $B_{[j+1][char-1]} > SoN_i \geq B_{[j][char-1]}$ ***then***
6. $SAX_{sm\ i} \leftarrow alphabet_j$
7. ***end if***
8. ***end for***
9. ***end for***
10. ***Return*** SAX_{sm}

A symbolic representation of data readings for IoT sensor nodes from a normalized one by using the SAX approach. The SAX can be generated the

quantification does this conversion by dividing the area under the Gaussian distribution into an equal proportional area using (N - 1) breakpoints.

3.2.1.4 Huffman Encoding

The proposed DEDaR approach employs a static alphabet dictionary derived from the implementation of the Huffman algorithm on the data of sensor 1 in the dataset and we named this fixed alphabet as Fixed Code Dictionary (FCD) based on Huffman encoding. The FCD is constructed as a fixed dictionary to avoid frequent sending for dictionary of variable codes with transmitted encoded data to the Gateway to decode received data. The fixed alphabet must achieve two conditions: the absence of ambiguity in the code, and the code must be as minimal as possible.

The FCD will be static for all periods with a variable bit of code for each symbol. The Gateway will receive this FCD dictionary only once. Table 3.1 shows the Fixed Code Dictionary (FCD). In this table, the first column represents the symbols generated by the SAX representation (Algorithm 3.3). The second column refers to the code of the symbols. The third column represents the number of required bits of each code to represent the symbols.

Table 3.1: Fixed Code Dictionary (FCD)

Symbol	Code	No. of Bit
a	00	2
b	0100	4
c	0101	4
d	0110	4
e	0111	4
f	1000	4
g	1001	4
h	10100	5
i	10101	5
j	10110	5

3.2.1.5 AutoRegressive Prediction Model

Autoregression predicts the value at future time steps by using data from prior time steps as input to a regression equation. It is a simple method that can produce reliable forecasts for a variety of time series issues. The DEDaR approach employs a model for time series forecasting based on an autoregressive prediction method. A linear regression model forecasts the result using a linear combination of input values and can be formulated as:

$$Y^{\wedge} = b_0 + b_1 * X_1 \qquad \textbf{(3.1)}$$

Where y_{hat} is the forecast, b_0 and b_1 are parameters discovered by model optimization on training data, and X is an input value. This approach can be applied to time series when the input parameters, referred to as lag parameters, are observations from prior time steps. For instance, the value of the next time step (t+1) can be estimated based on the data from the previous two-time steps t-1 and t-2. This might be represented as a predictive model as follows:

$$X_{t+1} = b_0 + (b_1 * X_{t-1}) + (b_2 * X_{t-2}) \qquad \textbf{(3.2)}$$

The regression model is known as an autoregression because it uses the same input variable as data at earlier time steps. An AutoRegression model is a linear regression model using lagged data as input variables. Manually designing the linear regression model and the lag input parameters could use Linear Regression. Alternatively, the statistical model package offers an autoregression model that requires a lag value and trains a linear regression model. It is part of the AutoReg class. This model can be used by first generating the AutoReg() model and then training it on our dataset with fit(). An

AutoRegResults object is returned. Once the model is fit, it can use it to generate predictions by using the predict() method for a set of future data.

The DEDaR approach collects the data inside the sensor node periodically. The collected data of the first period are used by the autoregressive prediction technique to predict the data of the next period. The sensed data of the first period will always be transmitted to the Gateway after compressing those using previous sections.

Algorithm 3.4: AutoRegressive Prediction

Input: *DF: set of data file reading from the sensor*
Output: *PD: set of prediction data*
Process:

```
1.  count ← 0
2.  for i ← 0   to   len(DF)  do
3.      train,  test ← splitTrainTest (DF, 0.3)
4.      selector = ar_select_order( train )
5.      model ←  AutoReg(train, lags=selector )
6.      model_fit  ←    model.fit()
7.      PD[count] ← model_fit.predict(start=len(train), end=len(train)+len(test)-1))
8.      count ← count+1
9.      DF ← shifting( DF)
10. end for
11. return PD
```

The AutoReg() method in Algorithm 3.4 gets data from the sensor as well as the number of lags to include in the model (if an integer) or a list of lag indices to be included. For example, (1, 4) includes only lags 1 and 4, whereas lags = 4 includes lags 1, 2, 3, and 4. None of the AR lags respond in the same way as 0. Then, using the fit() function, fit the model. Model fitting is a statistic for determining how well a machine learning model generalizes to data that is comparable to the data it was trained on. A well-fitted model produces more accurate results. After training the model, the following step is usually to make predictions on the testing set. To do so, we must use the predict() function, which will effectively use the learned parameters from fit () to make predictions

on fresh, unknown test data points. Predict(), in essence, will make a forecast for each test instance, and it normally only requires a single parameter (X). The predicted value for classifiers and regressors will be in the same space as the one observed in the training set.

3.2.1.6 Similarity Measures

The similarity algorithm using Euclidean distance measurement is demonstrated in Algorithm 3.5 which will be applied between the predicted data and newly collected data of the next period. If the results are greater than or equal to a predefined threshold ***Sim_th***, they are considered as similar and they will not be transmitted to the Gateway. Otherwise, if the result of the similarity function is less than ***Sim_th***, the sensed data are considered dissimilar and should be transmitted to the Gateway. The techniques in (**Section 3.2.1.2, Section 3.2.1.3, and Section 3.2.1.4**) are applied to the data before sending them to the Gateway. Every time the similarity between the actual collected data and the predicted data is less than ***Sim_th***, the prediction model will be trained with actual collected data to produce the new predicted data set that will be used for the next periods.

Algorithm 3.5: Similarity Algorithm

Input: *PD: Prediction Data set.*
AD: Actual Data set
Output: *Sim: similarity value*
Process
1. $sum \leftarrow 0$
2. ***for*** $i \leftarrow 0$ ***to*** *len(PD)* ***do***
3. $sum \leftarrow sum + |(PD_i - AD_i)|$
4. ***end for***
5. $D \leftarrow \sqrt{sum}$
6. $Sim \leftarrow \frac{1}{1 + D}$
return Sim

3.2.1.7 Computational Complexity of DEDaR

This section provides the analytical study of the computational complexity for the proposed DEDAR approach. Each sensor node i collects a series of data during each period, which is represented by R, where the total amount of readings during one period is T. The time complexity of the Algorithm 1, which is responsible for dimensionality reduction based on segmentation and APCA is $O(MAX(Len(SoD)Log_2Len(SoD), Len(SoD) * Len(\overrightarrow{set\_bfr})))$. The space requirement for Algorithm 1 is $\Theta(Len(SoG)*m + Len(SegLen) + Len(SoD) + Len(\overrightarrow{set\_bfr}))$. The time requirement for Algorithm 2 is $\Theta(SegLen^x)$, while it requires storage equal to $\Theta(Len(SoG^x)+Len(SoN^x))$. Algorithm 3 requires time and storage equal to $\Theta(SegLen^x * Len(B))$ and $\Theta(Len(SoN^x)+ Len(\beta)+SegLen^x)$ respectively. The time and space complexities are required by Algorithm 4 are $\Theta(Len(DF)* lags)$ and $\Theta(Len(DF)+Len(PD))$ respectively. Algorithm 5 requires $\Theta(Len(PD))$ of time complexity, whilst it requires storage equal to $\Theta(Len(PD)+Len(AD))$. The Huffman Encoding uses a fixed code dictionary (FCD) that includes the used symbols and their codes. The FCD will be saved at both sender (sensor node) and receiver (gateway). It is simple and requires a very small size of memory (121 bits). The encoding requires $\Theta(Len(SAX^x_{sm}))$ of time complexity.

Hence, the time complexity for DEDaR approach is $O(MAX(Len(SoD)Log_2 Len(SoD), Len(SoD) * Len(\overrightarrow{set\_bfr})))$ and the storage requirement is $\Theta(Len(SoG)*m + Len(SegLen) + Len(SoD) + Len(\overrightarrow{set\_bfr}))$. This computational complexity analysis shows the implementation cost of the proposed DEDaR approach from the time and storage requirements point of view.

3.2.2 Proposed DiPCoM Approach

We propose a distributed prediction-compression-based mechanism (DiPCoM) for saving power in IoT networks. The new contribution of this approach is using the ARIMA prediction method to predict the data for the next period and suggest LZW as an effective compression technique to reduce the amount of data sent to the Gateway and keep the energy of the sensor nodes. This makes the network's lifespan longer. The DiPCoM applied the same reduction techniques used in DEDaR with added Differential Encoding (DE) techniques after using APCA that it finds the difference between every two adjacent values in the time series, then puts the first value of the time series with the vector of differences to be used. This uses the fact that encoding the differences would require fewer bits than the original values. DiPCoM experiments are also achieved using a Python-based custom simulator.

3.2.2.1 Lempel-Ziv-Welch Compression

The LZW method is the most well-known lossless compression algorithm and employs a dynamic dictionary to encode modernistic words using sequences previously acquired. In the input file, one word typically receives the dictionary codes 0-255. LZW compression reads a stream of symbols, arranges these symbols into words, and converts these words into codes; thus, compression is made possible by the codes' smaller size compared to the words they replace. When encoding begins, the dictionary only has the first 256 entries. The rest of the entries are blank. The SAX algorithm uses an alphabet from A to J (for example, a=10 symbols), therefore the range of the initial dictionary will be from "A" to "J" depending on the SAX alphabet. Therefore,

we decided to use the sequence of symbols (1–10) in the dictionary. This will make it smaller and better suited for IoT sensor nodes with limited memory.

The LZW algorithm works by taking a token from a string of tokens obtained from SAX and checking each token. If it exists in the dictionary, it takes the next symbol and adds it to the previous set of symbols for the purpose of checking the new symbols again. If the new symbol is not present in the dictionary, it adds it to the dictionary and then gives it a special code and adds the index of the previous symbols to the token. It continues with this mechanism until all the symbols in the series are completed and a new symbol that is not in the dictionary appears. The result will be a token compression, which represents the code for the string of symbols entering the algorithm. The main objective behind using lossless LZW after lossy reduction methods is to compress the data without loss while keeping its accuracy at a suitable level at the gateway. Algorithm 3.6 demonstrates the LZW algorithm.

Algorithm 3.6: LZW Compression

Input: *SAX_{sm}: Symbol Segments, SegLen: length of Segments.*
Output: *TCom : Token Compression.*
Process:

1. *alphabet ← {'a' , 'b' , 'c' , 'd' , 'e' , 'f' , 'g' , 'h' , 'i' , 'j' }*
2. *Dic ← Initialize Dictionary (alphabet)*
3. ***for*** *i ← 0* ***to*** *SegLen* ***do***
4. ***if*** *SAXsm [i] ← in Dic* ***then***
5. *string ← string + SAXsm[i]*
6. ***else***
7. *TCom ← index of string in Dic*
8. *Dic ← string + SAXsm[i]*
9. *string ← SAXsm[i]*
10. ***end if***
11. ***end for***
12. ***Return*** *TCom*

We can explain the working of LZW by presenting an illustrative example for applying LZW of SAX representation, consider we have input stream getting from SAX algorithm which is {abcedajabdca}. Table 3.2 shows the LZW Dictionary.

Table 3.2: LZW Dictionary

Index	Entry	Token
1	a	
2	b	
3	c	
4	d	
5	e	
6	j	
7	ab	1
8	bc	2
9	ce	3
10	ed	5
11	da	4
12	aj	1
13	ja	6
14	abd	7
15	dc	4
16	ca	3
17	a	-

3.2.2.2 ARIMA Prediction Algorithm

A statistical analysis model called ARIMA examines time series data to clarify the data set or based trends. An autoregressive statistical model is one that forecasts future values using data from the past. For instance, an ARIMA model can attempt to predict sales income consistent with earlier periods or forecast the price of a stock based on past performance. The parameters p, d, and q are necessary for the model of ARIMA and can be determined. The order of the autoregressive (AR) term is p, and the number of differences needed to

keep the stationarity of the series is d. The order of the moving average (MA) term is q. Based on the Augmented Dickey-Full-Test, the value d is calculated and equal to 1. The partial autocorrelation function (PACF) is used to give p, and the parameter q is computed by the autocorrelation function (ACF), according to the Box-Jenkins technique [8].

The first lag, which is within the significance threshold, provides the values for the AR and MA terms. AR = 2 and MA = 0 from the ACF and PACF create the limits for evaluating various combinations of (p, and q). Before fitting the model, the data is separated into training and testing datasets. Then, using the training data for building the model and achieving the predictions for each value in the test dataset, a rolling forecast was constructed. The rolling forecasting method requires that the model be recreated once a new import data value is obtained. This is achieved by removing the first value in the data that indicates the oldest value, then doing shifting for all values in the data and adding the new prediction value at the end of the data, then looping that for all the data. Algorithm 3.7 shows the ARIMA prediction algorithm.

Algorithm 3.7: ARIMA Prediction

Input: *DF: set of data file reading from sensor.*
Output: *PD: set of prediction data.*
Process:

```
1.  count ← 0
2.  for i ← 0  to  len(DF)  do
3.      train,  test ← splitTrainTest(DF, 0.3)
4.      selector(p,d,q) = order_fu(train, p,d,q)
5.      model ←  ARIMA(train, order=selector(p,d,q) )
6.      model_fit ←   model.fit()
7.      PD[count] ← model_fit.predict(start=len(train), end=len(train)+len(test)-1))
8.      count ← count+1
9.      DF ← shifting( DF)
10. end for
11. return PD
```

The ARIMA prediction approach uses the first period gathered data to anticipate the data for the next period. After apply techniques in (**Section 3.2.1.2, Section 3.2.1.3, and Section 3.2.1.4**) that have been used to compress the first period's data, it will always be sent to the gateway. The predicted data will be compared to new data from the future using the similarity algorithm (see Algorithm 3.5). The findings are considered similar and will not be sent to the gateway if they meet or exceed a certain threshold (***Sim_th)***. If the similarity function yields a result lower than ***Sim_th***, the sensed data are deemed distinct and must be sent to the gateway.

3.2.2.3 Computational Complexity of DiPCoM

The computational complexity of the proposed DiPCoM technique is analytically studied in this section. Algorithm 1, which uses segmentation and APCA to reduce dimensionality, has a complexity of $\theta(MAX(Len(SoD)\ log_2 Len(SoD)\ ,\ Len(SoD) * Len((\overrightarrow{set\_bfr}))))$. Algorithm 1 needs space equal to $\theta(Len(SoG) * m + Len(SegLen) + Len(SoD) + Len(Seg_DE) + Len((\overrightarrow{set\_bfr}))$. Algorithm 2 has a time need of $\theta(SegLen)$ and a storage requirement of $\theta(SegLen) + Len(SoN) + Len(Seg_DE))$. The time and storage needs for Algorithm 3 are $\Theta(SegLen) * Len(\beta))$ and $\theta(Len(SoN) + Len(\beta)+SegLen)$, respectively. Algorithm 4 requires $\theta(SegLen)$ of time and storage. The computational need of Algorithm 5 is $\theta(n^2 * len(DF))$, where n is the number of parameters (n= p+q+P+Q) and len(DF) is the length of the time series. Algorithm 6 requires $\theta(Len(PD))$.

3.2.3 Proposed IDaPCoT Approach

Integrated Data Prediction and Compression Techniques (IDaPCoT) is a technique that combines data prediction and compression to optimize data transmission. It involves using predictive models to generate predictions of future data, which are then used to compress the data in the first period and compress the data that is different from the prediction data.

The idea behind IDaPCoT is that by predicting the future data, the amount of information that needs to be transmitted can be reduced, resulting in more efficient transmission. IDaPCoT involves several steps, including:

1. **Prediction:** this involves using machine learning models to predict the future data based on historical data. We are proposed AR prediction technique in this approach (**see section 3.2.1.5**)
2. **Dimensionality reduction:** this introduce the APCA method that is used for time series data. Which is described in (**section 3.2.1.2**). The other technique that using in data representation is SAX representation, which is described in (**section 3.2.1.3**). The third is Symbolization: The normalized segments are then mapped to a sequence of discrete symbols from a predefined alphabet. This mapping is based on a set of breakpoints (**see Figure 2.1**) that divide the range of possible values into equal-sized intervals.
3. **Data compression**: This step involves compressing using lossless compression techniques such as Lempel-Ziv-Welch (LZW) compression. Which is described in (**section 3.2.2.1**).

3.2.4 Proposed EDaRePE Approach

We propose an Energy-efficient Data Reduction based Prediction and Encoding (EDaRePE) in IoT networks. The EDaRePE approach is a proposed approach aimed at reducing energy consumption in wireless sensor networks (WSNs). This approach utilizes data reduction techniques and prediction algorithm to reduce the amount of data transmitted in the network, thus reducing energy consumption. Specifically, EDaRePE uses a combination of existing prediction algorithms and proposed Huffman encoding techniques to compress data within the WSN.

The prediction algorithm used is ARIMA prediction which is used to predict the values of data that are yet to be transmitted **(see section 3.2.2.2)**, and the Huffman encoding technique is used to compress the data that is been different from the data prediction and the data of the first period **(see section 3.2.1.4)**. This reduces the amount of data that needs to be transmitted, which in turn reduces the amount of energy consumed by the nodes in the network. Also this approach used the APCA with Differential Encoding (DE) **(see section 3.2.1.2)** and used the SAX representation approaches **(see section 3.2.1.3)**. Overall, the EDaRePE approach offers a promising solution to the problem of energy consumption in WSNs. By reducing the amount of data transmitted, EDaRePE can significantly improve the longevity of the network, as well as reduce the need for frequent battery replacements.

3.3 Gateway Level

Spatial correlated data reduction technique for energy conservation in the IoT network. We design approach named **Two-Tier Energy-efficient Data Reduction Technique for IoT Networks (TEDaReT)** to reduce data before sending to cloud. The TEDaReT is executed to remove the duplicated sets of data received from the sensor node level to reduce the data sets based on find the similarity between data sets. The main goal is to enable the Fog gateway to decrease the consumed energy and prolong the network's lifetime whilst maintaining the data's integrity. In this level (Gateway level), k data sets of readings and their repetition "$S^{AP} = (S_1^{AP}, S_2^{AP}, \ldots, S_k^{AP})$" have arrived at the Gateway at the end of each period.

3.3.1 TEDaReT Approach

The basic idea of this method is to group sensors whose data are similar into groups based on a threshold. In each group, a representative of the group is taken by finding the similarity between the data of one group, and the representative is the sensor has the highest percentage of similarity. As a result, a representative of each group is sent with the index of sensors that were removed. One of the simplest ways to grouping the sensors by discover similarity between data of sensors is by using Euclidean Distance measured. The purpose of this is to find the duplicate data of sensors which are similar in the second level to reduce the amount of data sets that are to be transmitted to the cloud. The Figure 3.6 illustrate the flowchart of TEDaReT approach.

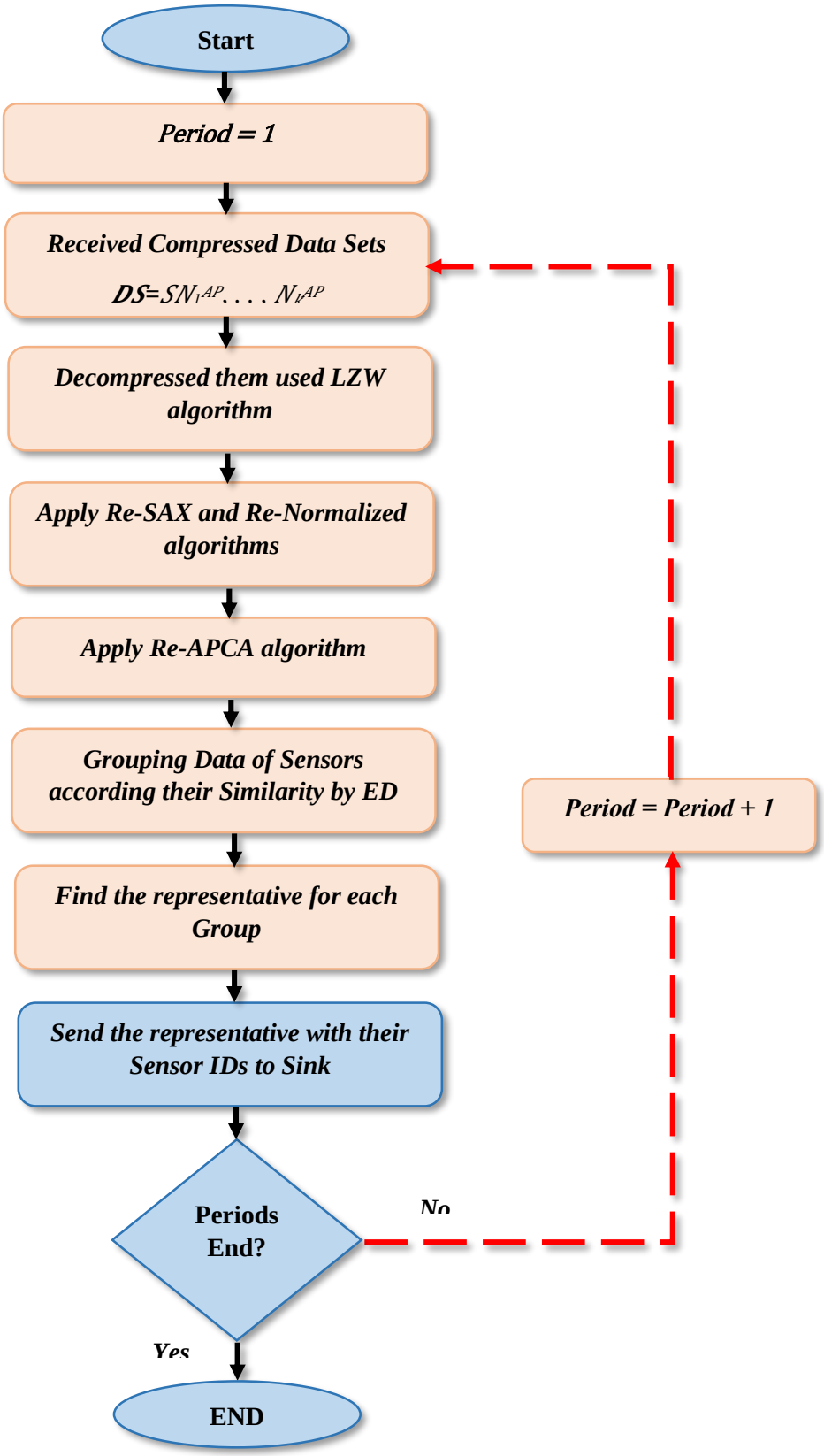


Figure 3.3. Flowchart of proposed TEDaReT approach

The TEDaReT approach is applied at the fog gateway to identify and discard duplicate sensor readings to reduce the amount of data of sensors. To reduce the fog gateway's total transmission and energy consumption, a similarity technique looks for correlations between sets of metrics. Since the sensor devices transmit varying sets of length measures to the fog gateway, a similarity measure is required to determine whether or not two sets of length measures are comparable.

Therefore, there was a need to return the data to its normal state before the reduction in order for it to be of the same length for each group, in order to find the similarity between the data groups using one of the methods of finding similarity. In our work, we used the Euclidean method to find the similarity ratio between the data set for the purpose of grouping the data within groups, and then we take the data set that has the highest similarity ratio to be representative of group.

For the purpose of returning the data before the reduction, this is done through the use of the De-LZW algorithm (algorithm 3.8) that returns the data to symbols, and uses the Re-SAX algorithm (algorithm 3.9), and uses the Re-APCA algorithm to get the original segmentation, then uses the Euclidean Distance approach to find the similarity in data sets depending on a specific threshold, if the similarity percentage between two sets is greater than the threshold this means It's from the same group, and so on. Thus, this step results in a group of data sets groups, where selected the data set that has the highest similarity among all data sets in the same group is representative of the group with save the IDs of the reduced data set in the same group. After the completion of the similarity process, all the compressed groups will be sent, with the IDs of all sensors that are excepted.

Algorithm 3.8: De-LZW Decompression

Input: *TCom : Token Compression, Dic: Dictionary of LZW*
Output: *Sym: Symbol Segments.*
Process:

```
1.  Sym ← []
2.  last ← len(Dic)
3.  p ← TCom[0]
4.  Sym ← (Dic[p])
5.  for i ← 0 to TCom do
6.  |    if TCom[i] in Dic then
7.  |    |    entry = Dic [TCom[i]]
8.  |    end if
9.  |    Sym ← entry
10. |    Dic [last] = Dic[p] + entry[0]
11. |    last ← last + 1
12. |    p = TCom[i]
13. end for
14. Return Sym
```

Algorithm 3.9: Re-SAX Representation

Input: *Sym: Set of Symbol, B (breakpoint values of table 1),*
Output: *Re-SAX: Data Segments.*
Process:

```
11. alphabet ← {'a' , 'b' , 'c' , 'd' , 'e' , 'f' , 'g' , 'h' , 'i' , 'j' }
12. char ← 10   # The no. of Symbol that represents the values
13. k ← 0
14. for i ← 0 to Sym do
15. |    for j ← 0 to len(B) do
16. |    |    if Sym[i] = alphabet[j] then
17. |    |    |    Re-SAX[k] ← Average ( B[j+1][char-1], B[j][char-1])
18. |    |    |    k ← k+1
19. |    |    end if
20. |    end for
21. end for
22. Return Re-SAX
```

3.3.2 Data of Sensors Grouping

Data of sensors grouping refers to the process of categorizing or grouping data of sensors based on specific criteria, such as shared characteristics or attributes. This can be useful in analyzing and organizing large data sets, as it allows for easier comparison and identification of patterns or trends within the data. Common methods for grouping data sets include grouping by similar features, such as demographic information or geographical location, or grouping by specific variables or attributes that are relevant to the analysis being conducted. In addition to facilitating data analysis, data set grouping can also be useful in data visualization, as it allows for the creation of charts or graphs that show how different groups or categories compare to each other. Overall, data set grouping is an important step in the data analysis process that can help to reveal important insights and inform decision-making. One of the most common approaches to Data of Sensors Grouping in Wireless Sensor Networks (WSNs) is to find similarities between data sets.

In algorithm 3.10, the data sets group is checking similarity of each data set with other sets, if there is similarity then considered in the same group, and then will find the similarity between sets in each group and take the highest similarity as representative of group with their IDs, and continuing for all groups.

Algorithm 3.10: Data of Sensors Grouping & Representative of each group

Input: ***DSs:*** *Data Sets,* ***thre:*** *Threshold of Similarity.*
Output: ***All-Group:*** *Vectors of all Groups of Data Sets with their IDs.*
Process:

```
k ← 0
Group ← [ ][ ]
for i ← 0 to len(DSs)-1 do
|    k ← k+1
|    l ← 0
|    if DSs[i] not in Group then
|    |    Group[k][l] ← DSs[i]
|    |    l ← l+1
|    endif
|    for j ← i+1 to len(DSs) do
|    |    Sim ← Similarity(DSs[i], DSs[j])   #call fun. Sim.
|    |    if Sim >= thre then
|    |    |    Group[k][l] ← DSs[j]
|    |    |    l ← l + 1
|    |    endif
|    endfor
end for
k ← 0
for i ← 1 to len(Group) do
|    All-Group ← (highest-Sim(Group[i]), IDs)
endfor
Return All-Group
```

3.3.3 Similarity Function

The clustering-based similarity approach is applied at the fog gateway to identify and discard duplicate sensor readings to reduce the amount of data sent from the sensors to the base station. To reduce the fog gateway's total transmission and energy consumption, a similarity technique looks for correlations between sets of metrics. Since the sensor devices transmit varying

sets of length measures to the fog gateway, a similarity measure is required to determine whether or not two sets of length measures are comparable.

The suggested clustering technique utilizes similarity criteria based on the **Euclidean Distance (see section 3.2.1.6)** to evaluate the degree of similarity between sets of equal length that have been received from various sensors. To reduce the amount of energy used and lengthen the lifespan of the PSN, the clustering technique is being used at the fog gateway to summarize the sets of measurements from sensor devices before sending them on to the base station. Since the data set that was received from the sensors from the first stage is unequal in length, therefore, is a need to return the data to its original position before reducing it to be of equal length for the purpose of finding similarity between the data set depending on a specific threshold.

3.4 Summary of The Chapter

In this chapter, we have studied the reduction the energy consumption to save the lifetime of PSNs by utilizing data reduction techniques. **Firstly**, we proposed four different approaches that work at the level of the sensor node. The primary concept behind the four approaches is to leverage the similarity between the gathered data's predictive value for a given period and the actual data for the subsequent period, if there is similarity then will not be sent to the Gateway, and if not similarity then will pre-processing to reduce the data by using reduction approaches and compress the data before sending them to the Gateway. The first approach (DEDaR) is regarded as an effective method to save power and reduce the amount of transferred data and thus extending the lifetime of the network while maintaining the accuracy of received data at the gateway to enable each sensor to make a prediction and compression on the data by using AR and Huffman Encoding technique. The second approach

(DiPCoM) is used prediction and data compression with different techniques such as ARIMA and LZW to reduce the amount of transferred data. The third approach (IDaPCoT) used the (AR) prediction technique and LZW data compression technique. The fourth approach (EDaRePE) its work is not very different from all previous approaches, it is used for the ARIMA prediction and Huffman Encoding. All approaches use preprocessing on data as APCA and SAX to reduce data before sending to Gateway and also use the Differential Encoding technique except the DEDaR approach, which is an efficient approach to lossless data compression by reducing the bits of data before sending them to Gateway.

Secondly, we suggest in the Fog Gateway level, prior to transmission to the base station, the sets of measures received from the sensor devices will go through a reduction process to eliminate any additional spatially correlated data. The fundamental concept underlying the second tier is to group sensors whose data are similar into groups based on a threshold. In each group, a representative of the group is taken by finding the similarity between the data of one group, and the representative is the sensor has the highest percentage of similarity. As a result, a representative of each group is sent with the index of sensors that were removed. In order to accomplish this, a similarity metric is employed, the suggested clustering technique utilizes similarity criteria based on the **Euclidean Distance** to evaluate the degree of similarity between sets of equal length that have been received from various sensors. The second level is divided into two steps: the first is data of sensors grouping depends on finding the similarity between the set of data read by sensors and the second is select the represent sensor of each set to transmit the data compressed and an index of sensors that been shorthanded. As a result, a representative of each group is sent with the index of sensors that were removed.

CHAPTER FOUR

SIMULATION RESULTS AND DISCUSSION

CHAPTER FOUR: SIMULATION RESULTS AND DISCUSSION

4.1 Introduction

The assessment of the performance as well as the results of the simulation are shown here in the form of graphs, and a discussion of the recommended approaches that were provided in Chapter 3 is included. The purpose of this research is twofold: first, to assess the efficacy of the proposed approaches through the utilization of factual sensor data and a diverse range of performance metrics; and second, to evaluate the offered approaches in relation to other competing approaches that belong to the same area.

4.2 Simulation Framework

The simulation results, analysis, and discussion in this chapter are used to assess and demonstrate the performances of the proposed techniques. The simulation experiments are carried out with the help of a custom simulator written in Python. In these simulation experiments, actual data from sensor nodes deployed at the Intel Berkeley Research Lab are used [99]. The Lab's PSN comprises 54 Mica2Dot sensors that are positioned according to the localization depicted in Figure 4.1, the Berkeley database contains data on date, time, epoch, moteid, voltage, humidity, temperature, voltage, and light acquired every 31 s from 54 sensors. In this case, epoch is a monotonically increasing sequence number from each mote. Two readings from the same epoch number were produced from different motes at the same time. There are some missing epochs in this data set. Moteids range from 1-54; data from some motes may be

missing or truncated. Temperature is in degrees Celsius. Humidity is temperature corrected relative humidity, ranging from 0-100%. Light is in Lux (a value of 1 Lux corresponds to moonlight, 400 Lux to a bright office, and 100,000 Lux to full sunlight.) Voltage is expressed in volts, ranging from 2-3; the batteries in this case were lithium ion cells which maintain a fairly constant voltage over their lifetime; note that variations in voltage are highly correlated with temperature.

The gateway is in the middle of the building. Mica2Dot sensor nodes in the lab have already received over 2.3 million readings, and we u tilize them in our simulations. Here, we utilize temperature and humidity readings taken from sensor nodes as our primary metrics. Each sensor node in Figure 4.1 is labeled in yellow to indicate that it is being left out of our simulation since its data may be incomplete or incorrect. As a result, we choose and save the temperature and humidity recorded by 47 different sensor nodes. Those numbers represent an average across 47 different sensors.

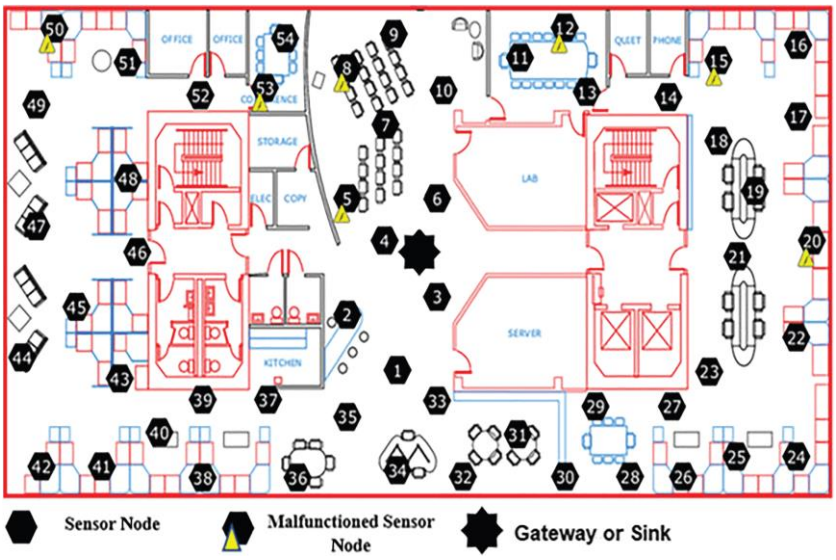


Figure 4.1: Intel Berkeley Research Lab

The energy consumption model called the "first order radio model," proposed by Heinzelman.41 was used. Figure 4.2 presents the first order radio model.

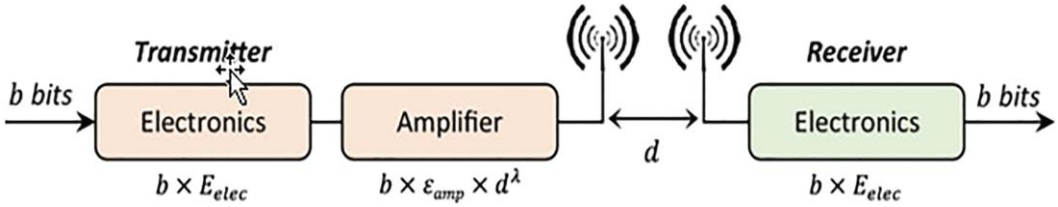


Figure 4.2: First order radio model

4.3 DEDaR Approach Performance Evaluation

To study and evaluate the DEDaR approach, we will employ 10,000 sensed data measures from three sensors in this work, and the focus will be met on one type of sensor measurement: humidity, for the purpose of simplicity (Sensor1, Sensor 2, and Sensor 3). The captured reading is of length 64 bits; therefore, the size of the packet is the number of readings multiplied by 64 bits. Table 4.1 presents the simulation parameters.

Table 4.1: Presents the Simulation Parameters Values.

Parameter	Value
sensed data measures from sensors S1, S2, and S3	10000
e_max	0.03, 0.05, 0.07, and 0.09
Eelec	50 * 10-9
εamp	100 *10-12
Sim_th	70%
Captured Reading length	64 bits

In order to show the high efficiency of the proposed DEDaR approach, it is compares with some related recent existing methods such as DPDR [35], DP LSTM [36], DDR-IoT [23], and LMS [37]. The comparison is achieved based

on some performance metrics such as the percentage of data reduction, data accuracy, transmitted data size, energy consumption, and network lifetime. The threshold ***Sim_th*** is selected after achieving several experiments by the proposed DEDaR approach and by using different ***Sim_th*** ratios. Figures 4.3, 4.4, and 4.5 show the transmitted data, energy consumption, and accuracy for Sensor1, Sensor 2, and Sensor 3, respectively. It can be seen from the results in the Figures 4.6, 4.7, and 4.8 that the suitable ratio for the threshold ***Sim_th*** is 70% because the DEDaR approach introduces better results from the energy consumption and accuracy point of view. The other ratios sometimes provide a slightly lower energy consumption but with lower accuracy. In this thesis, the similarity threshold ***Sim_th*** ratio was selected, which balances between energy consumption, and data accuracy. In real world applications, this similarity threshold ***Sim_th*** ratio will be selected according to the application's need.

Performance metrics are utilized in experimental simulations to evaluate the efficacy of the DEDaR approach. These metrics include:

- **Data Reduction,**
- **Number of Sending Readings,**
- **Energy Consumption, and**
- **Data Accuracy.**

Other performance metrics presents further results, analysis, and discussions about the DEDaR approach uses the temperature readings during the simulation, such as:

- **Transmitted Readings,**
- **Energy Consumption, and**
- **Data Loss Percentage.**

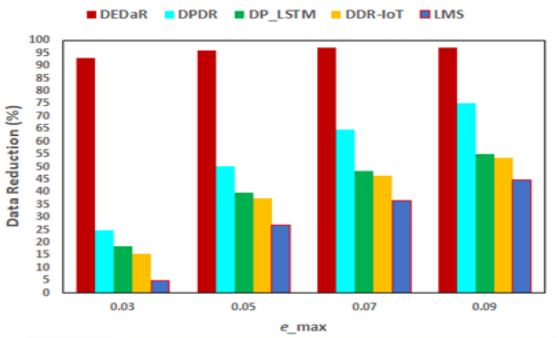
4.3.1 Data Reduction

This experiment investigates the efficiency of the proposed DEDaR approach in removing redundant data and compressing sensed data before sending it to the gateway. The data reduction percentage can be calculated as in Equations (**4.1 and 4.2**).

$$\mathbf{TTR = TCR - NTR} \quad \mathbf{(4.1)}$$

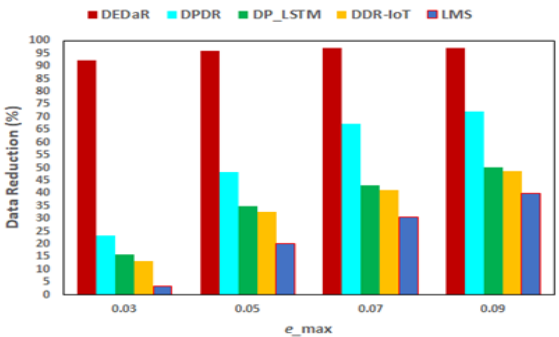
$$\mathbf{DPR} = \left|\left(\frac{\mathbf{TTR}}{\mathbf{TCR}} * \mathbf{100}\right) - \mathbf{100}\right| \quad \mathbf{(4.2)}$$

Where TCR, TTR, NTR, and DRP refer to the Total Collected Readings, Total Transmitted Readings, Non-Transmitted Readings, and the Data Reduction Percentage respectively. Figure 4.3 shows the data reduction percentage for different methods on the sensed data of sensors 1, 2, and 3 using different ***e_max*** values. It can be seen from the results of Figure 4.3 that the proposed DEDaR approach increased the percentage of reduction from 92.48% up to 97.58%, while the other methods introduced a percentage of reduction from 23.42% up to 75.75%, from 16.04% up to 54.99%, from 13.41% up to 53.39%, and from 3.33% up to 45.34% for DPDR, DP_LSTM, DDRIoT, and LMS, respectively. The proposed DEDaR approach achieves better performances compared to other approaches, and it reduces the sensed data efficiently by removing the redundant data before sending it to the gateway.



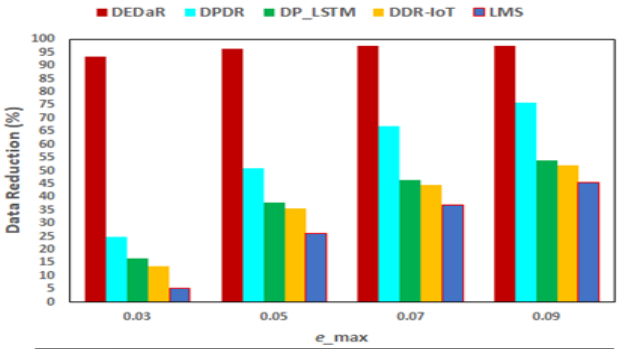
e-max	Data Reduction (%)				
	DEDaR	DPDR	DP_LSTM	DDR-IoT	LMS
0.03	93.14	24.92	18.4	15.61	4.92
0.05	96.19	50.34	39.61	37.55	26.92
0.07	97.15	64.67	48.39	46.54	36.49
0.09	97.3	75.14	54.99	53.39	44.71

a-Sensor1



e-max	Data Reduction (%)				
	DEDaR	DPDR	DP_LSTM	DDR-IoT	LMS
0.03	92.48	23.42	16.04	13.41	3.33
0.05	95.98	48.4	34.98	32.71	20.31
0.07	97.15	67.34	43.26	41.08	30.61
0.09	97.19	72.16	50.27	48.57	39.77

b-Sensor2



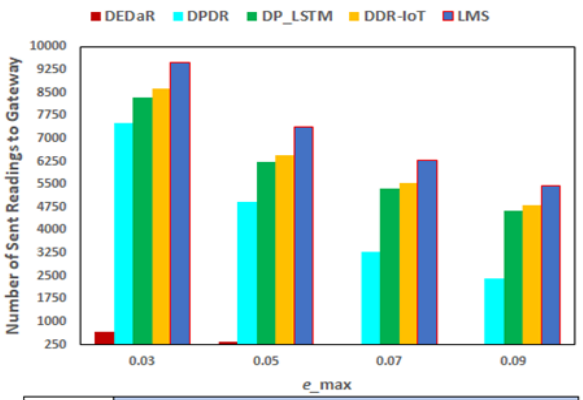
e-max	Data Reduction (%)				
	DEDaR	DPDR	DP_LSTM	DDR-IoT	LMS
0.03	93.44	24.99	16.72	13.51	5.27
0.05	96.45	50.74	37.78	35.48	26.31
0.07	97.43	66.95	46.52	44.55	37.07
0.09	97.58	75.75	53.75	51.91	45.34

c-Sensor3

Figure 4.3: Data reduction percentage of Sensor1, 2, and 3.

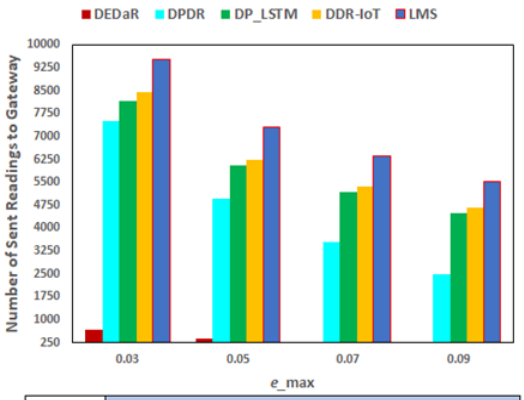
4.3.2 Number of Sent Readings

In this experiment, the volume of sent data readings to the gateway is studied. Figure 4.4 introduces the number of sent readings to the gateway after applying the proposed DEDaR approach.



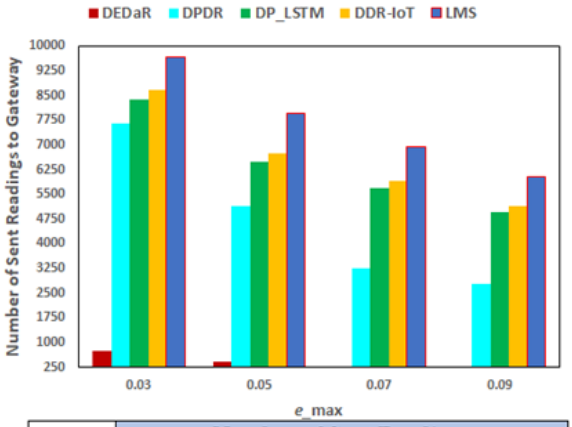
e-max	Number of Sent Readings				
	DEDaR	DPDR	DP_LSTM	DDR-IoT	LMS
0.03	656	7501	8328	8649	9473
0.05	355	4926	6222	6452	7369
0.07	257	3305	5348	5545	6293
0.09	242	2425	4625	4809	5466

a- Sensor1



e-max	Number of Sent Readings				
	DEDaR	DPDR	DP_LSTM	DDR-IoT	LMS
0.03	686	7508	8160	8439	9508
0.05	381	4966	6039	6245	7308
0.07	285	3533	5161	5346	6351
0.09	270	2486	4501	4661	5529

b- Sensor2



e-max	Number of Sent Readings				
	DEDaR	DPDR	DP_LSTM	DDR-IoT	LMS
0.03	752	7658	8396	8659	9667
0.05	402	5160	6502	6729	7969
0.07	285	3266	5674	5892	6939
0.09	281	2784	4973	5143	6023

c- Sensor3

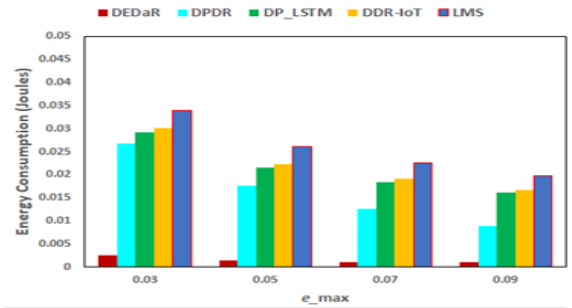
Figure 4.4: Number of Sent Readings of Sensor1, 2, and 3.

From the results in Figure 4.4, it can be observed that the proposed DEDaR approach reduces the transmitted data to the gateway from 2.42% up to 7.52% compared with other approaches. The DPDR, DP_LSTM, DDR-IoT, and LMS reduced the transmission to the gateway from 24.25% up to 76.58%, from 45.01% up to 83.96%, from 46.61% up to 86.59%, and from 53.66% up to

96.67%, respectively. This can ensure that the proposed DEDaR approach is more powerful in removing the redundant data before sending it to the gateway.

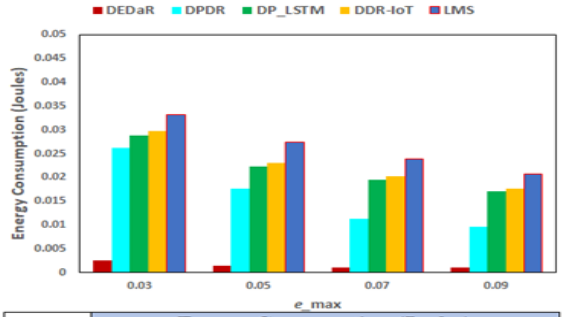
4.3.3 Energy Consumption

Due to the limited resources of sensor nodes, energy consumption is critical in these devices of IoT networks. Figure 4.5 refers to the consumed energy inside each sensor node.



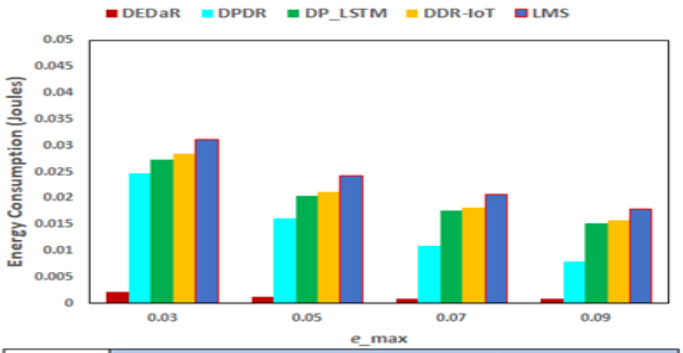
e-max	Energy Consumption (Joules)				
	DEDaR	DPDR	DP_LSTM	DDR-IoT	LMS
0.03	0.0024	0.0268	0.0291	0.0301	0.0339
0.05	0.0014	0.0177	0.0215	0.0223	0.0261
0.07	0.0010	0.0126	0.0184	0.0191	0.0227
0.09	0.0010	0.0089	0.0161	0.0166	0.0197

a- Sensor1



e-max	Energy Consumption (Joules)				
	DEDaR	DPDR	DP_LSTM	DDR-IoT	LMS
0.03	0.0026	0.0263	0.0288	0.0297	0.0332
0.05	0.0014	0.0177	0.0223	0.0231	0.0273
0.07	0.0010	0.0112	0.0195	0.0202	0.0238
0.09	0.0010	0.0096	0.0171	0.0177	0.0207

b- Sensor2



e-max	Energy Consumption (Joules)				
	DEDaR	DPDR	DP_LSTM	DDR-IoT	LMS
0.03	0.0022	0.0246	0.0274	0.0284	0.0311
0.05	0.0012	0.0162	0.0204	0.0212	0.0242
0.07	0.0008	0.0109	0.0176	0.0182	0.0207
0.09	0.0008	0.0080	0.0152	0.0158	0.0180

c- Sensor3

Figure 4.5: Energy Consumption of Sensor1, 2, and 3.

It can be noticed from the results in Figure 4.5 that the proposed DEDaR approach decreased the consumed energy from 90% up to 90.3%, from 91.1% up to 94.74%, from 91.4% up to 94.94%, and from 92.33% up to 95.56% in comparison with DPDR, DP_LSTM, DDR-IoT, and LMS, respectively. It can be concluded from the results that the proposed DEDaR approach saves more energy than other methods due to transmitting as little as few data as possible readings to the gateway. This improves the performances of the network and extends its lifetime.

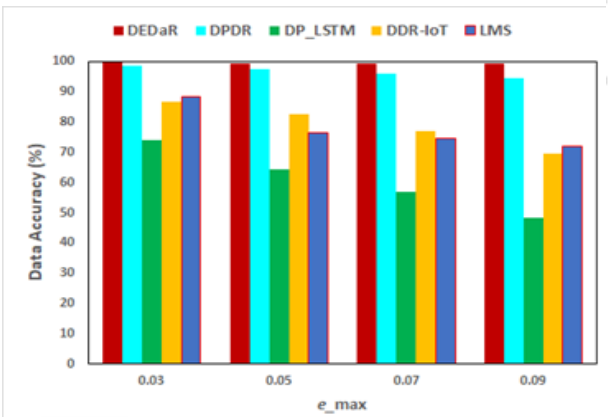
4.3.4 Data Accuracy

Data accuracy can be calculated by computing the deviation between the gathered (input data) and transmitted data readings (obtained results) after employing the reduction algorithm, including the prediction algorithm. Equations (4.3, and 4.4) are used to produce the percentage of accuracy.

$$DD = \left|\frac{\sum TCR - (\sum TTR + \sum TER)}{TCR}\right| * 100 \quad \textbf{(4.3)}$$

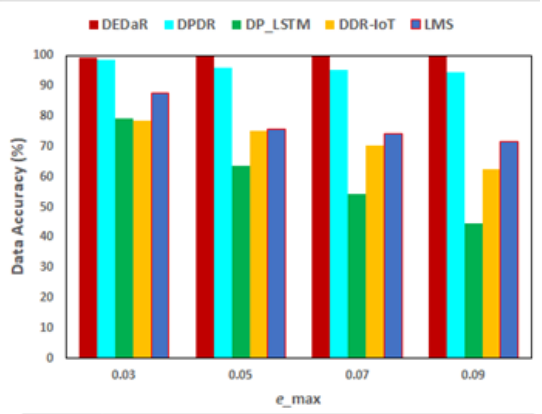
$$DA = (1 - DD) * 100 \quad \textbf{(4.4)}$$

The data deviation is DD, the total estimated readings of the sensor ***Si*** is TER, and the data accuracy is DA. The total transmitted readings are denoted by TTR, while the total collected readings are denoted by TCR. Figure 4.6 shows the data accuracy.



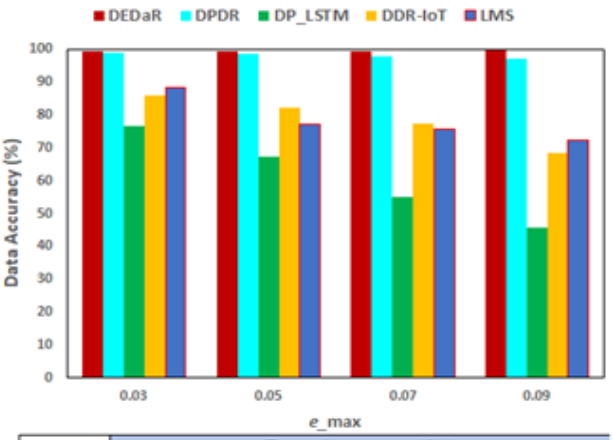
e-max	Data Accuracy (%)				
	DEDaR	DPDR	DP_LSTM	DDR-IoT	LMS
0.03	99.74	98.8	74.12	86.83	88.42
0.05	99.46	97.66	64.23	82.77	76.39
0.07	99.37	96.07	56.81	77.12	74.51
0.09	99.33	94.56	48.26	69.41	71.92

a- Sensor1



e-max	Data Accuracy (%)				
	DEDaR	DPDR	DEDaR	DDR-IoT	DEDaR
0.03	99.45	98.71	79.18	78.7	87.6
0.05	99.68	96.23	63.45	75.22	75.67
0.07	99.73	95.16	54.38	70.24	74.16
0.09	99.72	94.5	44.71	62.6	71.54

b- Sensor2



e-max	Data Accuracy (%)				
	DEDaR	DPDR	DEDaR	DDR-IoT	DEDaR
0.03	99.22	99.06	76.79	86.13	88.37
0.05	99.57	98.52	67.31	82.36	77.09
0.07	99.5	97.85	55.09	77.43	75.72
0.09	99.79	97.18	45.75	68.56	72.49

c- Sensor3

Figure 4.6: Data Accuracy of Sensor1, 2, and 3.

It can be seen from the results of Figure 4.6 that the proposed DEDaR approach introduces a suitable data accuracy of 92.48% up to 99.74% compared with other methods introduced of 23.42% up to 75.75%, from 16.04% up to 54.99%, from 13.41% up to 53.39%, and from 3.33% up to 45.34% for DPDR,

DP_LSTM, DDR-IoT, and LMS, respectively. This will ensure that the proposed DEDaR approach provides a high reduction with suitable data accuracy at the gateway. The proposed DEDaR produces this high ratio of data accuracy due to losing a lower number of data readings at the sensor node. It uses efficient techniques to remove redundant data from the sensor node while maintaining adequate data quality at the gateway. The DEDaR approach uses a fixed code dictionary based on lossless Huffman encoding to compress the segments of the SAX technique, and this decreases the lost data and increases the accuracy of received data at the gateway.

4.3.5 Further Results and Discussion

This section presents further results, analysis, and discussions about the DEDaR approach, and the findings are compared with some related works to show the efficiency of the proposed DEDaR approach. In the next experiments, the DEDaR approach uses the temperature readings during the simulation. It uses different sizes per period such as T= 20, 50, and 100 readings per period. Moreover, the e_max uses different values like 0.03, 0.05, and 0.07. The DEDaR approach is compared with DaReCA [33], prefix frequency filtering (PFF) [38], and aggregation and transmission approach (ATP) [39].

4.3.5.1 Transmitted Readings

It can be observed from the results presented in Figure 4.7 that the DEDaR, DaReCA, ATP, and PFF approaches sent data from 1.63% up to 6.06%, from 3.2% up to 5.9%, from 13.03% up to 31.68%, and 100% respectively. It can be seen from Figure 4.7 that the proposed DEDaR approach reduced the transmitted data efficiently compared with other approaches. These findings

ensure the efficiency of combining the prediction and compression methods in decreasing the volume of transmitted data by the sensor node.

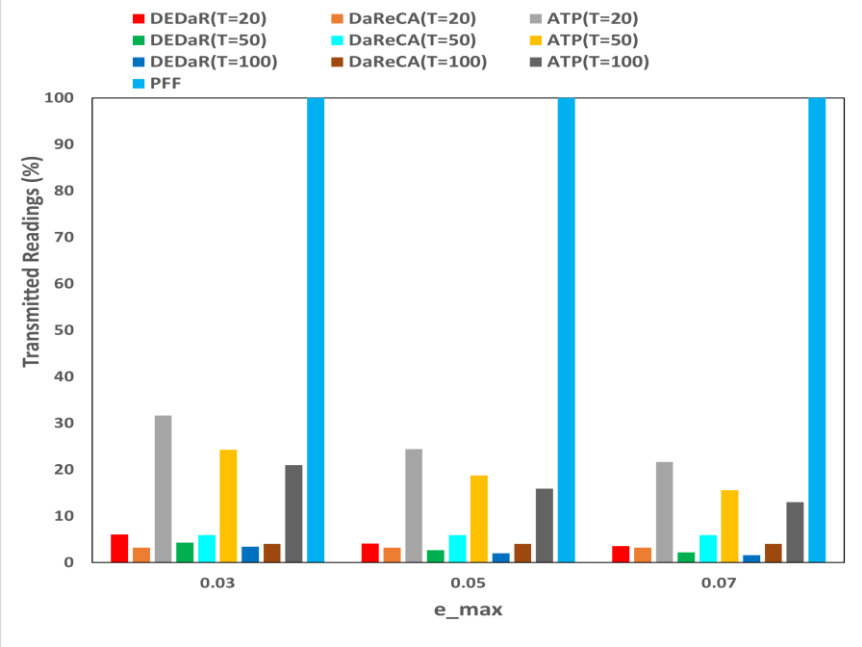


Figure 4.7: Transmitted Readings

4.3.5.2 Energy Consumption

Figure 4.8 refers to the energy consumed by the sensor node using the DEDaR approach compared with DaReCA, ATP, and PFF approaches. It can be seen from the results in Figure 4.8 that the proposed DEDaR approach lowered the consumed energy by the sensor node from 92.82% up to 94.73%, from 97.77% up to 98.13%, and 98.15% up to 98.34% compared with DaReCA, ATP, and PFF methods. The high performance of the suggested DEDaR in reducing this volume of energy due to its use of an efficient data reduction method that reduces the sensed data before sending it by the sensor node (see Figure 4.7).

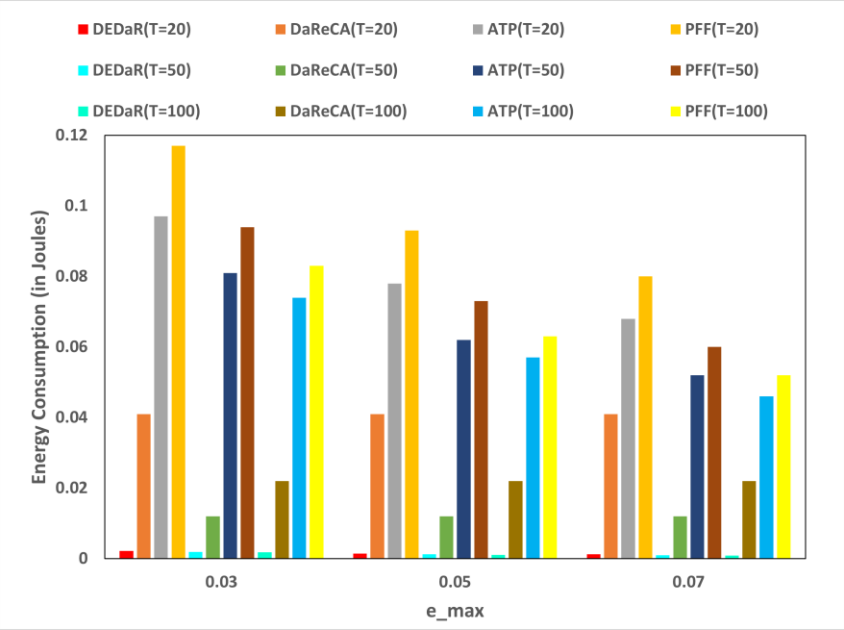


Figure 4.8: Energy Consumption

4.3.5.3 Data Loss Percentage

It is critical to lower the size of transferred data before sending them to the gateway in order to save energy, but it is also important to ensure an adequate rate of data quality at the gateway. The percentage of the data loss represents as an indicator for the data accuracy. Figure 4.9 refers to the percentage of the data loss.

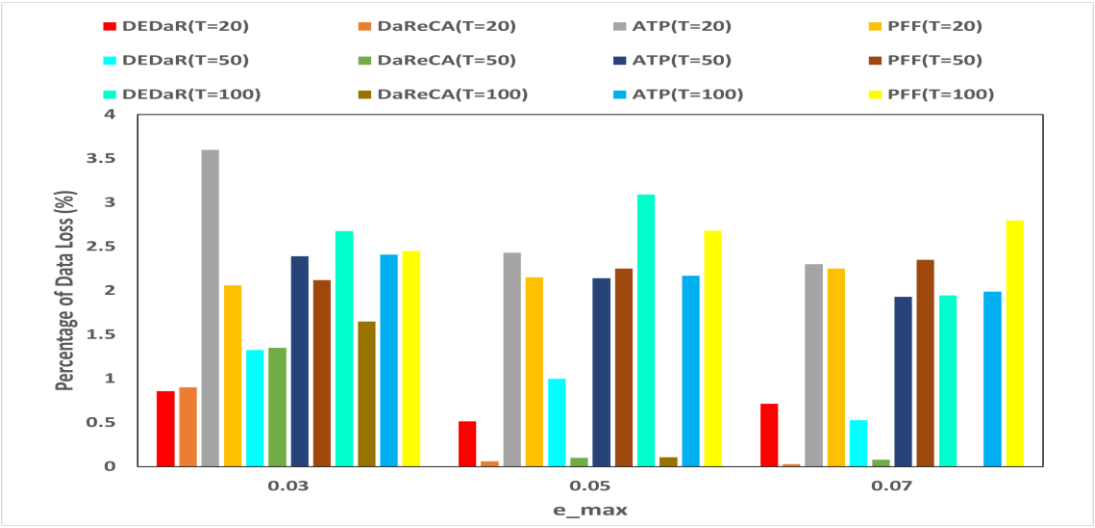


Figure 4.9: Data Loss Percentage

It can be noted that the DEDaR approach decreased the lost data from 0.52% up to 3.09%, while the DaReCA, ATP and PFF reduced the lost data from 0.03% up to 1.65%, from 1.93% up to 3.6% and from 2.06% up to 2.68% respectively. The proposed DEDaR approach introduced better data accuracy in most cases compared with other methods in spite of achieving higher data reduction at the sensor node. The DaReCA method introduces lower data loss in some cases because it sends a larger volume of data when it is implemented at the sensor node that participates in reducing the lost data. Hence, the proposed DEDaR could further decrease the sensed data and keep the energy of sensor nodes while preserving a suitable level of data quality.

4.4 DiPCoM Approach Performance Evaluation

This section achieved several experiments to evaluate the effectiveness of the DiPCoM technique. These simulation results are conducted using a custom simulator built in Python. Actual data from sensor nodes placed at the Intel Berkeley Research Lab are used in these simulation studies [43]. The Berkeley database comprises information on voltage, humidity, temperature, and light that were collected every 31 seconds from 54 sensors, as shown in Figure 4.1. In this study, we used three sensors, 10,000 sensed data are taken from each sensor, with the emphasis on only one kind of sensor measurement: humidity, for the sake of simplicity (Sensor1, Sensor 2, and Sensor 3). The building's sink, or Gateway, is in the center.

The proposed DiPCoM is compared to some current existing approaches, including LMS [37], DDR-IoT [23], DP-LSTM [36], and DPDR [35]. Metrics such as percentage of data reduction, data accuracy, size of transmitted data, network lifespan, and energy consumption are used to assess the performance of all algorithms. The ***Sim-th*** threshold is chosen after multiple attempts using

the proposed DiPCoM method and various rates of ***Sim-th***. Other rates can sometimes offer a somewhat lower consumed power, but with less accuracy. This work made use of the ***Sim-th*** similarity criterion. It finds a good balance between energy use and data accuracy. This ***Sim-th*** ratio for similarity threshold will be used in real-world applications based on the requirements of the application. Performance metrics are utilized in experimental simulations to evaluate the efficacy of the DiPCoM approach. These metrics include:

- **Data Reduction,**
- **Number of sending readings,**
- **Energy consumption, and**
- **Data Accuracy.**

Other performance metrics present further results, analysis, and discussions about the DEDaR approach uses the temperature readings during the simulation, such as:

- **Transmitted readings,**
- **Energy consumption, and**
- **Data loss percentage.**

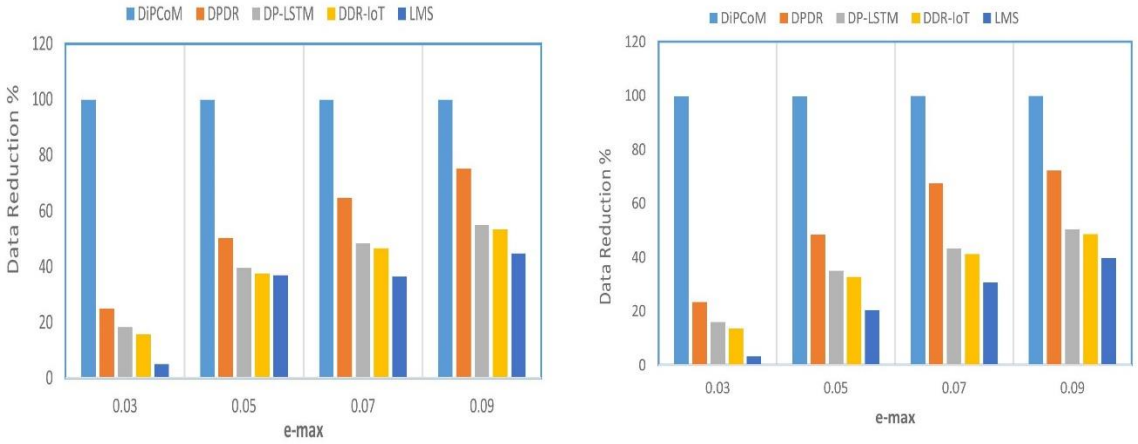
4.4.1 Data Reduction

The experiment aims to evaluate the effectiveness of the suggested DiPCoM technique in reducing and compressing sensed data prior to transmission to the gateway. Equations (4.5 and 4.6) can be used to compute the percentage of data reduction.

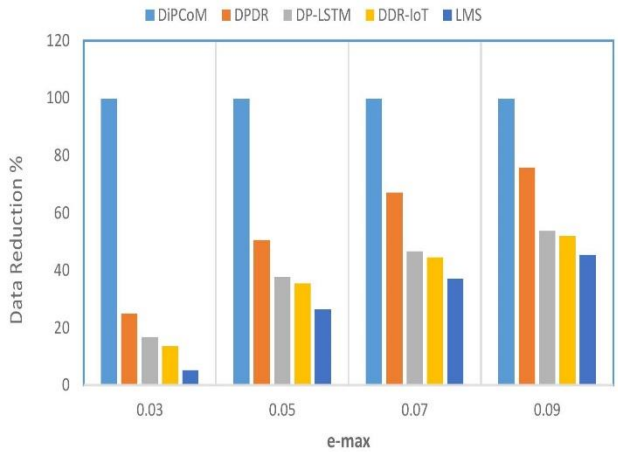
$$S^{AP} = \text{OSD} = \text{OCD} - \text{NSD} \qquad (4.5)$$

$$\boldsymbol{DRP} = \left|\left(\frac{\boldsymbol{OSD}}{\boldsymbol{OCD}} * \mathbf{100}\right) - \mathbf{100}\right| \qquad \textbf{(4.6)}$$

Where DRP, OCD, NSD, and OSD are acronyms for data reduction percentage, overall captured data, non-sent data, and overall sent data, respectively. Figure (4.10) depicts the data reduction percentage for several approaches using data obtained from sensors 1, 2, and 3 with various e_max values.



a- Sensor1 **b- Sensor2**



c- Sensor3

Figure 4.10: Data Reduction of Sensor1, 2, and 3.

Tables 4.2, 4.3, and 4.4 show the data reduction percentage for the collected humidity readings by the sensor devices 1, 2, and 3, respectively. Figure (4.10) illustrates that the proposed DiPCoM approach reduced the reduction ratio from 99.71% up to 99.76% over sensor devices 1, 2, and 3. The DPDR, DPLSTM, DDR-IoT, and LMS methods decreased the reduction ratio from 23.42% up to 75.75%, from 16.04% up to 54.99%, from 13.41% up to 53.39%, and from 3.33% up to 45.34%, respectively, over sensor devices 1, 2, and 3. The suggested DiPCoM method improves performance and reduces the size of data by getting rid of redundant data before transferring them to the gateway.

Table 4.2: Data Reduction (Sensor Device 1)

e-max	Data Reduction %				
	DiPCoM	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.751	24.92	18.4	15.61	4.92
0.05	99.750	50.34	39.61	37.55	36.92
0.07	99.753	64.67	48.39	46.54	36.49
0.09	99.753	75.14	54.99	53.39	44.71

Table 4.3: Data Reduction (Sensor Device 2)

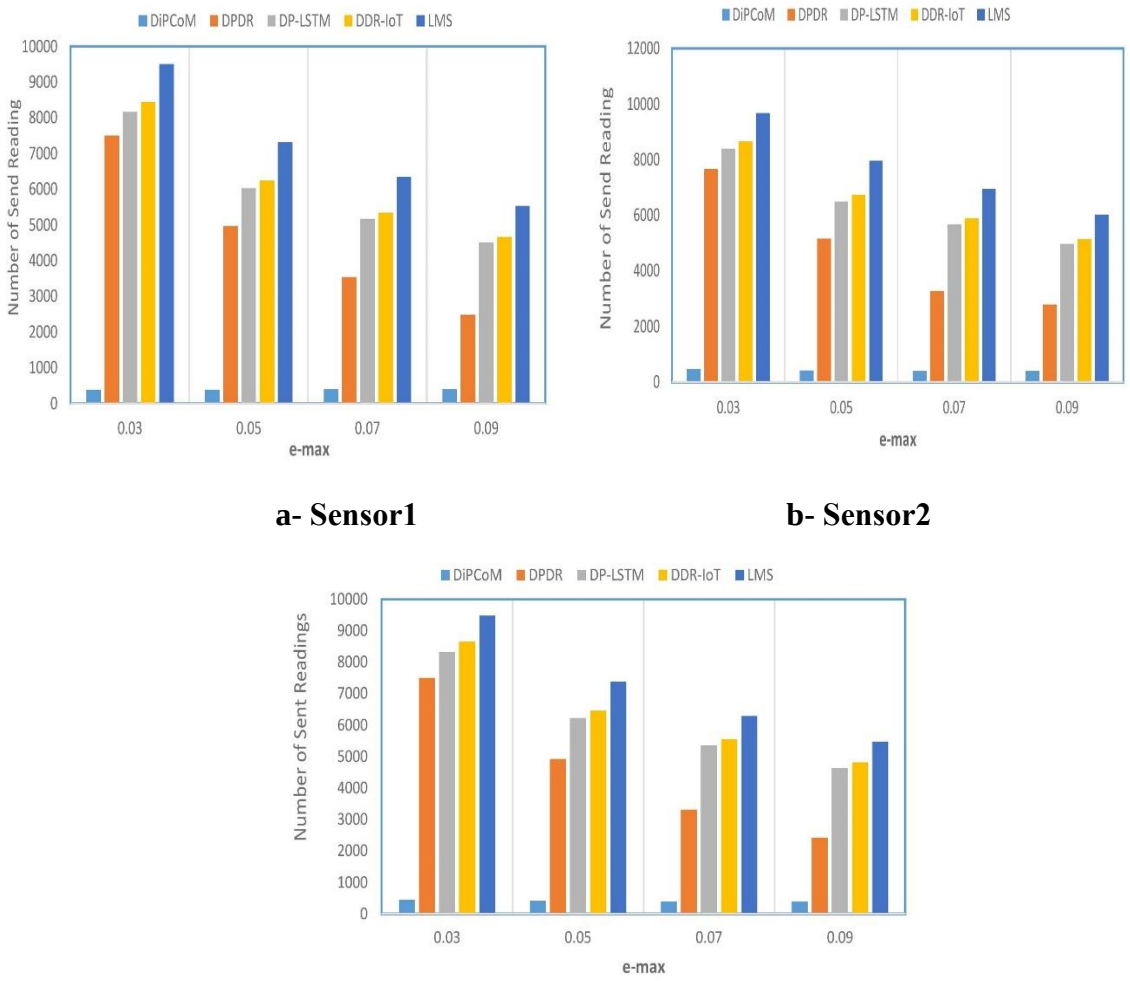
e-max	Data Reduction %				
	DiPCoM	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.71	23.42	16.04	13.41	3.33
0.05	99.74	48.4	34.98	32.71	20.31
0.07	99.76	67.34	43.26	41.08	30.61
0.09	99.76	72.16	50.27	48.57	39.77

Table 4.4: Data Reduction (Sensor Device 3)

e-max	Data Reduction %				
	DiPCoM	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.72	24.99	16.72	13.51	5.27
0.05	99.74	50.47	37.78	35.48	26.31
0.07	99.75	66.95	46.52	44.55	37.07
0.09	99.75	75.75	53.75	51.91	45.34

4.4.2 Number of Sent Readings

In this study, the number of data transmitted to the gateway is being investigated. Figure (4.11) shows how many measurements were sent to the gateway after the DiPCoM technique was used.



a- Sensor1 **b- Sensor2**

c- Sensor3

Figure 4.11: Number of Sent Readings of Sensor1, 2, and 3.

Tables 4.5, 4.6, and 4.7 show the number of sent humidity readings by sensor devices 1, 2, and 3, respectively. It can be observed from Figure (4.11) that the DiPCoM approach decreases the amount of data transmitted to the

gateway from 381 up to 465 readings over sensor devices 1, 2, and 3. The DPDR, DP-LSTM, DDR-IoT, and LMS methods reduced the number of readings from 2425 up to 7658, from 4501 up to 8396, from 4661 up to 8659, and from 5466 up to 9667, respectively, over sensor devices 1, 2, and 3. As a result, the proposed DiPCoM approach outperforms other methods because it effectively removes duplicate data before sending them to the gateway.

Table 4.5: Number of Send Reading (Sensor Device 1)

e-max	Number of Send Reading				
	DiPCoM	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	381	7508	8160	8439	9508
0.05	387	4966	6039	6245	7308
0.07	395	3533	5161	5346	6351
0.09	395	2486	4501	4661	5529

Table 4.6: Number of Send Reading (Sensor Device 2)

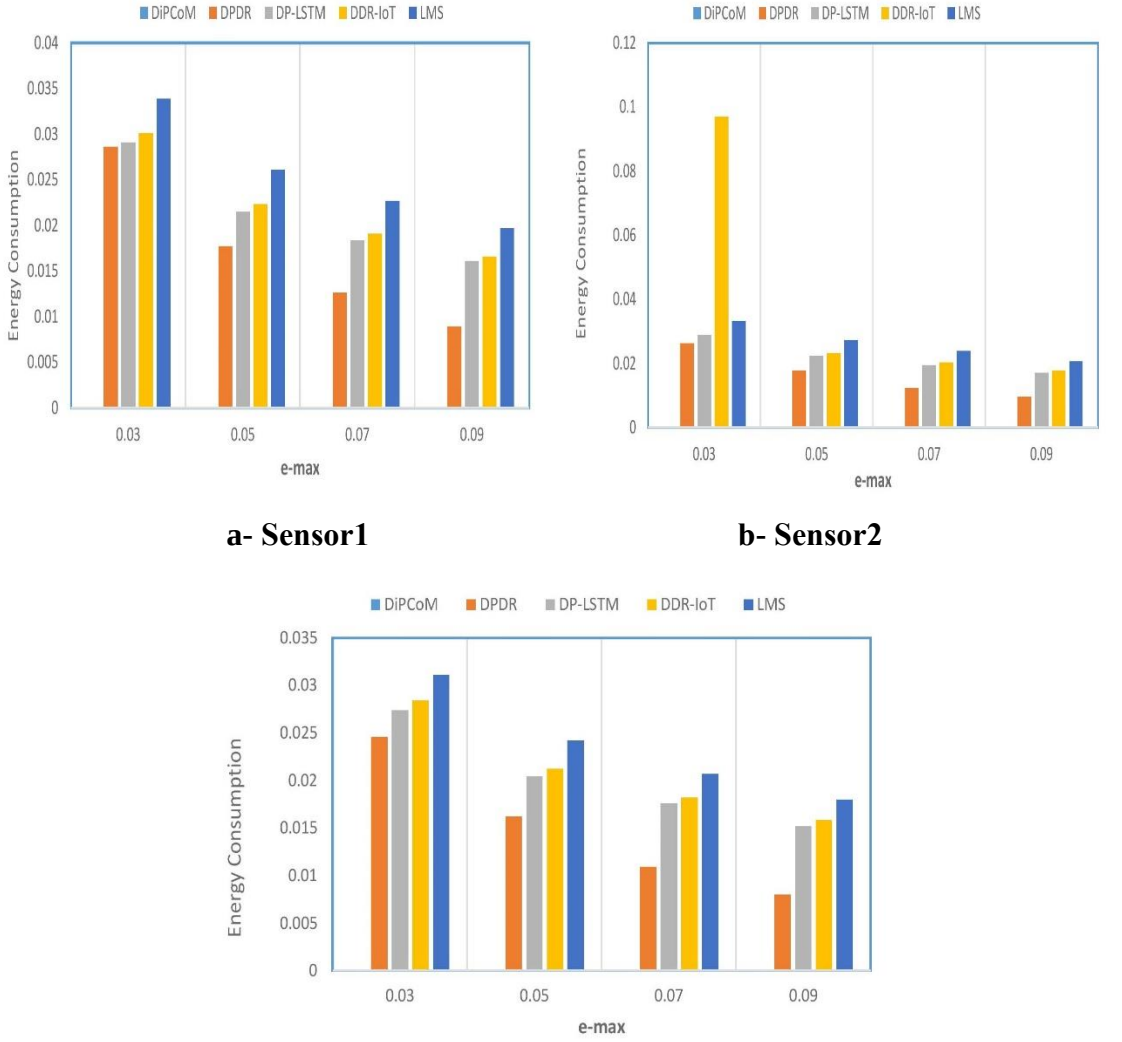
e-max	Number of Send Reading				
	DiPCoM	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	465	7658	8396	8659	9667
0.05	410	5160	6502	6729	7969
0.07	386	3266	5674	5892	6939
0.09	386	2784	4973	5143	6023

Table 4.7: Number of Send Reading (Sensor Device 3)

e-max	Number of Send Reading				
	DiPCoM	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	449	7501	8328	8649	9473
0.05	411	4926	6222	6452	7369
0.07	399	3305	5348	5545	6293
0.09	395	2425	4625	4809	5466

4.4.3 Energy Consumption

Power is a crucial component of sensor devices due to the constrained resources of the sensor nodes. Figure (4.12) represents the amount of energy used by every sensor device.



a- Sensor1 **b- Sensor2**

c- Sensor3

Figure 4.12: Energy Consumption of Sensor1, 2, and 3.

Tables 4.8, 4.9, and 4.10 show the consumed energy by sensor devices 1, 2, and 3, respectively.

Table 4.8: Energy Consumption (Sensor Device 1)

e-max	Energy Consumption				
	DiPCoM	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	0.0000209	0.0286	0.0291	0.0301	0.0339
0.05	0.0000212	0.0177	0.0215	0.0223	0.0261
0.07	0.0000217	0.0126	0.0184	0.0191	0.0227
0.09	0.0000217	0.0089	0.0161	0.0166	0.0197

Table 4.9: Energy Consumption (Sensor Device 2)

e-max	Energy Consumption				
	DiPCoM	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	0.0000255	0.0263	0.0288	0.097	0.0332
0.05	0.0000225	0.0177	0.0223	0.0231	0.0273
0.07	0.0000212	0.0122	0.0195	0.0202	0.0238
0.09	0.0000212	0.0096	0.0171	0.0177	0.0207

Table 4.10: Energy Consumption (Sensor Device 3)

e-max	Energy Consumption				
	DiPCoM	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	0.0000247	0.0246	0.0274	0.0284	0.0311
0.05	0.000026	0.0162	0.0204	0.0212	0.0242
0.07	0.0000219	0.0109	0.0176	0.0182	0.0207
0.09	0.0000217	0.008	0.0152	0.0158	0.018

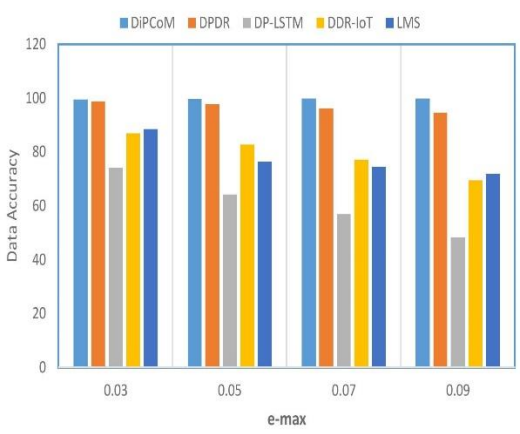
4.4.4 Data Accuracy

It is important to have acceptable data accuracy at the gateway after eliminating the redundant data at the sensor device level. In this paper, the accuracy ratio is computed using equations (4.7) and (4.8).

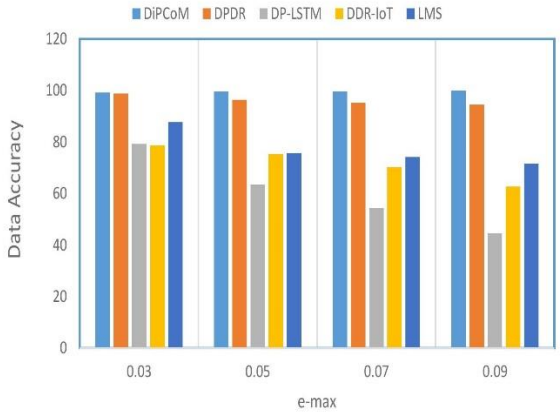
$$DD = \left|\frac{\sum OCD - (\sum OSD + \sum OED)}{OCD}\right| * 100 \quad (4.7)$$

$$DA = (1 - DD) * 100 \quad (4.8)$$

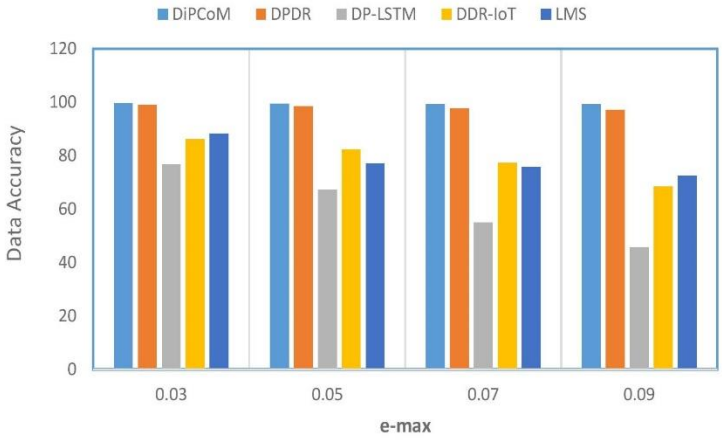
Where OED, DA, OSD, OCD, and DD refer to the overall estimated data, data accuracy, overall sent data, overall captured data, and data deviation, respectively. The data accuracy is displayed in Figure 4.13.



a- Sensor1



b- Sensor2



c- Sensor3

Figure 4.13: Data Accuracy of Sensors 1, 2, and 3.

Tables 4.11, 4.12, and 4.13 show the data accuracy at the gateway after eliminating the redundant humidity data by sensor devices 1, 2, and 3, respectively. It can be observed that the DiPCoM improves the data accuracy

from 99.22% up to 99.79% over sensor devices 1, 2, and 3. But the DPDR, DP-LSTM, DDR-IoT, and LMS methods introduced data accuracy from 94.5% up to 99.06%, from 44.71% up to 79.18%, from 62.6% up to 86.83%, and from 71.54% up to 88.42% compared with the DP-LSTM, DPDR, DDR-IoT, and LMS, respectively, over the sensor devices 1, 2, and 3. Therefore, the proposed DiPCoM will ensure high data reduction while appropriate data accuracy is preserved at the gateway.

Table 4.11: Data Accuracy (Sensor Device 1)

e-max	Data Accuracy				
	DiPCoM	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.45	98.8	74.12	86.83	88.42
0.05	99.68	97.66	64.23	82.77	76.39
0.07	99.73	96.07	56.81	77.12	74.51
0.09	99.72	94.56	48.26	69.41	71.92

Table 4.12: Data Accuracy (Sensor Device 2)

e-max	Data Accuracy				
	DiPCoM	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.22	98.71	79.18	78.7	87.6
0.05	99.57	96.23	63.45	75.22	75.67
0.07	99.50	95.16	54.38	70.24	74.16
0.09	99.79	94.5	44.71	62.6	71.54

Table 4.13: Data Accuracy (Sensor Device 3)

e-max	Data Accuracy				
	DiPCoM	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.74	99.06	76.79	86.13	88.37
0.05	99.46	98.52	67.31	82.36	77.09
0.07	99.37	97.85	55.09	77.43	75.72
0.09	99.34	97.18	45.75	68.56	72.49

4.4.5 Further Results and Discussion

To demonstrate the effectiveness of the DiPCoM, more results, analyses, and discussions are presented in this section as well as comparisons with recent

publications. The DiPCoM technique uses temperature measurements throughout the simulation in the following experiments. It uses several sizes for each period, including T=20, 50, and 100 readings. Additionally, the ***e-max*** uses other values, including 0.03, 0.05, and 0.07. The DaReCA [33], PFF [38], and ATP [39] approaches are compared with the proposed DiPCoM approach. Figure 4.14 illustrates the transmitted readings.

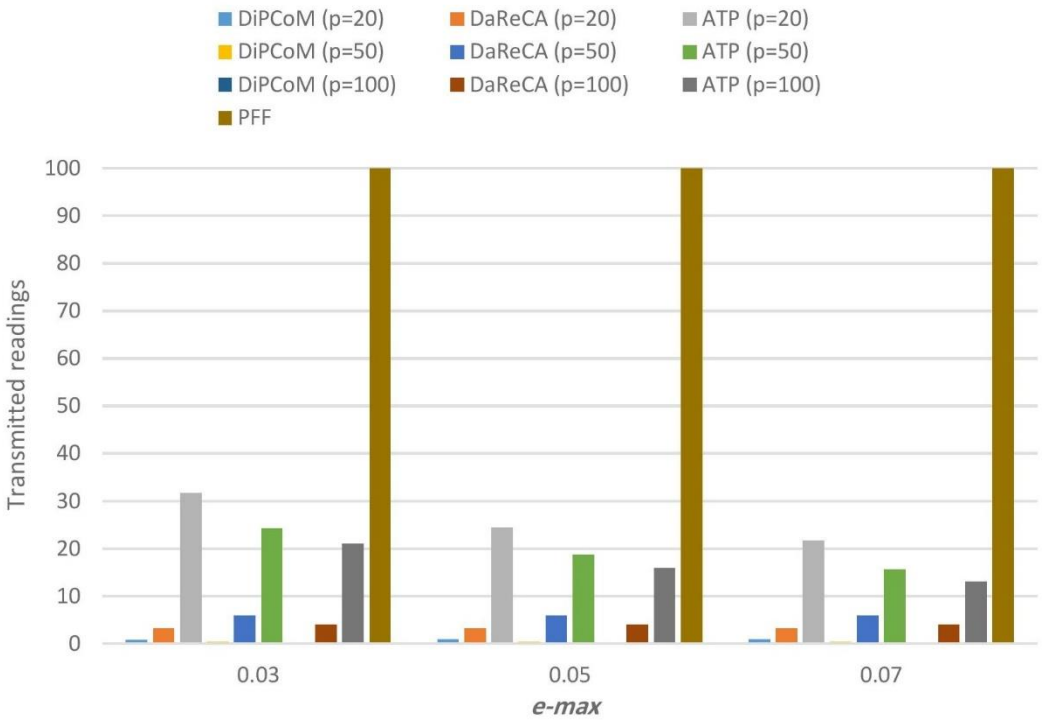


Figure 4.14: Transmitted Readings.

Table 4.14 shows the transmitted readings using different methods. In Figure 4.14, it can be seen that the ATP, DaReCA, DiPCoM, and PFF send data from 13.03% up to 31.68%, 3.2% up to 5.9%, 0.16% up to 0.88%, and 100%, respectively. Figure 4.14 shows how well the proposed DiPCoM reduces transmitted data when compared to other methods. These results demonstrate

the effectiveness of using the prediction and compression approaches together to minimize the size of data sent by the device.

Table 4.14: Transmitted Readings

e-max	DiPCoM (p=20)	DaReCA (p=20)	ATP (p=20)	DiPCoM (p=50)	DaReCA (p=50)	ATP (p=50)	DiPCoM (p=100)	DaReCA (p=100)	ATP (p=100)	PFF
0.03	0.82	3.2	31.68	0.33	5.9	24.31	0.17	4	20.96	100
0.05	0.86	3.2	24.41	0.34	5.9	18.76	0.16	4	15.9	100
0.07	0.88	3.2	21.69	0.35	5.9	15.58	0.16	4	13.03	100

Figure 4.15 shows how much energy the sensor node used when the DiPCoM method was used compared to the DaReCA, ATP, or PFF methods.

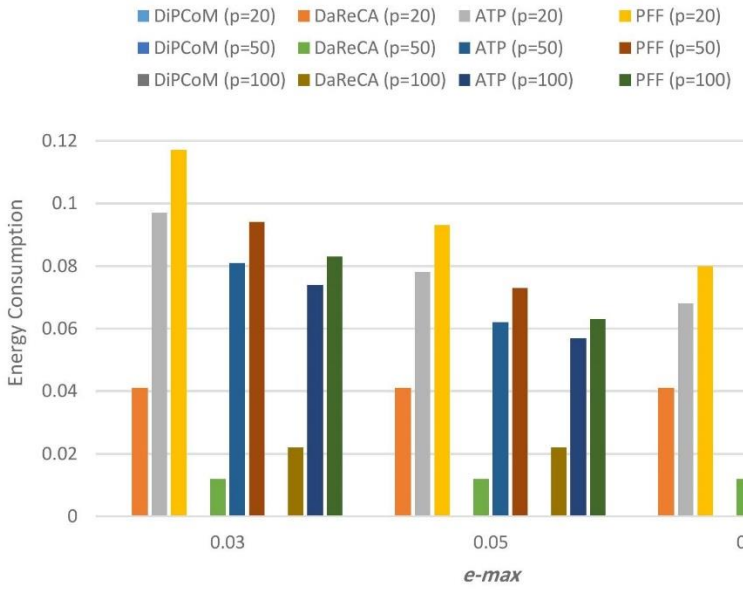


Figure 4.15: Energy Consumption

Table 4.15 shows the consumed energy by using different methods. The findings in Figure 4.15 show that the DiPCoM reduced the depleted power by the sensor device (in joules) from 0.000089 up to 0.000334, while the DaReCA, ATP, and PFF approach consumed from 0.012 up to 0.041, from 0.046 up to

0.097, and from 0.06 up to 0.117, respectively. Due to the implementation of an effective data reduction technique at the sensor devices, the suggested DiPCoM has good performance in lowering this amount of power at the sensor device. As a result, the device's power consumption is reduced.

Table 4.15: Energy Consumption

e-max	**DiPCoM (p=20)**	**DaReCA (p=20)**	**ATP (p=20)**	**PFF (p=20)**	**DiPCoM (p=50)**	**DaReCA (p=50)**	**ATP (p=50)**	**PFF (p=50)**	**DiPCoM (p=100)**	**DaReCA (p=100)**	**ATP (p=100)**	**PFF (p=100)**
0.03	0.000311118	0.041	0.097	0.117	0.00015932	0.012	0.081	0.094	0.000096	0.022	0.074	0.083
0.05	0.000327314	0.041	0.078	0.093	0.00016481	0.012	0.062	0.073	0.000090	0.022	0.057	0.063
0.07	0.000333627	0.041	0.068	0.08	0.000169421	0.012	0.052	0.06	0.000089	0.022	0.046	0.052

The next experiment looks at the effect of reducing the data at the sensor devices on the accuracy of the data that are reached at the gateway. To save power, it is important to decrease the size of data transferred to the gateway before sending them, but it is also essential to make sure that the data accuracy at the gateway is high enough. The data loss percentage serves as a measure for the accuracy of the data. The percentage of lost data is illustrated in Figure 4.15.

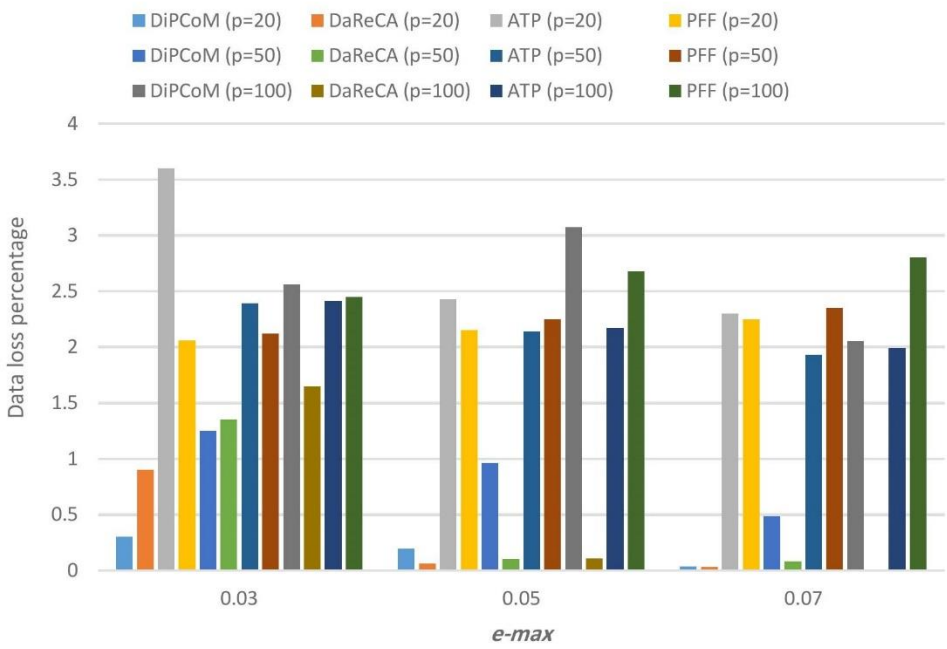


Figure 4.16: Data Loss Percentage

Table 4.16 shows the percentage of lost data readings by using different methods. In Figure 4.16, it can be seen that the DaReCA, ATP, and PFF lose data from 0.0045% to 1.65%, 1.93% to 3.6%, and 2.06% to 2.68%, respectively. On the other hand, the DiPCoM technique loses data from 0.04% up to 3.07%. While achieving a greater reduction of data at the sensor device, the DiPCoM technique frequently provided greater data accuracy when compared with existing approaches because it delivered a larger amount of data when executed at the sensor device, which actually reduced the lost data. The DaReCA approach can sometimes result in less data loss. As a result, the proposed DiPCoM could further reduce sensed data while preserving sensor node energy and maintaining an appropriate data quality level.

Table 4.16: Data Loss Percentage

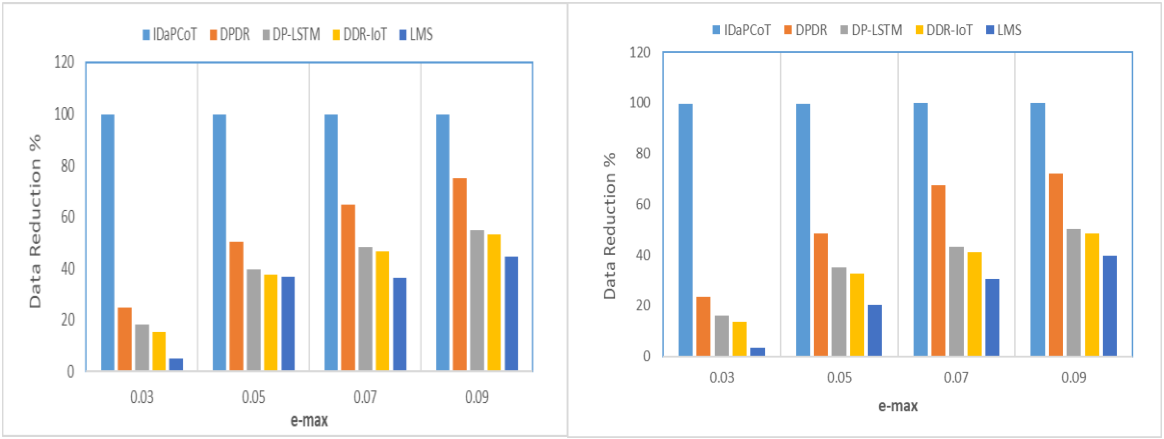
e-max	DiPCoM (p=20)	DaReCA (p=20)	ATP (p=20)	PFF (p=20)	DiPCoM (p=50)	DaReCA (p=50)	ATP (p=50)	PFF (p=50)	DiPCoM (p=100)	DaReCA (p=100)	ATP (p=100)	PFF (p=100)
0.03	0.3	0.9	3.6	2.06	1.247	1.35	2.39	2.12	2.559	1.65	2.41	2.45
0.05	0.196	0.06	2.43	2.15	0.961	0.1	2.14	2.25	3.073	0.107	2.17	2.68
0.07	0.035	0.03	2.3	2.25	0.484	0.08	1.93	2.35	2.053	0.0045	1.99	2.8

4.5 IDaPCoT Approach Performance Evaluation

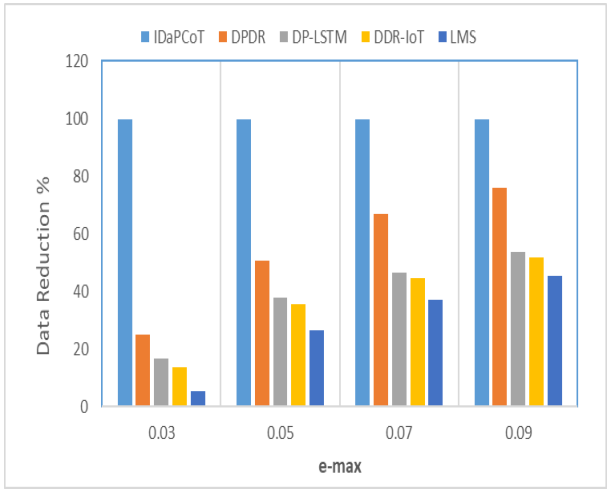
We propose an Integrated Data Prediction and Compression Techniques (IDaPCoT) for energy saving in IoT networks. The new contribution of this approach is using the AR prediction method to predict the data for the next period and suggest LZW as an effective compression technique to reduce the amount of data sent to the Gateway and keep the energy of the sensor nodes. The IDaPCoT applied the same reduction techniques that were used in DiPCoM.

4.5.1 Data Reduction

It can be seen from the results of the Figure 4.17 that the proposed IDaPCoT approach increased the percentage of reduction from 99.72% up to 99.76%, while the other methods introduced a percentage of reduction from 23.42% up to 75.75%, from 16.04% up to 54.99%, from 13.41% up to 53.39%, and from 3.33% up to 45.34% for DPDR, DP_LSTM, DDR-IoT, and LMS, respectively.



a- Sensor1 **b- Sensor2**



c- Sensor3

Figure 4.17: Data Reduction of Sensors 1, 2, and 3.

The proposed IDaPCoT approach achieves better performances compared to other approaches, and it reduces the sensed data efficiently by removing the redundant data before sending it to the gateway. Tables 4.17, 4.18, and 4.19 illustrate the values of sensors devices 1, 2, and 3 respectively.

Table 4.17: Data Reduction (Sensor Device 1)

e-max	Data Reduction %				
	IDaPCoT	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.71875	24.92	18.4	15.61	4.92
0.05	99.74375	50.34	39.61	37.55	36.92
0.07	99.753125	64.67	48.39	46.54	36.49
0.09	99.756875	75.14	54.99	53.39	44.71

Table 4.18: Data Reduction (Sensor Device 2)

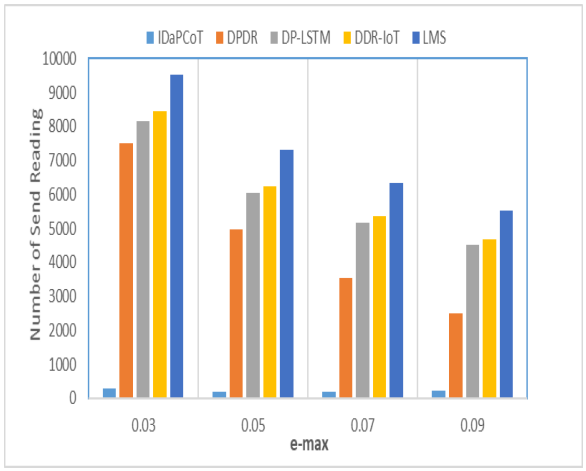
e-max	Data Reduction %				
	IDaPCoT	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.709375	23.42	16.04	13.41	3.33
0.05	99.74375	48.4	34.98	32.71	20.31
0.07	99.75875	67.34	43.26	41.08	30.61
0.09	99.75875	72.16	50.27	48.57	39.77

Table 4.19: Data Reduction (Sensor Device 3)

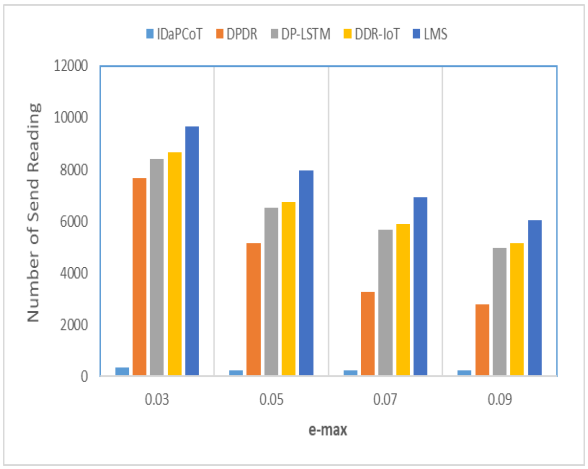
e-max	Data Reduction %				
	IDaPCoT	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.71875	24.99	16.72	13.51	5.27
0.05	99.7425	50.47	37.78	35.48	26.31
0.07	99.75	66.95	46.52	44.55	37.07
0.09	99.7525	75.75	53.75	51.91	45.34

4.5.2 Number of Send Reading

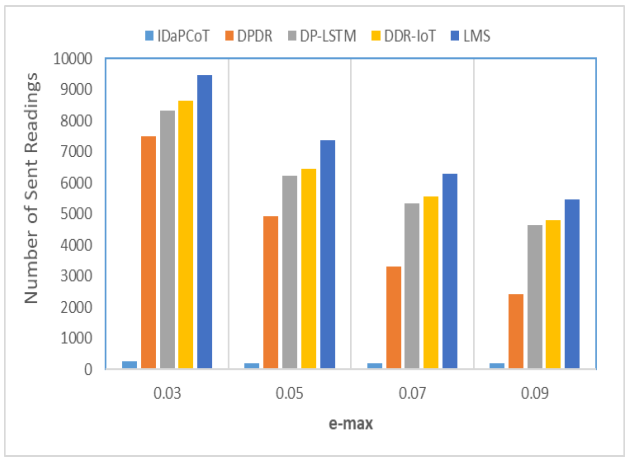
It can be observed from Figure 4.18 that IDaPCoT decreases the amount of data transmitted to the Gateway from 96.54% up to 98.07%, 23.42% up to 75.75%, 13.41% up to 53.4%, and 3.3% up to 45.3%, compared to the DPDR, DP-LSTM, DDR-IoT, and LMS, respectively. As a result, the duplicate data is effectively removed by the suggested IDaPCoT technique before being sent to the Gateway.



a- Sensor1



b- Sensor2



c- Sensor3

Figure 4.18: Number of Send Readings of Sensors 1, 2, and 3.

As a result, the duplicate data is effectively removed by the suggested IDaPCoT technique before being sent to the Gateway. Tables 4.20, 4.21, and 4.22 illustrate the values of sensors devices 1, 2, and 3 respectively.

Table 4.20: Number of Send Reading (Sensor Device 1)

e-max	Number of Send Reading				
	IDaPCoT	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	296	7508	8160	8439	9508
0.05	203	4966	6039	6245	7308
0.07	202	3533	5161	5346	6351
0.09	209	2486	4501	4661	5529

Table 4.21: Number of Send Reading (Sensor Device 2)

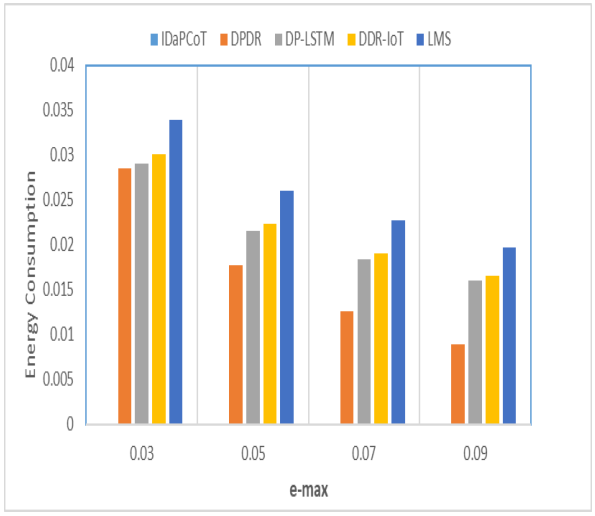
e-max	Number of Send Reading				
	IDaPCoT	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	346	7658	8396	8659	9667
0.05	220	5160	6502	6729	7969
0.07	224	3266	5674	5892	6939
0.09	226	2784	4973	5143	6023

Table 4.22: Number of Send Reading (Sensor Device 3)

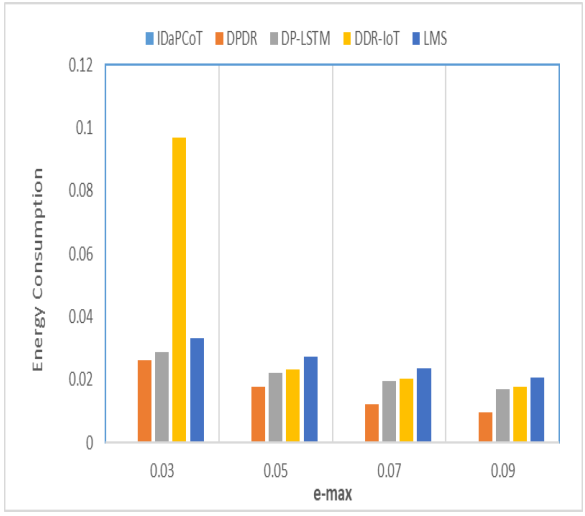
e-max	Number of Send Reading				
	IDaPCoT	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	251	7501	8328	8649	9473
0.05	193	4926	6222	6452	7369
0.07	208	3305	5348	5545	6293
0.09	205	2425	4625	4809	5466

4.5.3 Energy Consumption

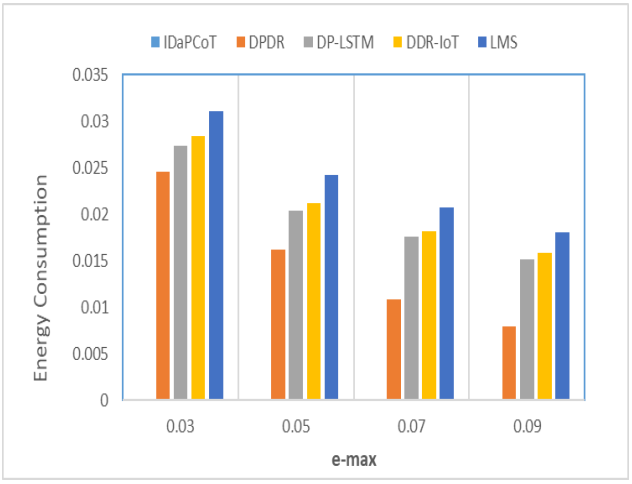
It can be observed from Figure 4.19 that IDaPCoT decreases the amount of data transmitted to the Gateway from 99.999981% up to 99.9999894%, 99.9714% up to 99.992%, 99.9709% up to 99.9848%, 99.903% up to 99.9842%, and 99.9661% up to 99.982% compared to the DPDR, DP-LSTM, DDR-IoT, and LMS, respectively.



a- Sensor1



b- Sensor2



c- Sensor3

Figure 4.19: Energy Consumption of Sensors 1, 2, and 3.

As a result, the duplicate data is effectively removed by the suggested IDaPCoT technique before being sent to the Gateway. Tables 4.23, 4.24, and 4.25 illustrate the values of sensors devices 1, 2, and 3 respectively.

Table 4.23: Energy Consumption (Sensor Device 1)

e-max	Energy Consumption				
	IDaPCoT	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	1.62504E-05	0.0286	0.0291	0.0301	0.0339
0.05	1.11447E-05	0.0177	0.0215	0.0223	0.0261
0.07	1.10898E-05	0.0126	0.0184	0.0191	0.0227
0.09	1.14741E-05	0.0089	0.0161	0.0166	0.0197

Table 4.24: Energy Consumption (Sensor Device 2)

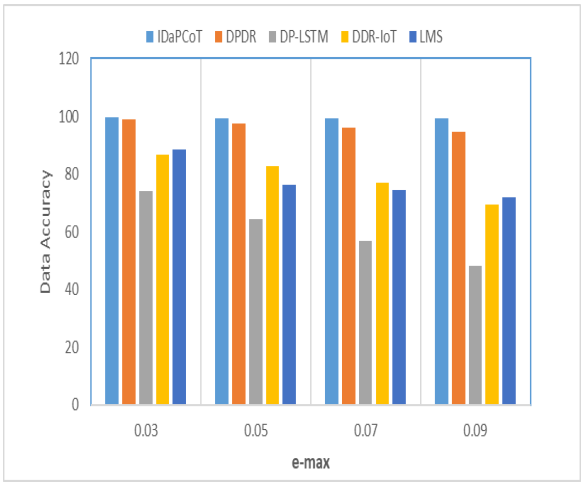
e-max	Energy Consumption				
	IDaPCoT	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	1.89954E-05	0.0263	0.0288	0.097	0.0332
0.05	0.000012078	0.0177	0.0223	0.0231	0.0273
0.07	1.22976E-05	0.0122	0.0195	0.0202	0.0238
0.09	1.24074E-05	0.0096	0.0171	0.0177	0.0207

Table 4.25: Energy Consumption (Sensor Device 3)

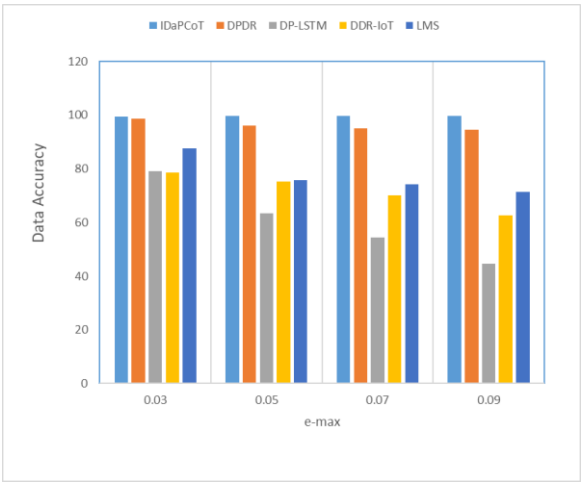
e-max	Energy Consumption				
	IDaPCoT	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	1.37799E-05	0.0246	0.0274	0.0284	0.0311
0.05	1.05957E-05	0.0162	0.0204	0.0212	0.0242
0.07	1.14192E-05	0.0109	0.0176	0.0182	0.0207
0.09	1.12545E-05	0.008	0.0152	0.0158	0.018

4.5.4 Data Accuracy

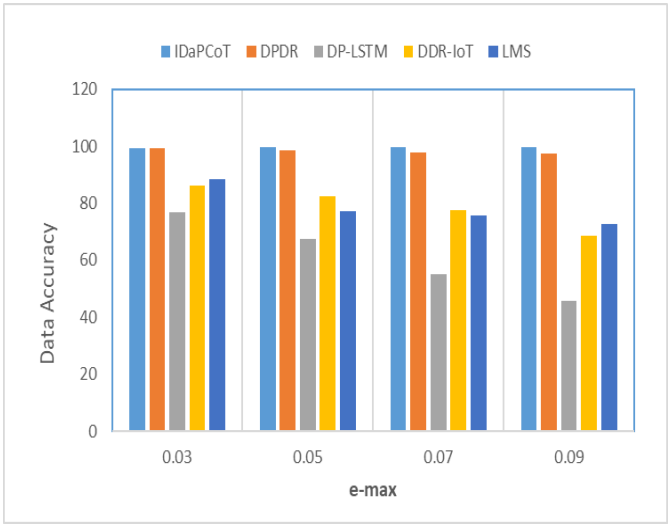
It can be seen from the results of Figure 4.20 that the proposed IDaPCoT approach introduces a suitable data accuracy compared with other methods introduced from 0.7% up to 4.7%, from 20.6% up to 53.5%, from 13% up to 36.6%, and from 11.4% up to 27.7% for DPDR, DP_LSTM, DDR-IoT, and LMS, respectively.



a- Sensor1



b- Sensor2



c- Sensor3

Figure 4.20: Data Accuracy of Sensors 1, 2, and 3.

Therefore, the proposed IDaPCoT will ensure high data reduction while appropriate data accuracy is preserved at the Gateway. Tables 4.26, 4.27, and 4.28 illustrate the values of sensors devices 1, 2, and 3 respectively.

Table 4.26: Data Accuracy (Sensor Device 1)

e-max	Data Accuracy				
	IDaPCoT	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.65874237	98.8	74.12	86.83	88.42
0.05	99.45875623	97.66	64.23	82.77	76.39
0.07	99.29854632	96.07	56.81	77.12	74.51
0.09	99.24563896	94.56	48.26	69.41	71.92

Table 4.27: Data Accuracy (Sensor Device 2)

e-max	Data Accuracy				
	IDaPCoT	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.44922547	98.71	79.18	78.7	87.6
0.05	99.67783505	96.23	63.45	75.22	75.67
0.07	99.72752044	95.16	54.38	70.24	74.16
0.09	99.7235023	94.5	44.71	62.6	71.54

Table 4.28: Data Accuracy (Sensor Device 3)

e-max	Data Accuracy				
	IDaPCoT	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.39802548	99.06	76.79	86.13	88.37
0.05	99.58848965	98.52	67.31	82.36	77.09
0.07	99.72469752	97.85	55.09	77.43	75.72
0.09	99.79568742	97.18	45.75	68.56	72.49

4.5.5 Evaluation of IDaPCoT for Further Results

In the next experiments, the IDaPCoT approach use temperature readings during the simulation. Like the first approach, we used the same previous comparisons and also with different sizes of the periods, such as T= 20, 50, and 100 readings per period. Moreover, the e-max uses different values like 0.03, 0.05, and 0.07. In this approach, we used the same measures used in the first approach, such as transmitted readings, energy consumption, and Data loss percentage.

4.5.5.1 Transmitted Readings

In Figure 4.21, it can be seen that the IDaPCoT, DaReCA, ATP, and PFF send data from 0.162% up to 0.845%, 3.2% up to 5.9%, 13.03% up to 31.68%, and 100%, respectively.

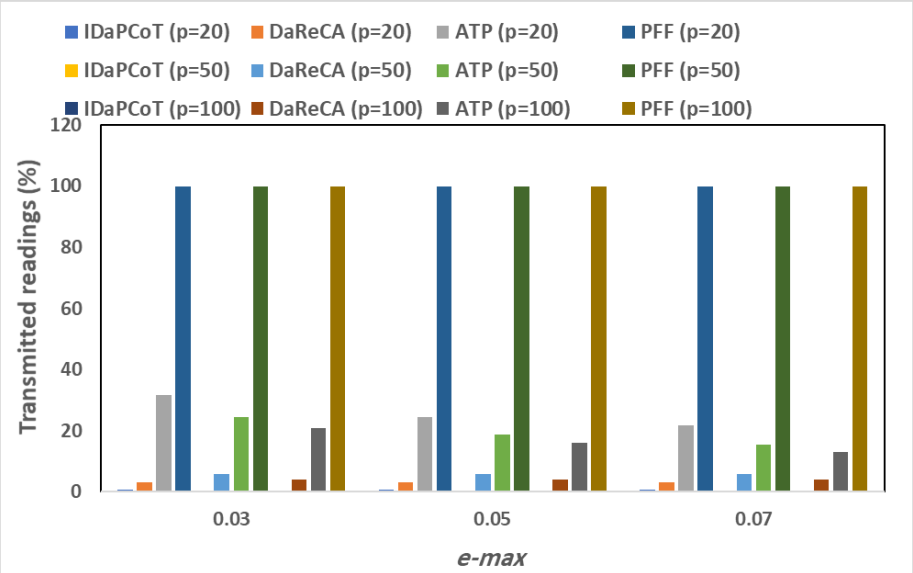


Figure 4.21: Transmitted Readings

The Figure 4.21 shows how well the proposed IDaPCoT reduces transmitted data when compared to other methods. These results demonstrate the effectiveness of using the prediction and compression approaches together to minimize the size of data that the device sends. Table 2.29 illustrate the values of Transmitted reading of the approach IDaPCoT.

Table 4.29: Transmitted Readings

e-max	IDaPCoT (p=20)	DaReCA (p=20)	ATP (p=20)	PFF (p=20)	IDaPCoT (p=50)	DaReCA (p=50)	ATP (p=50)	PFF (p=50)	IDaPCoT (p=100)	DaReCA (p=100)	ATP (p=100)	PFF (p=100)
0.03	0.83	3.2	31.68	100	0.33	5.9	24.31	100	0.17	4	20.96	100
0.05	0.84	3.2	24.41	100	0.35	5.9	18.76	100	0.16	4	15.9	100
0.07	0.85	3.2	21.69	100	0.35	5.9	15.58	100	0.16	4	13.03	100

4.5.5.2 Energy Consumption

The Figure 4.22 shows how much energy the sensor node used when the IDaPCoT method was used compared to the DaReCA, ATP, or PFF methods. The findings in Figure 4.22 show that the IDaPCoT reduced the depleted power by the sensor device from 99.99965% up to 99.99966%, from 99.99980% up to 99.99982%, and from 99.9999% up to 99.99991% compared with the DaReCA, ATP, and PFF approaches.

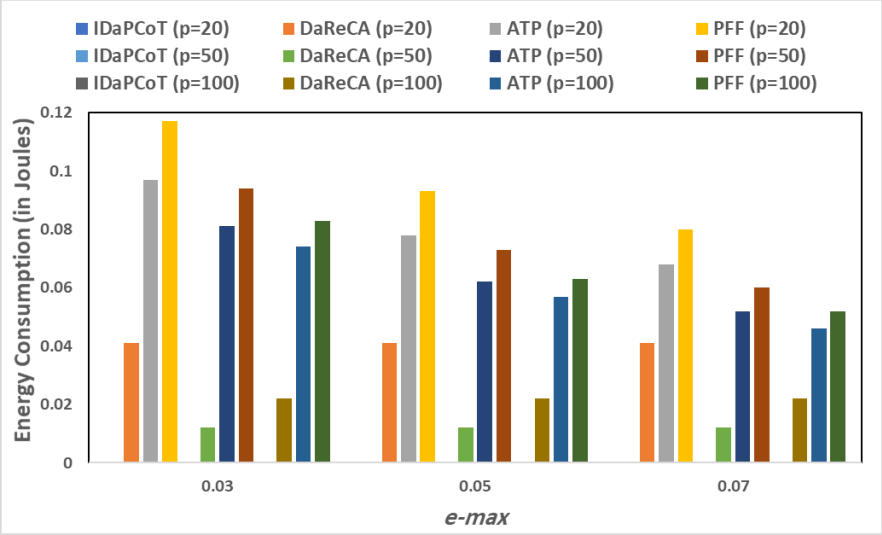


Figure 4.22: Energy Consumption

Due to the implementation of an effective data reduction technique at the sensor devices, the suggested IDaPCoT has good performance in lowering this amount of power at the sensor device. As a result, the device's power consumption is reduced. Table 4.30 illustrate the values of Energy Consumption of the approach IDaPCoT.

Table 4.30: Energy Consumption

e-max	IDaPCoT (p=20)	DaReCA (p=20)	ATP (p=20)	PFF (p=20)	IDaPCoT (p=50)	DaReCA (p=50)	ATP (p=50)	PFF (p=50)	IDaPCoT (p=100)	DaReCA (p=100)	ATP (p=100)	PFF (p=100)
0.03	0.000339	0.041	0.097	0.117	0.000178	0.012	0.081	0.094	0.000095	0.022	0.074	0.083
0.05	0.000346	0.041	0.078	0.093	0.000188	0.012	0.062	0.073	0.00009	0.022	0.057	0.063
0.07	0.00035	0.041	0.068	0.08	0.00019	0.012	0.052	0.06	0.000089	0.022	0.046	0.052

4.5.5.3 Data Lose Percentage

In Figure 4.23, it can be seen that the DaReCA, ATP, and PFF lose data from 0.0045% to 1.65%, 1.93% to 3.6%, and 2.06% to 2.68% respectively. On the other hand, the IDaPCoT technique loses data from 0.034% up to 3.07%.

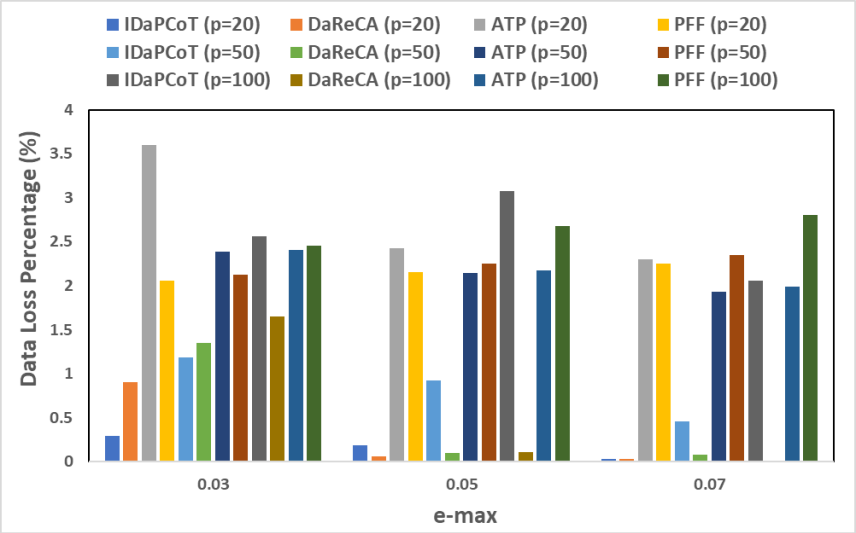


Figure 4.23: Data Lose Percentage

The IDaPCoT method introduces lower data loss in some cases because it sends a larger volume of data when it is implemented at the sensor node which participates in reducing the lost data. Table 4.31 illustrate the values of the Data Lose Percentage of the IDaPCoT approach.

Table 4.31: Data Lose Percentage

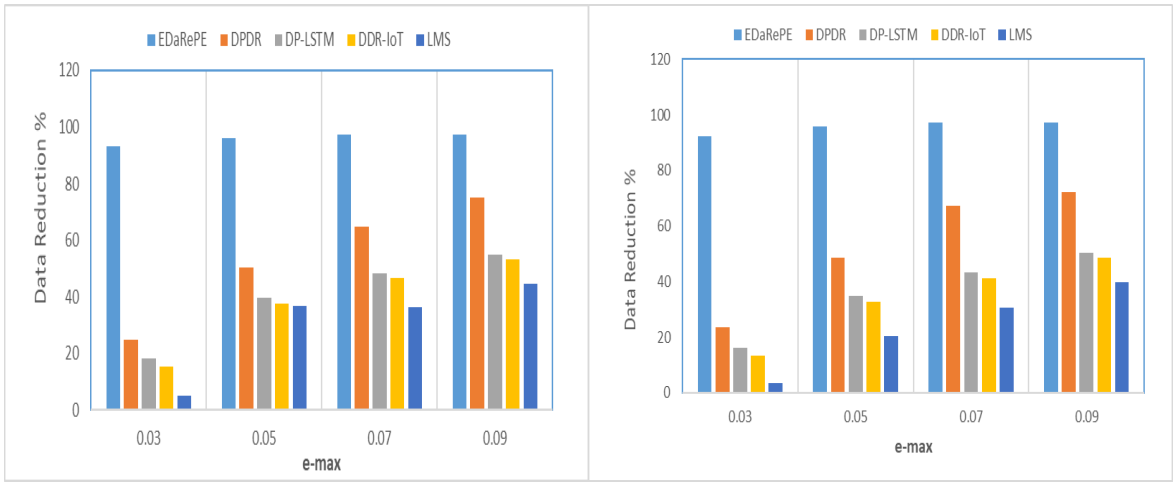
e-max	IDaPCoT (p=20)	DaReCA (p=20)	ATP (p=20)	PFF (p=20)	IDaPCoT (p=50)	DaReCA (p=50)	ATP (p=50)	PFF (p=50)	IDaPCoT (p=100)	DaReCA (p=100)	ATP (p=100)	PFF (p=100)
0.03	0.29	0.9	3.6	2.06	1.18	1.35	2.39	2.12	2.56	1.65	2.41	2.45
0.05	0.19	0.06	2.43	2.15	0.92	0.1	2.14	2.25	3.075	0.107	2.17	2.68
0.07	0.03	0.03	2.3	2.25	0.47	0.08	1.93	2.35	2.054	0.0045	1.99	2.8

4.6 EDaRePE Approach Performance Evaluation

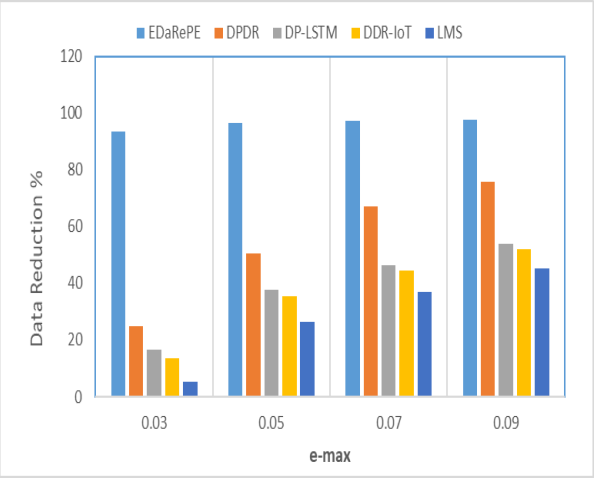
We propose an Energy-efficient Data Reduction based Prediction and Encoding (EDaRePE) in IoT networks. This approach's new contribution is its use of the ARIMA prediction approach to anticipate future data and its recommendation of Huffman as an efficient compression technique to lessen the transmitted data between the sensor nodes and the Gateway. As in DiPCoM, the EDaRePE made use of reduction methods.

4.6.1 Data Reduction

The Figures 4.24 illustrate that the EDaRePE enhanced the ratio of reduction from 92.42% to 97.58%, 44.8% to 83.7%, 24% to 76.3%, 46.4% to 86.3%, and 54.4% to 96.4% compared with DP-LSTM, DPDR, DDR-IoT, and LMS, respectively.



a- Sensor1 **b- Sensor2**



c- Sensor3

Figure 4.24: Data Reduction of Sensors 1, 2, and 3.

The suggested EDaRePE method improves performance and lowers the size of data before transferring it to the Gateway by getting rid of duplicates. Tables 4.32, 4.33, and 4.34 show the data reduction values of sensors devices 1, 2, and 3 respectively.

Table 4.32: Data Reduction (Sensor Device 1)

e-max	Data Reduction %				
	EDaRePE	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	93.13625	24.92	18.4	15.61	4.92
0.05	96.191875	50.34	39.61	37.55	36.92
0.07	97.146875	64.67	48.39	46.54	36.49
0.09	97.305	75.14	54.99	53.39	44.71

Table 4.33: Data Reduction (Sensor Device 2)

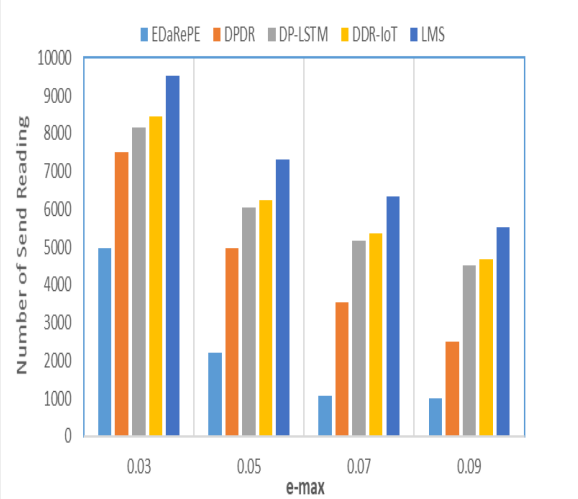
e-max	Data Reduction %				
	EDaRePE	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	92.42045455	23.42	16.04	13.41	3.33
0.05	95.95391414	48.4	34.98	32.71	20.31
0.07	97.12752525	67.34	43.26	41.08	30.61
0.09	97.16666667	72.16	50.27	48.57	39.77

Table 4.34: Data Reduction (Sensor Device 3)

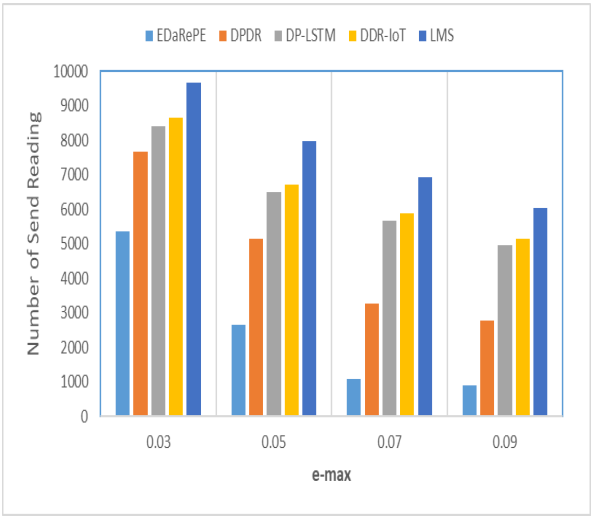
e-max	Data Reduction %				
	EDaRePE	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	93.44125	24.99	16.72	13.51	5.27
0.05	96.450625	50.47	37.78	35.48	26.31
0.07	97.431875	66.95	46.52	44.55	37.07
0.09	97.579375	75.75	53.75	51.91	45.34

4.6.2 Number of Send Reading

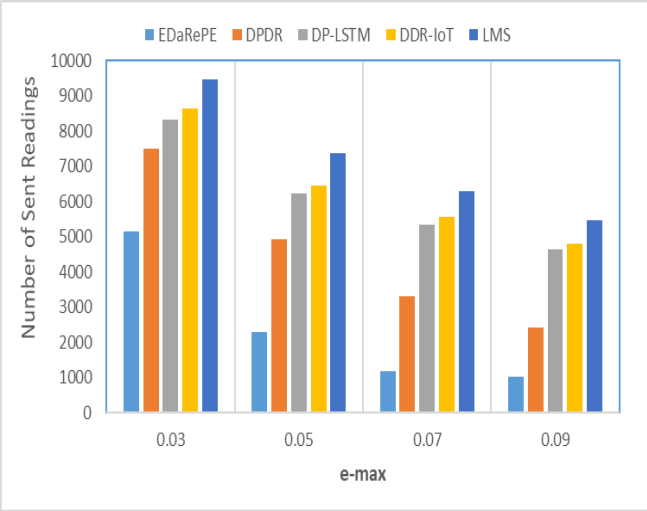
It can be observed from the Figure 4.25 that EDaRePE decreases the amount of data transmitted to the Gateway from 84.3% up to 93.9%, 91.8% up to 94.6 %, and 93% up to 95.2%, compared to the DP-LSTM, DPDR, DDR-IoT, and LMS, respectively.



a- Sensor1



b- Sensor2



c- Sensor3

Figure 4.25: Number of Send Readings of Sensors 1, 2, and 3.

As a result, the duplicate data is effectively removed by the suggested EDaRePE technique before being sent to the Gateway. Tables 4.35, 4.36, and 4.37 show the values of numbers of send readings of sensors devices 1, 2, and 3 respectively.

Table 4.35: Number of Send Readings (Sensor Device 1)

e-max	Number of Send Reading				
	EDaRePE	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	4984	7508	8160	8439	9508
0.05	2215	4966	6039	6245	7308
0.07	1054	3533	5161	5346	6351
0.09	998	2486	4501	4661	5529

Table 4.36: Number of Send Readings (Sensor Device 2)

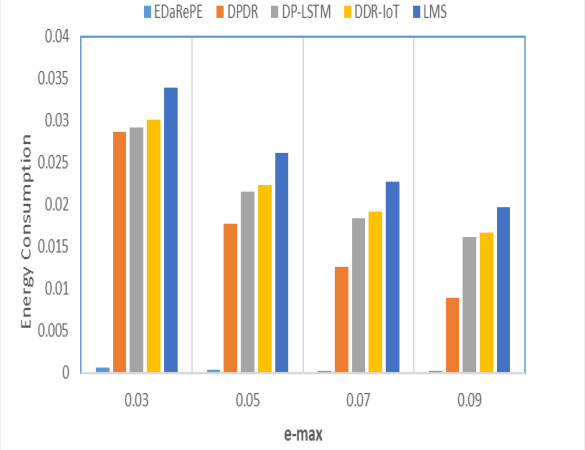
e-max	Number of Send Reading				
	EDaRePE	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	5365	7658	8396	8659	9667
0.05	2652	5160	6502	6729	7969
0.07	1078	3266	5674	5892	6939
0.09	891	2784	4973	5143	6023

Table 4.37: Number of Send Readings (Sensor Device 3)

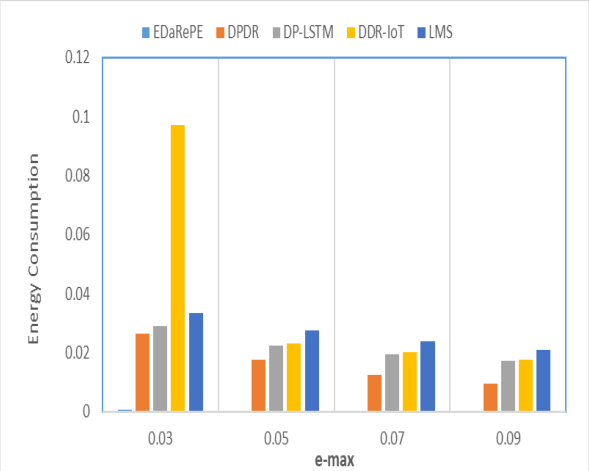
e-max	Number of Send Reading				
	EDaRePE	**DPDR**	**DP-LSTM**	**DDR-IoT**	**LMS**
0.03	5143	7501	8328	8649	9473
0.05	2276	4926	6222	6452	7369
0.07	1183	3305	5348	5545	6293
0.09	1032	2425	4625	4809	5466

4.6.3 Energy Consumption

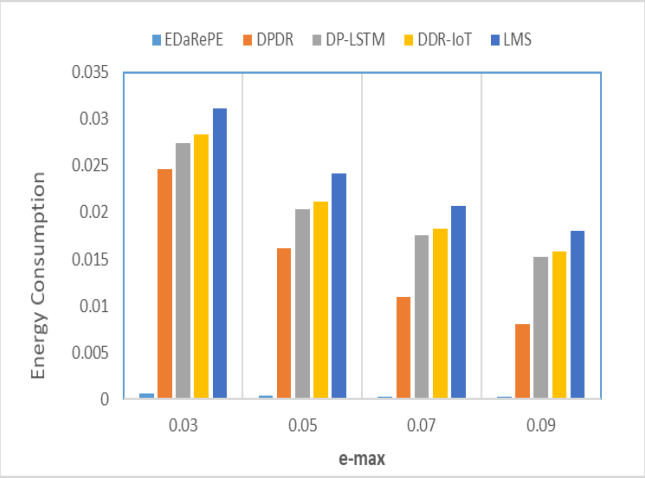
The findings in the Figure 4.26 that the EDaRePE approach reduced the consumed energy from 99.735% up to 99.911%, from 99.861% up to 99.912%, and from 99.915% up to 99.978%, 99.882 up to 99.925 compared with the DPDR, DP-LSTM, DDR-IoT, and LMS, respectively. The findings indicate that the EDaRePE preserves more power than other approaches since it transmits the least amount of data to the Gateway.



a- Sensor1



b- Sensor2



c- Sensor3

Figure 4.26: Energy Consumption of Sensors 1, 2, and 3.

The findings indicate that the EDaRePE preserves more power than other approaches since it transmits the least amount of data to the Gateway. Tables 4.38, 4.39, and 4.40 show the values of energy consumption of sensors devices 1, 2, and 3 respectively.

Table 4.38: Energy Consumption (Sensor Device 1)

e-max	Energy Consumption				
	EDaRePE	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	0.000602912	0.0286	0.0291	0.0301	0.0339
0.05	0.000334506	0.0177	0.0215	0.0223	0.0261
0.07	0.000250619	0.0126	0.0184	0.0191	0.0227
0.09	0.000236729	0.0089	0.0161	0.0166	0.0197

Table 4.39: Energy Consumption (Sensor Device 2)

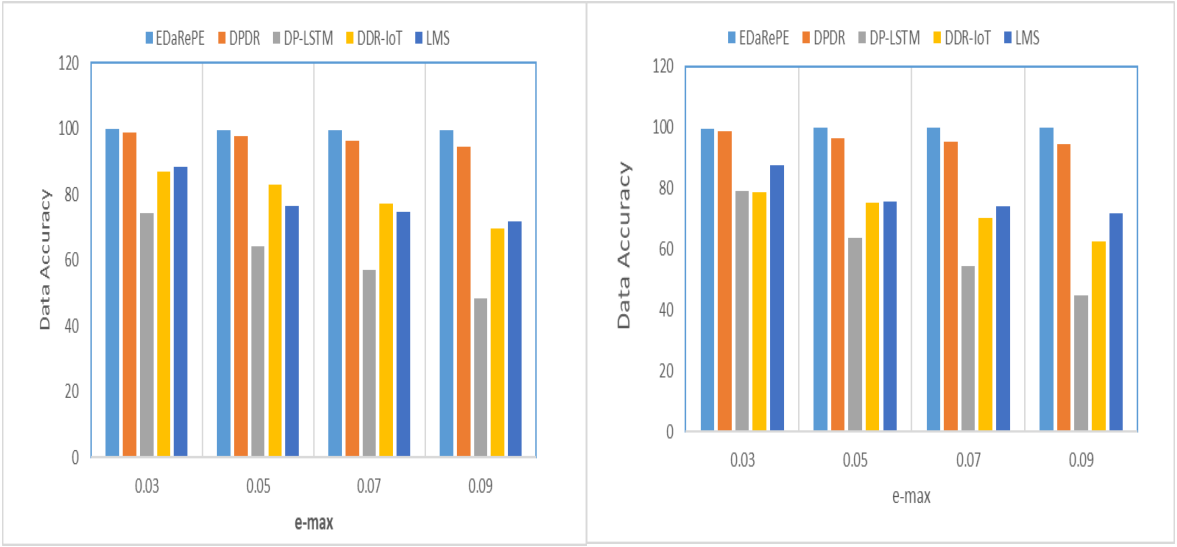
e-max	Energy Consumption				
	EDaRePE	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	0.000659129	0.0263	0.0288	0.097	0.0332
0.05	0.000351854	0.0177	0.0223	0.0231	0.0273
0.07	0.000249795	0.0122	0.0195	0.0202	0.0238
0.09	0.000246391	0.0096	0.0171	0.0177	0.0207

Table 4.40: Energy Consumption (Sensor Device 3)

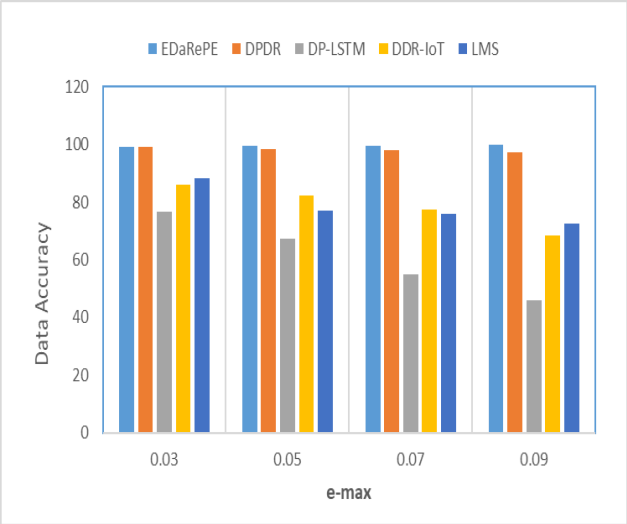
e-max	Energy Consumption				
	EDaRePE	**DPDR**	**DP-LSTM**	**DDR-IoT**	**LMS**
0.03	0.000576121	0.0246	0.0274	0.0284	0.0311
0.05	0.000311777	0.0162	0.0204	0.0212	0.0242
0.07	0.000225584	0.0109	0.0176	0.0182	0.0207
0.09	0.000212628	0.008	0.0152	0.0158	0.018

4.6.4 Data Accuracy

It can be seen from the results of the Figure 4.27 that the proposed EDaRePE approach introduces a suitable data accuracy compared with other methods introduced from 0.7% up to 4.8%, from 20.6% up to 53.5%, from 13% up to 36.6%, and from 11.4% up to 27.7% for DPDR, DP_LSTM, DDR-IoT, and LMS, respectively.



a- Sensor1 **b- Sensor2**



c- Sensor3

Figure 4.27: Data Accuracy of Sensors 1, 2, and 3.

Therefore, the proposed EDaRePE will ensure high data reduction while appropriate data accuracy is preserved at the gateway. Tables 4.41, 4.42, and 4.43 show the values of data accuracy of sensors devices 1, 2, and 3 respectively.

Table 4.41: Data Accuracy (Sensor Device 1)

e-max	Data Accuracy				
	EDaRePE	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.73723724	98.8	74.12	86.83	88.42
0.05	99.46055293	97.66	64.23	82.77	76.39
0.07	99.37219731	96.07	56.81	77.12	74.51
0.09	99.33460076	94.56	48.26	69.41	71.92

Table 4.42: Data Accuracy (Sensor Device 2)

e-max	Data Accuracy				
	EDaRePE	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.44770452	98.71	79.18	78.7	87.6
0.05	99.67679379	96.23	63.45	75.22	75.67
0.07	99.72652689	95.16	54.38	70.24	74.16
0.09	99.72273567	94.5	44.71	62.6	71.54

Table 4.43: Data Accuracy (Sensor Device 3)

e-max	Data Accuracy				
	EDaRePE	DPDR	DP-LSTM	DDR-IoT	LMS
0.03	99.25490196	99.06	76.79	86.13	88.37
0.05	99.56647399	98.52	67.31	82.36	77.09
0.07	99.6007984	97.85	55.09	77.43	75.72
0.09	99.78835979	97.18	45.75	68.56	72.49

4.6.5 Evaluation of EDaRePE for Further Results

In the next experiments, the EDaRePE approach use temperature readings during the simulation. Like the first approach, we used the same previous comparisons and also with different sizes of the periods, such as T= 20, 50, and 100 readings per period. Moreover, the e-max uses different values like 0.03, 0.05, and 0.07.

4.6.5.1 Transmitted Readings

It can be observed from the results presented in Figure 4.28 that the EDaRePE, DaReCA, ATP, and PFF approaches sent data from 1.65% up to 6.15%, from 3.2% up to 5.9%, from 13.03% up to 31.68%, and 100% respectively.

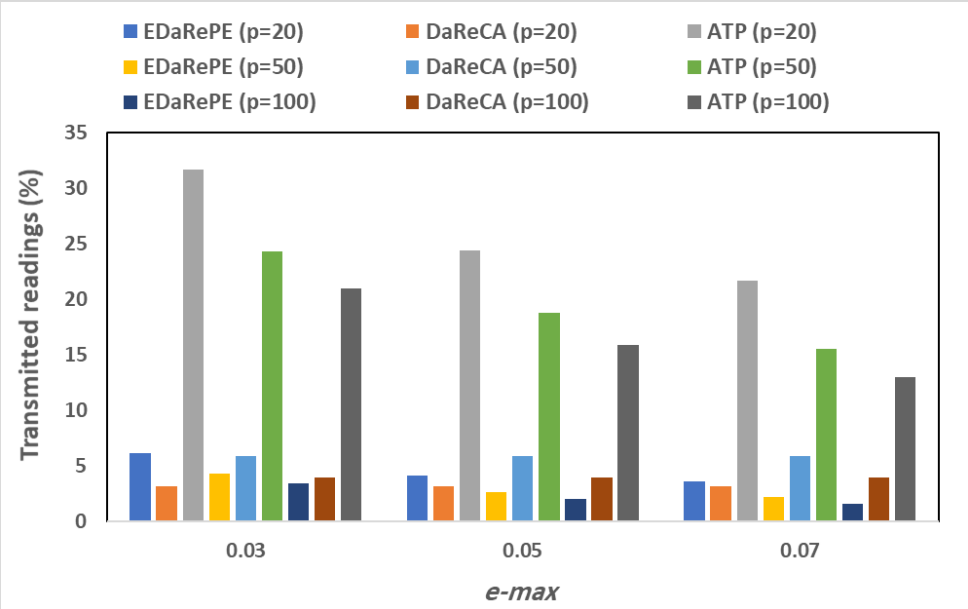


Figure 4.28: Transmitted Readings

It can be seen from the Figure 4.28 that the proposed EDaRePE approach reduced the transmitted data efficiently compared with other approaches. These findings ensure the efficiency of combining the prediction and compression methods in decreasing the volume of transmitted data by the sensor node. Table 4.44 illustrate the values of transmitted readings of the EDaRePE approach.

Table 4.44: Transmitted Readings

e-max	EDaRePE (p=20)	DaReCA (p=20)	ATP (p=20)	EDaRePE (p=50)	DaReCA (p=50)	ATP (p=50)	EDaRePE (p=100)	DaReCA (p=100)	ATP (p=100)
0.03	6.15	3.2	31.68	4.34	5.9	24.31	3.46	4	20.96
0.05	4.19	3.2	24.41	2.67	5.9	18.76	2.05	4	15.9
0.07	3.62	3.2	21.69	2.21	5.9	15.58	1.65	4	13.03

4.6.5.2 Energy Consumption

It can be seen from the results in Figure 4.29 that the proposed EDaRePE approach lowered the consumed energy by the sensor node from 99.9979% up to 99.9987%, from 99.998% up to 99.9989%, and 99.9981% up to 99.9991% compared with DaReCA, ATP, and PFF methods.

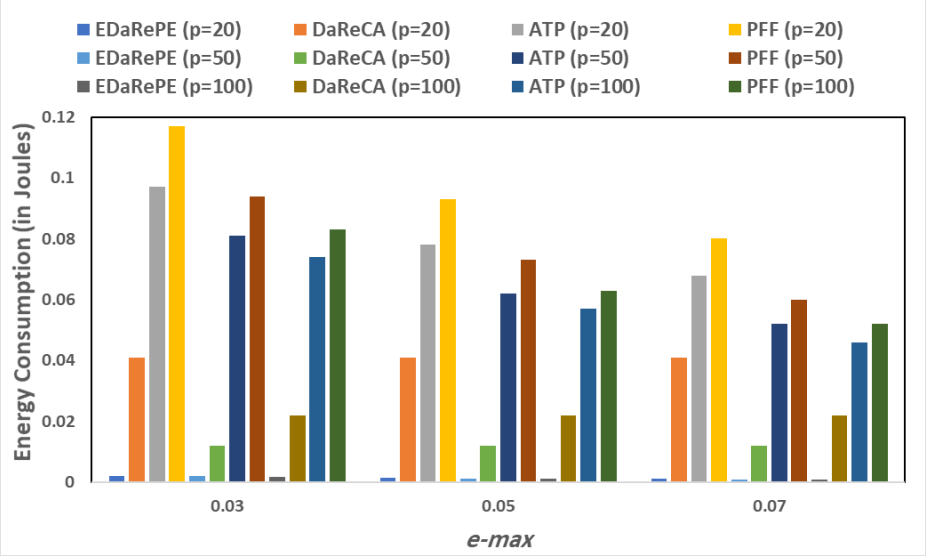


Figure 4.29: Energy Consumption

The high performance of the suggested EDaRePE in reducing this volume of energy is due to its use of an efficient data reduction method that reduces the sensed data before sending it by the sensor node. Table 4.45 shows the values of energy consumption of the EDaRePE approach.

Table 4.45: Energy Consumption

e-max	EDaRePE (p=20)	DaReCA (p=20)	ATP (p=20)	PFF (p=20)	EDaRePE (p=50)	DaReCA (p=50)	ATP (p=50)	PFF (p=50)	EDaRePE (p=100)	DaReCA (p=100)	ATP (p=100)	PFF (p=100)
0.03	0.0021	0.041	0.097	0.117	0.002	0.012	0.081	0.094	0.0019	0.022	0.074	0.083
0.05	0.0014	0.041	0.078	0.093	0.0012	0.012	0.062	0.073	0.001	0.022	0.057	0.063
0.07	0.0012	0.041	0.068	0.08	0.001	0.012	0.052	0.06	0.0009	0.022	0.046	0.052

4.6.5.3 Data Lose Percentage

It can be noted in Figure 4.30 that the EDaRePE approach decreased the lost data from 0.038% up to 3.108%, while the DaReCA, ATP, and PFF reduced the lost data from 0.03% up to 1.65%, from 1.93% up to 3.6% and from 2.06% up to 2.68% respectively.

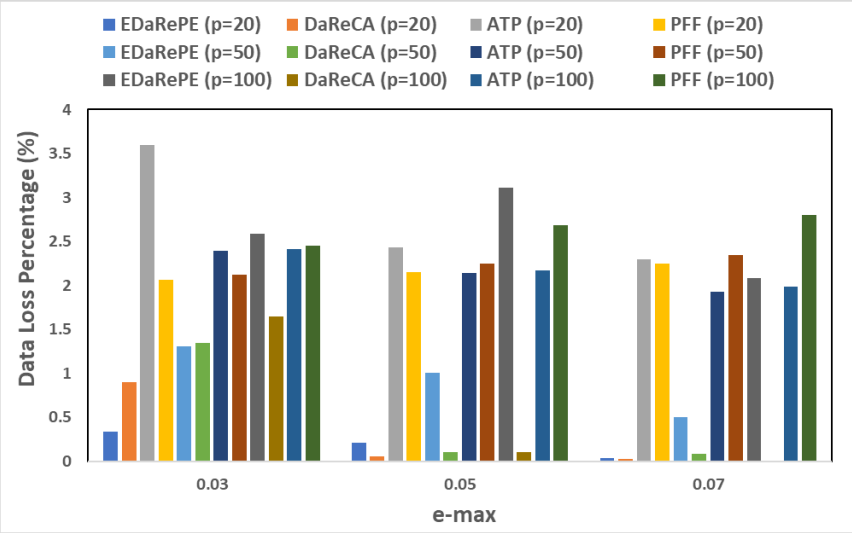


Figure 4.30: Data Lose Percentage

The proposed EDaRePE approach introduced better data accuracy in most cases compared with other methods in spite of achieving higher data reduction at the sensor node. The DaReCA method introduces lower data loss in some cases because it sends a larger volume of data when it is implemented at the sensor node which participates in reducing the lost data. Hence, the proposed EDaRePE could further decrease the sensed data and keep the energy of sensor nodes while preserving a suitable level of data quality. Table 4.46 illustrate the values of data lose percentage.

Table 4.46: Data Lose Percentage

e-max	EDaRePE (p=20)	DaReCA (p=20)	ATP (p=20)	PFF (p=20)	EDaRePE (p=50)	DaReCA (p=50)	ATP (p=50)	PFF (p=50)	EDaRePE (p=100)	DaReCA (p=100)	ATP (p=100)	PFF (p=100)
0.03	0.34	0.9	3.6	2.06	1.305	1.35	2.39	2.12	2.59	1.65	2.41	2.45
0.05	0.21	0.06	2.43	2.15	1.003	0.1	2.14	2.25	3.11	0.107	2.17	2.68
0.07	0.04	0.03	2.3	2.25	0.506	0.08	1.93	2.35	2.08	0.005	1.99	2.8

4.7 Comparison Between Four Approaches

1. Through the experiments that we made in the four approaches, we can see that the DiPCoM and the IDaPCoT approaches are superior to other approaches in measurements of Data Accuracy, Energy Consumption, Transmitted Data, and Data Loss Percentage, as these approaches recorded lower rates.
2. The Data Accuracy in the DEDaR and the EDaRePE approaches may be better, but we see that the rates of Energy Consumption and Transmitted Data are high, and we also see that the Data Loss Percentage is also high.
3. Also, we can note that the IDaPCoT approach is better than the DiPCoM approach in the above measurements.
4. In a conclusion, we can say that the IDaPCoT approach is the best among the other approaches.

It can be seen from the results in Figure 4.31, 4.32, and 4.33 that the proposed DiPCoM approach is superior to other approaches. It use temperature readings during the simulation. It uses different sizes per period such as T= 20, 50, and 100 readings per period. Moreover, the e-max uses different values like 0.03, 0.05, and 0.07.

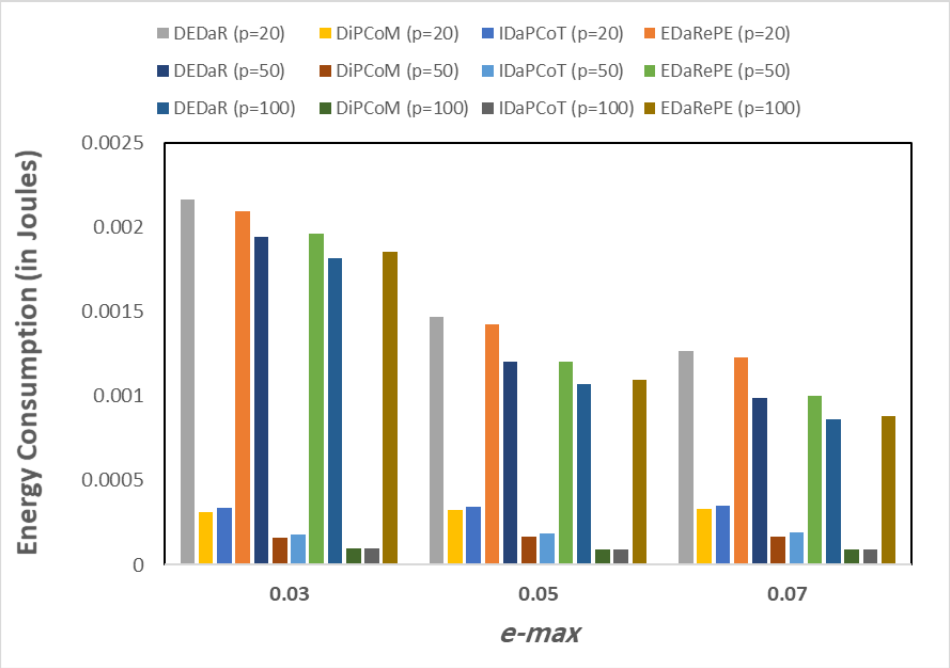


Figure 4.31: Energy Consumption

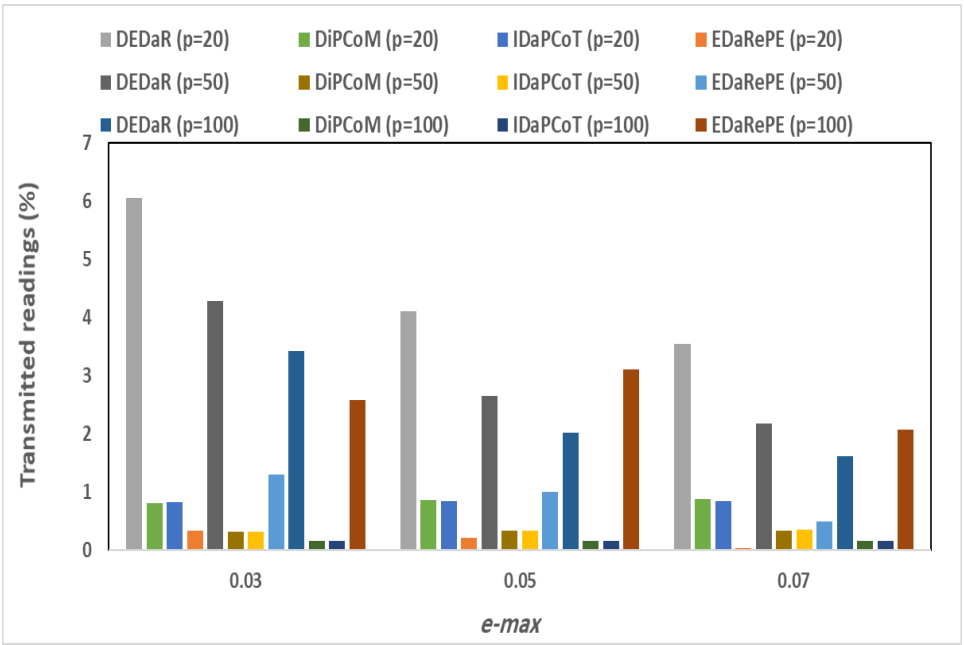


Figure 4.32: Transmitted Reading

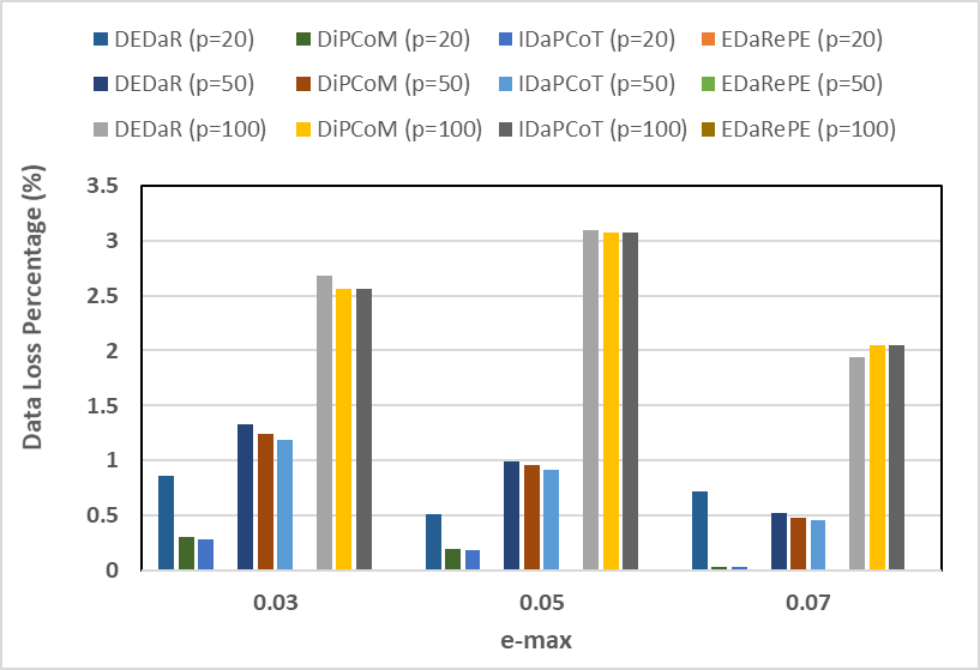


Figure 4.33: Data Lose Percentage

4.8 Second Level

In this stage, further redundant spatial correlated data will be reduced on the received sets of measures from the sensor devices before transmitting them to the base station. The main goal is to enable the Fog gateway to decrease the consumed energy and prolong the network's lifetime whilst maintaining the data's integrity. In this level (Gateway level), k data sets of readings and their repetition "$S^{AP} = (S_1^{AP}, S_2^{AP}, \dots, S_k^{AP})$" have arrived at the Gateway at the end of each period.

4.8.1 TEDaReT Approach Performance Evaluation

The TEDaReT approach use temperature readings during the simulation. It uses different sizes, such as T= 200, 500, and 1000 readings per period. Moreover, the e-max uses different values like 0.03, 0.05, 0.07, and 0.1. The TEDaReT approach is compared with an energy-efficient two-layer data transmission reduction (ETDTR) [32], prefix frequency filtering (PFF) [38], and aggregation and transmission approach (ATP) [39]. In the experimental simulations, some performance metrics are applied to assess the effectiveness of the TEDaReT approach, such as:

1. **Percentage of Transmitted Sets to Cloud.**
2. **Number of pairs of redundant data sets.**
3. **Energy Consumption at Fog Gateway.**

4.8.1.1 Percentage of Transmitted Sets to Cloud

It can be observed from the results in all ***P***= 200, 500, and 1000 presented in Figures 4.34 that TEDaReT approach, ETDTR, Harb (FISHER), Harb (TUKEY), (PFF 0.75), and (PFF 0.8) approaches sent data from 26.09% up to 40.34%, from 31.09% up to 70.38%, from 38.48% up to 45.36%, from 45.5% up to 53.04%, from 54.7% up to 79.8%, and from 61.4% up to 84.1% respectively.

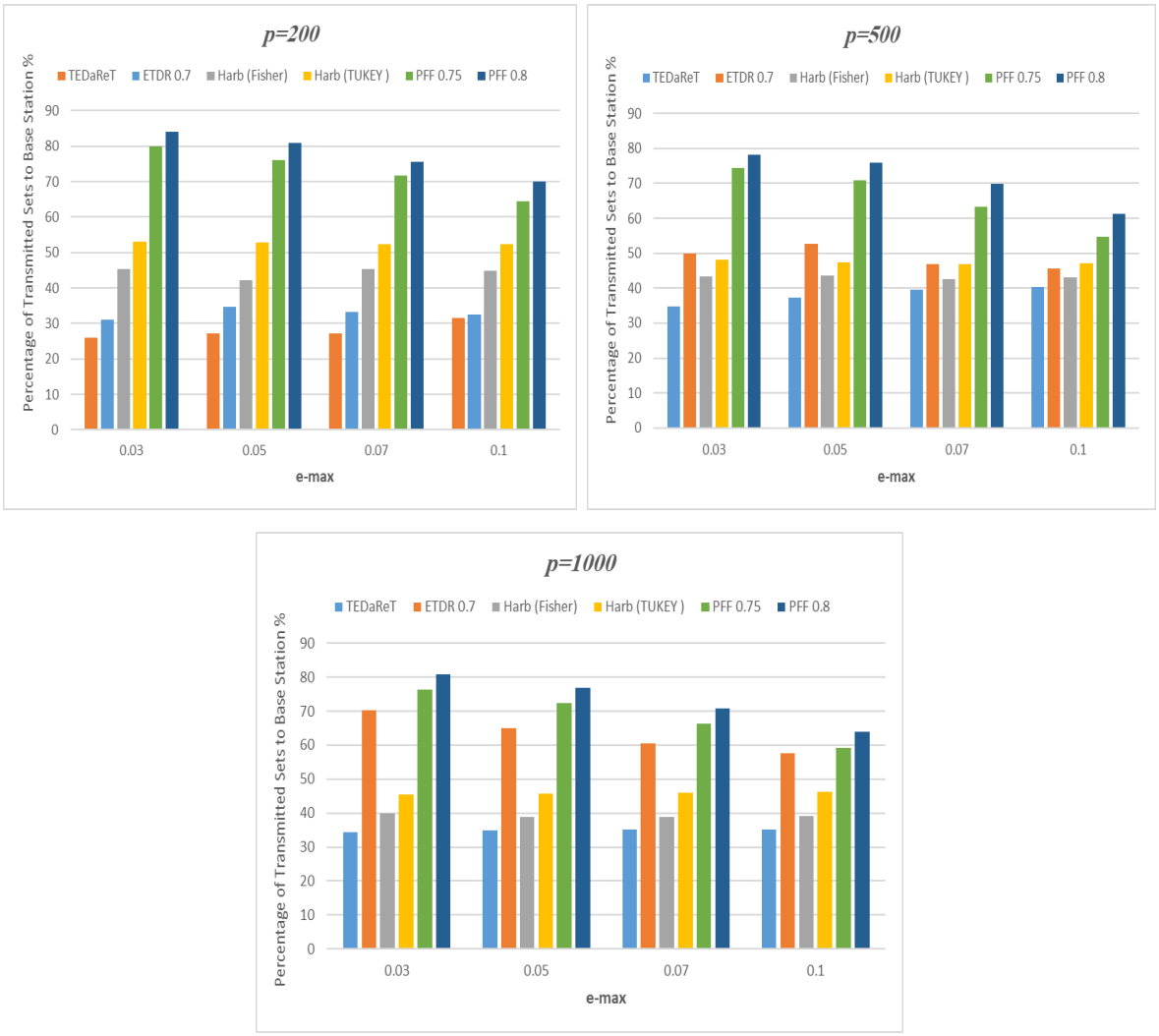


Figure 4.34: Percentage of Transmitted Sets to Cloud

It can be seen from Figure 4.34 that the proposed TEDaReT approach reduced the transmitted data sets efficiently compared with other approaches. Table 4.47 shows values of percentage of transmitted sets to cloud.

Table 4.47: Percentage of Transmitted Sets to Cloud

	200					
e-max	Percentage of Transmitted Sets to Base Station %					
	TEDaReT	ETDR 0.7	Harb (Fisher)	Harb (TUKEY)	PFF 0.75	PFF 0.8
0.03	26.081	31.0928	45.21	53.04	79.8	84.1
0.05	27.186	34.703	42.21	52.89	76.1	80.9
0.07	27.1863	33.232	45.36	52.46	71.8	75.6
0.1	31.63	32.5817	44.92	52.46	64.5	69.9
	500					
e-max	Percentage of Transmitted Sets to Base Station %					
	TEDaReT	ETDR 0.7	Harb (Fisher)	Harb (TUKEY)	PFF 0.75	PFF 0.8
0.03	34.678	49.8257	43.33	48.11	74.4	78.1
0.05	37.214	52.6583	43.62	47.39	71	75.9
0.07	39.527	46.8156	42.6	46.81	63.2	69.8
0.1	40.341	45.6882	43.04	47.1	54.7	61.4
	1000					
e-max	Percentage of Transmitted Sets to Base Station %					
	TEDaReT	ETDR 0.7	Harb (Fisher)	Harb (TUKEY)	PFF 0.75	PFF 0.8
0.03	34.47	70.378	40	45.5	76.2	80.9
0.05	34.85	65.0692	38.84	45.79	72.4	76.8
0.07	35.08	60.416	38.98	45.94	66.4	70.8
0.1	35.14	57.4649	39.13	46.23	59.2	63.9

4.8.1.2 Number of Pairs of Redundant Data Sets

It can be observed from the results in all ***P***= 200, 500, and 1000 presented in Figure 4.35 that TEDaReT approach, ETDTR, Harb (FISHER), Harb (TUKEY), (PFF 0.75), and (PFF 0.8) approaches reduces the number of pairs of redundant data sets from 1185 to 2650, from 315 to 2445, from 827 to 1305, from 622 to 999, from 268 to 808, and from 210 to 688 respectively.

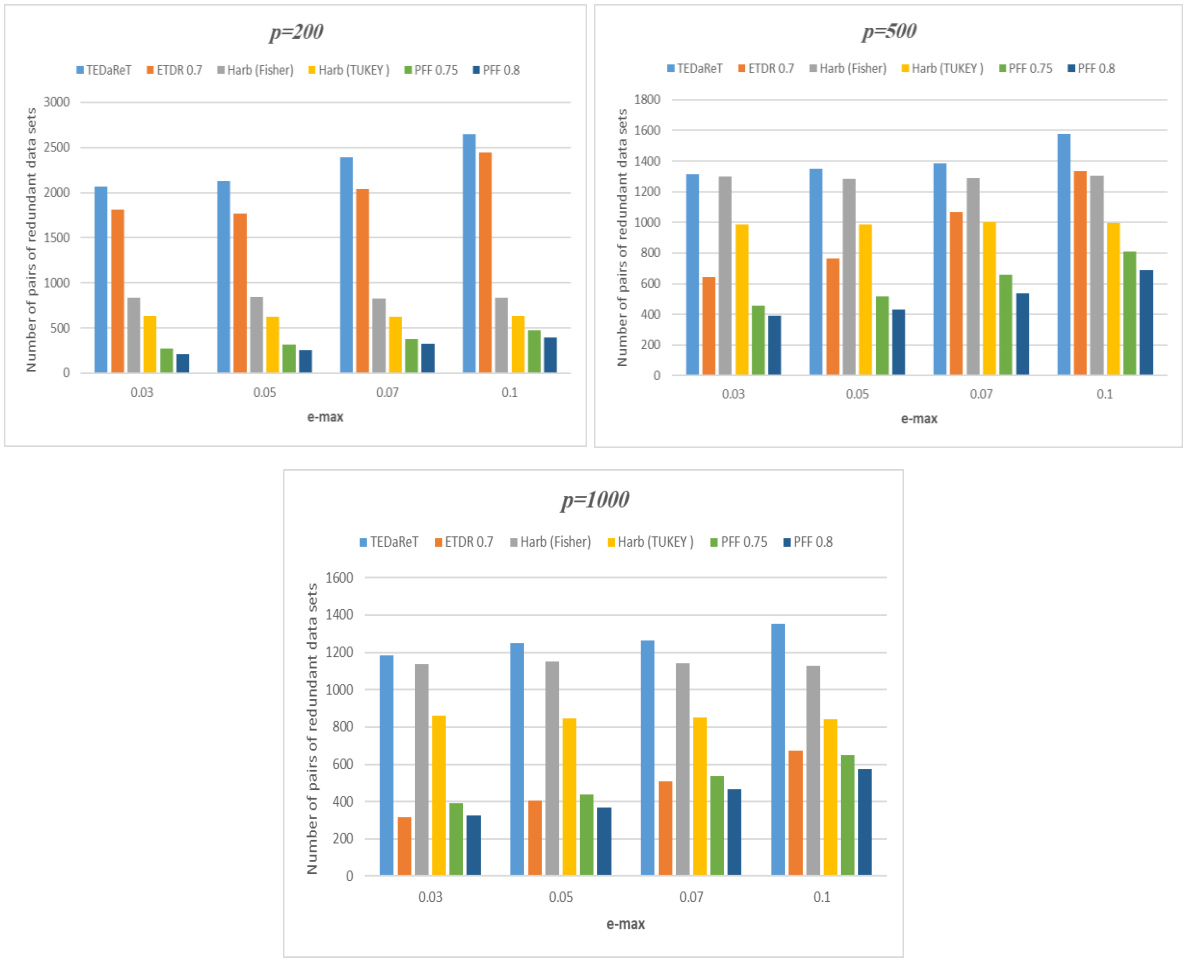


Figure 4.35: Number of Pairs of Redundant Data Sets

It can be seen from Figure 4.35 that the proposed TEDaReT approach reduces the number of pairs of redundant data sets efficiently compared with other approaches. Table 4.48 shows the number of pairs of redundant data sets.

Table 4.48: Number of Pairs of Redundant Data Sets

	200					
e-max	Number of Pairs of Redundant Data Sets					
	TEDaReT	ETDR 0.7	Harb (Fisher)	Harb (TUKEY)	PFF 0.75	PFF 0.8
0.03	2065	1815	838	630	268	210
0.05	2126	1770	846	622	317	253
0.07	2391	2040	827	628	374	323
0.1	2650	2445	832	630	471	399
	500					
e-max	Number of Pairs of Redundant Data Sets					
	TEDaReT	ETDR 0.7	Harb (Fisher)	Harb (TUKEY)	PFF 0.75	PFF 0.8
0.03	1315	645	1299	986	456	390
0.05	1350	765	1286	986	517	429
0.07	1385	1065	1289	999	656	538
0.1	1575	1335	1305	994	808	688
	1000					
e-max	Number of Pairs of Redundant Data Sets					
	TEDaReT	ETDR 0.7	Harb (Fisher)	Harb (TUKEY)	PFF 0.75	PFF 0.8
0.03	1185	315	1135	859	390	326
0.05	1250	405	1149	847	440	370
0.07	1264	510	1141	850	536	465
0.1	1354	675	1129	841	651	576

4.8.1.3 Energy Consumption at Fog Gateway

It can be observed from the results In all ***P***= 200, 500, and 1000 presented in Figure 4.36 that TEDaReT approach, ETDTR, Harb (FISHER), Harb (TUKEY), (PFF 0.75), and (PFF 0.8) approaches calculated the energy consumption from 0.0017 up to 0.0075, from 0.045 up to 0.474, from 0.162 to 1.07, from 0.18 to 1.179, from 0.209 to 1.686, and from 0.222 to 1.743 respectively.

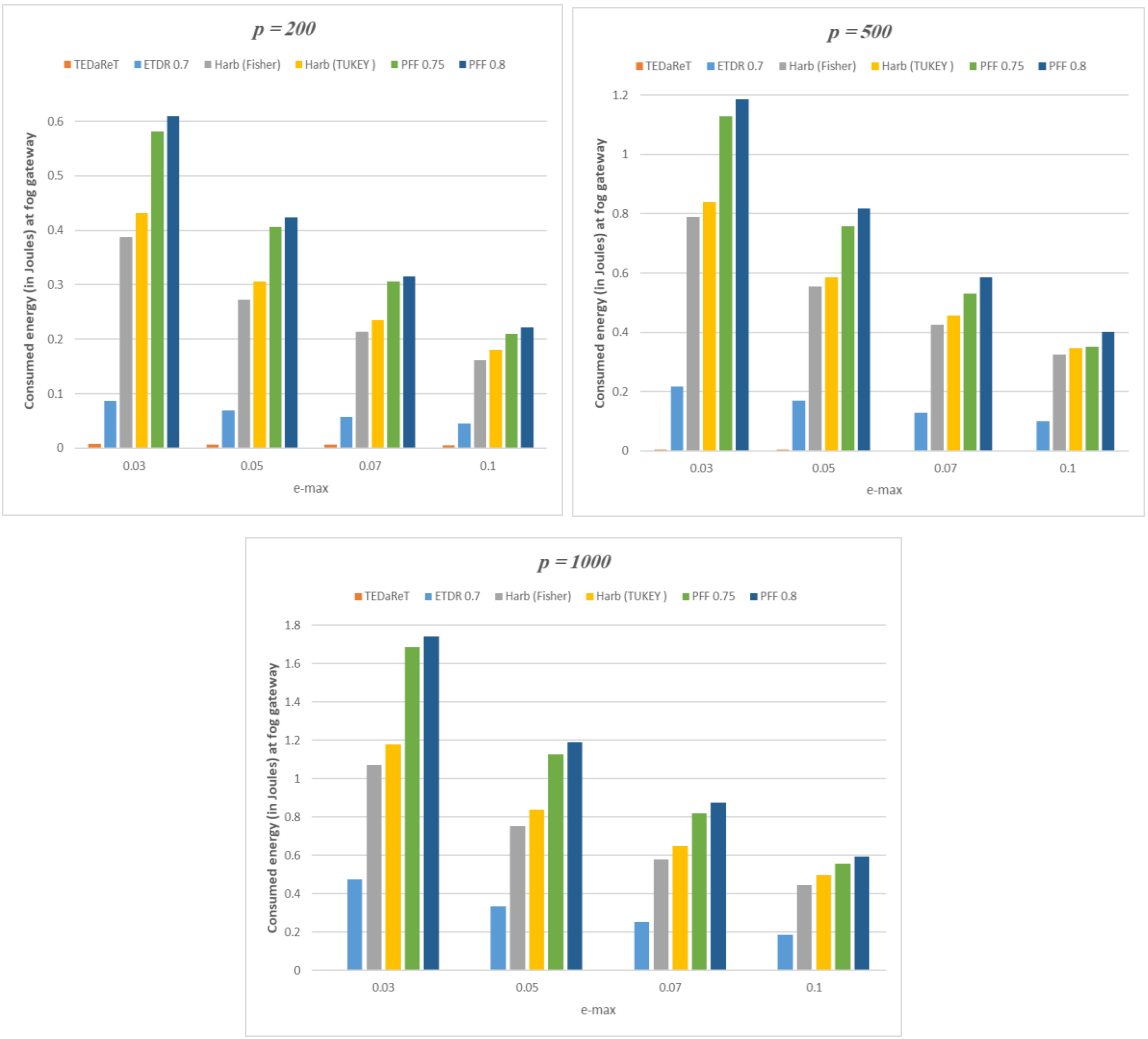


Figure 4.36: Energy Consumption at Fog Gateway

It can be seen from Figure 4.36 that the proposed TEDaReT approach reduces energy consumption efficiently compared with other approaches. Table 4.49 illustrate the energy consumption at Fog Gateway.

Table 4.49: Energy Consumption at Fog Gateway

	200					
e-max	Energy Consumption at Fog Gateway					
	TEDaReT	ETDR 0.7	Harb (Fisher)	Harb (TUKEY)	PFF 0.75	PFF 0.8
0.03	0.007456141	0.086578	0.387	0.431	0.581	0.61
0.05	0.006224648	0.069687	0.273	0.306	0.406	0.424
0.07	0.006004829	0.056916	0.214	0.235	0.306	0.315
0.1	0.00575552	0.044774	0.162	0.18	0.209	0.222
	500					
e-max	Energy Consumption at Fog Gateway					
	TEDaReT	ETDR 0.7	Harb (Fisher)	Harb (TUKEY)	PFF 0.75	PFF 0.8
0.03	0.004528396	0.216978	0.79	0.839	1.128	1.185
0.05	0.003328178	0.168377	0.554	0.585	0.758	0.818
0.07	0.003106201	0.129076	0.426	0.456	0.531	0.586
0.1	0.002752237	0.099281	0.325	0.347	0.352	0.402
	1000					
e-max	Energy Consumption at Fog Gateway					
	TEDaReT	ETDR 0.7	Harb (Fisher)	Harb (TUKEY)	PFF 0.75	PFF 0.8
0.03	0.002220365	0.474953	1.07	1.179	1.686	1.743
0.05	0.00181974	0.334908	0.752	0.839	1.126	1.192
0.07	0.001816503	0.254157	0.581	0.648	0.819	0.875
0.1	0.001742878	0.188237	0.446	0.497	0.556	0.593

4.9 Summary of the Chapter

The performance evaluations for the four proposed approaches as graphs and discussions were presented. The effectiveness and efficiency of the proposed approaches are proven by using several performance parameters such as the number of collected data readings, number of sent data readings, energy consumption, and data accuracy. Also, use the further results with other parameters such as energy consumption, number of sending data, and data loss percentage. In the second level, we using several performance parameters such as Percentage of Transmitted Sets to Cloud, Number of pairs of redundant data

sets, and Energy Consumption at Fog Gateway. Also, a comparisons with some existing related works were done in this chapter. Moreover, a comparison was made between the four approaches presented in the first level. A summary of the chapter and through the results presented in this chapter at both levels, that the proposed approaches achieved the best results compared to the related works and by a large percentage.

CHAPTER FIVE
CONCLUSIONS AND FUTURE WORKS

CHAPTER FIVE: CONCLUSION AND FUTURE WORKS

5.1 Conclusions

The increasing utilization of PSNs by various applications has resulted in a significant amount of sensed data that needs to be transmitted through the network. The elevated communication costs will reduce the lifespan of PSN. Efficient data reduction methods are crucial for eliminating duplicate data in the Periodic Sensor Networks and conserving energy.

We have explored the problem of how to extend the lifetime of PSNs by designing distributed data reduction approaches in this dissertation. Removing unnecessary data to conserve energy while keeping an appropriate degree of data accuracy effectively while monitoring a specific region is the primary goal in this dissertation. This goal is done by using the prediction and data compression techniques for reducing transmitted data over PSNs. This dissertation's work was divided into two levels: Sensor Node (SN) and Fog Gateway (FG).

In the first level, we have proposed four approaches depend on prediction, reduction, and compression approaches to minimize redundant data while preserving a high accuracy and extending the useful lifetime of the network. These approaches named DEDaR, DiPCoM, IDaPCoT, and EDaRePE. In all proposed approaches used AR and ARIMA for prediction and Huffman and LZW for compression and APCA and SAX for data reduction. The results of the simulation demonstrate efficiency in the SN level, that is, the proposed approaches increased the percentage of data reduction, and reduced the number

of sent readings and consumed energy in while maintaining the accuracy of sent data as high, comparison with other approaches.

At the Fog Gateway level, the primary objective is to optimize the energy consumption and extend the network's lifespan of the Fog gateway, while preserving data integrity. We design new approach named Two-Tier Energy-efficient Data Reduction Technique for IoT Networks (TEDaReT) to remove the duplicated sets of data before sending to the cloud. The TEDaReT algorithm is utilized to eliminate duplicate data obtained from sensor nodes by identifying similarities between them, resulting in a reduction of data sensors. The results of the simulation demonstrate efficiency at the FG level, that is, the proposed approach reduces the number of redundant data of sensors the percentage reaching 26.081% in comparison with other approaches.

A Python language-based custom simulator is utilized to evaluate proposed approaches through simulation experiments using real data collected from sensor nodes that are used at the Intel Berkeley Research Lab.

5.2 Future Works

1. Future works will focus on reducing the transmitted data from the Gateway to the cloud using machine learning and deep learning approaches.
2. Also, we plan to use deep learning to predict missing data at the Gateway in order to increase the accuracy of data so that it may be used in decision-making.
3. We plan to apply a scheduling algorithm at the gateway to schedule the sensor nodes into sleep/active modes.
4. Utilize metaheuristic methods to select the optimal representative sets at the gateway based on the spatial correlation between the data sets of sensor nodes to send them to the next level of the network.
5. Finally, we want to apply the suggested approach on a real network.

REFERENCES

REFERENCES

[1] Al-Qurabat, A. K. M., & Kadhum Idrees, A. (2020). Data gathering and aggregation with selective transmission technique to optimize the lifetime of Internet of Things networks. *International Journal of Communication Systems, 33*(11), e4408.

[2] Idrees, S. K., & Idrees, A. K. (2022). New fog computing enabled lossless EEG data compression scheme in IoT networks. *Journal of Ambient Intelligence and Humanized Computing*, 1-14.

[3] Al-Qurabat, A. K. M., Idrees, A. K., & Abou Jaoude, C. (2020, June). Dictionary-based DPCM method for compressing IoT big data. In *2020 International Wireless Communications and Mobile Computing (IWCMC)* (pp. 1290-1295). IEEE.

[4] Al-Qurabat, A. K. M., Abou Jaoude, C., & Idrees, A. K. (2019, June). Two tier data reduction technique for reducing data transmission in IoT sensors. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)* (pp. 168-173). IEEE.34.

[5] Prasad P (2015) Recent trend in wireless sensor network and its applications: a survey. Sens Rev.

[6] Khan S, Pathan ASK, Alrajeh NA (2016) Wireless sensor networks: current status and future trends.

[7] Dargie W, Poellabauer C (2010) Fundamentals of wireless sensor networks: theory and practice. JohnWiley & Sons.

[8] Kumar SA, Ilango P (2018) The impact of wireless sensor network in the field of precision agriculture: a review. Wireless Personal Commun 98(1):685–698.

[9] Goel K, Bindal AK (2018) Wireless sensor network in precision agriculture: a survey report. In: 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC), pp. 176–181. IEEE.

[10] Alhmiedat T et al (2015) A survey on environmental monitoring systems using wireless sensor networks. J. Netw 10(11):606–615.

[11] Adu-Manu KS, Tapparello C, Heinzelman W, Katsriku FA, Abdulai JD (2017) Water quality monitoring using wireless sensor networks: current trends and future research directions. ACM Trans Sens Netw (TOSN) 13(1):1–41.

[12] Saad A, Gamatié A et al (2020) Water management in agriculture: a survey on current challenges and technological solutions. IEEE Access 8:38082–38097.

[13] Khan RA, Pathan ASK (2018) The state-of-the-art wireless body area sensor networks: a survey. Int J Distrib Sens Netw 14(4):1550147718768994.

[14] Chelbi S, Duvallet C, Abdouli M, Bouaziz R (2016) Event-driven wireless sensor networks based on consensus pp. 1–6.

[15] Maheshwari A, Chand N (2019) A survey on wireless sensor networks coverage problems pp. 153–164.

[16] Yu, L., Wang, N., and Meng, X., (2005), "Real-time forest fire detection with wireless sensor networks", In Wireless Communications, Networking and Mobile Computing, 2005. Proceedings, 2005 International Conference on (Vol. 2, pp. 1214-1217), IEEE.

[17] Dalbro, M., Eikeland, E., in't Veld, A. J., Gjessing, S., Lande, T. S., Riis, H. K., and Søråsen, O., (2008), "Wireless sensor networks for off-shore oil and gas installations", In Sensor Technologies and Applications, 2008, SENSORCOMM'08. Second International Conference on (pp. 258-263), IEEE.

[18] Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J., (2002), "Wireless sensor networks for habitat monitoring", In Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (pp. 88-97). Acm.

[19] Goense, D., and Thelen, J., (2005), "Wireless sensor networks for precise Phytophthora decision support", In 2005 ASAE Annual Meeting (p. 1), American Society of Agricultural and Biological Engineers.

[20] Akyildiz, I. F., & Vuran, M. C. (2010). *Wireless sensor networks*. John Wiley & Sons.

[21] Mejia, J., Ochoa-Zezzatti, A., Cruz-Mejía, O., & Mederos, B. (2020). Prediction of time series using wavelet Gaussian process for wireless sensor networks. *Wireless Networks*, *26*(8), 5751-5758.

[22] Zhang, C., Liu, Y., Wu, F., Fan, W., Tang, J., & Liu, H. (2019). Multi-dimensional joint prediction model for IoT sensor data search. *IEEE Access*, *7*, 90863-90873.

[23] Ismael, W. M., Gao, M., Al-Shargabi, A. A., & Zahary, A. (2019). An in-networking double-layered data reduction for internet of things (IoT). *Sensors*, *19*(4), 795.

[24] Fathy, Y., & Barnaghi, P. (2019). Quality-based and energy-efficient data communication for the internet of things networks. *IEEE Internet of Things Journal*, *6*(6), 10318-10331.

[25] Almeida Jr, F. R., Brayner, A., Rodrigues, J. J., & Maia, J. E. B. (2017). Improving multidimensional wireless sensor network lifetime using pearson correlation and fractal clustering. *Sensors*, *17*(6), 1317.

[26] Liazid, H., Lehsaini, M., & Liazid, A. (2019). An improved adaptive dual prediction scheme for reducing data transmission in wireless sensor networks. *Wireless Networks*, *25*, 3545-3555.

[27] Russo, A., Verdier, F., & Miramond, B. (2018). Energy saving in a wireless sensor network by data prediction by using self-organized maps. *Procedia computer science*, *130*, 1090-1095.

[28] Karjee, J., & Kleinsteuber, M. (2017). Data estimation with predictive switching mechanism in wireless sensor networks. *International Journal of Sensor Networks*, *25*(3), 184-197.

[29] Wu, M., Tan, L., & Xiong, N. (2014). A structure fidelity approach for big data collection in wireless sensor networks. *Sensors*, *15*(1), 248-273.

[30] Dhimal, S., & Sharma, K. (2015). Energy conservation in wireless sensor networks by exploiting inter-node data similarity metrics. *International Journal of Energy, Information and Communications*, *6*(2), 23-32.

[31] Alhussaini, R., Idrees, A. K., & Salman, M. A. (2018). Data transmission protocol for reducing the energy consumption in wireless sensor networks. In *New Trends in Information and Communications Technology Applications: Third International Conference, NTICT 2018, Baghdad, Iraq, October 2–4, 2018, Proceedings 3* (pp. 35-49). Springer International Publishing.

[32] Idrees, A.K., Alhussaini, R. & Salman, M.A. (2023). Energy-efficient two-layer data transmission reduction protocol in periodic sensor networks of IoTs. *Pers Ubiquit Comput* **27**, 139–158. https://doi.org/10.1007/s00779-020-01384-5.

[33] Idrees, A. K., Abou Jaoude, C., & Al-Qurabat, A. K. M. (2020, October). Data reduction and cleaning approach for energy-saving in wireless sensors networks of IoT.

In *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (pp. 1-6). IEEE.

[34] Idrees, A. K., & Al-Qurabat, A. K. M. (2021). Energy-efficient data transmission and aggregation protocol in periodic sensor networks based fog computing. *Journal of Network and Systems Management*, *29*(1), 4..

[35] Wang, H., Yemeni, Z., Ismael, W. M., Hawbani, A., & Alsamhi, S. H. (2021). A reliable and energy efficient dual prediction data reduction approach for WSNs based on Kalman filter. *IET Communications*, *15*(18), 2285-2299.

[36] Jarwan, A., Sabbah, A., & Ibnkahla, M. (2019). Data transmission reduction schemes in WSNs for efficient IoT systems. *IEEE Journal on Selected Areas in Communications*, *37*(6), 1307-1324.

[37] Fathy, Y., Barnaghi, P., & Tafazolli, R. (2018, February). An adaptive method for data reduction in the internet of things. In *2018 IEEE 4th World Forum on Internet of things (WF-IoT)* (pp. 729-735). IEEE.

[38] Bahi, J. M., Makhoul, A., & Medlej, M. (2014). A two tiers data aggregation scheme for periodic sensor networks. *Ad Hoc Sens. Wirel. Networks*, *21*(1-2), 77-100.

[39] Harb, H., Makhoul, A., Couturier, R., & Medlej, M. (2015, June). ATP: An aggregation and transmission protocol for conserving energy in periodic sensor networks. In *2015 IEEE 24th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises* (pp. 134-139). IEEE.

[40] Zhu R, Yu M, Li Y, Wang J, Liu L. (2021). Edge sensing-enabled multistage hierarchical clustering deredundancy algorithm in WSNs. Wireless Commun Mobile Comput.;2021:1-14.

[41] Rida, M., Makhoul, A., Harb, H., Laiymani, D., & Barhamgi, M. (2019). EK-means: A new clustering approach for datasets classification in sensor networks. *Ad Hoc Networks*, *84*, 158-169.

[42] Loganathan, D., Balasubramani, M., & Sabitha, R. (2021). Energy aware efficient data aggregation (EAEDAR) with re-scheduling mechanism using clustering techniques in wireless sensor networks. *Wireless Personal Communications*, *117*, 3271-3287.

[43] Alam, M. K., Aziz, A. A., Latif, S. A., & Awang, A. (2020). Error-aware data clustering for in-network data reduction in wireless sensor networks. *Sensors*, *20*(4), 1011.

[44] Shahina, K., & Pradeep Kumar, T. S. (2022). Similarity-based clustering and data aggregation with independent component analysis in wireless sensor networks. *Transactions on Emerging Telecommunications Technologies*, *33*(7), e4462.

[45] Tsiropoulou, E. E., Paruchuri, S. T., & Baras, J. S. (2017, March). Interest, energy and physical-aware coalition formation and resource allocation in smart IoT applications. In *2017 51st Annual conference on information sciences and systems (CISS)* (pp. 1-6). IEEE.

[46] Idrees, A. K., Al-Qurabat, A. K. M., Abou Jaoude, C., & Al-Yaseen, W. L. (2019, June). Integrated divide and conquer with enhanced k-means technique for energy-saving data aggregation in wireless sensor networks. In *2019 15th International wireless communications & mobile computing conference (IWCMC)* (pp. 973-978). IEEE.

[47] Nayak, A., and Stojmenovic, I., (2010), "*Wireless sensor and actuator networks: algorithms and protocols for scalable coordination and data communication"*. John Wiley & Sons.

[48] Rajagopalan, R., and Varshney, P. K., (2006), "Data aggregation techniques in sensor networks: A survey".

[49] Hoogenboom, G., and Coker, D. D., (2003), "The Georgia automated environmental monitoring network: Ten years of weather information for water resources management", Georgia Institute of Technology.

[50] Werner-Allen, G., Johnson, J., Ruiz, M., Lees, J., and Welsh, M., (2005), "Monitoring volcanic eruptions with a wireless sensor network", In *Wireless Sensor Networks, 2005, Proceedings of the Second European Workshop on* (pp. 108-120). IEEE.

[51] Al-Qurabat, A. K.M., and Idrees, A. K., (2018, In Press), "Energy-efficient Adaptive Distributed Data Collection method for Periodic Sensor Networks", International Journal of Internet Technology and Secured Transactions, Vol. x, No. x.

[52] Rawat, P., Singh, K. D., Chaouchi, H., and Bonnin, J. M., (2014), "Wireless sensor networks: a survey on recent developments and potential synergies", The Journal of supercomputing, Vol. 68, No. 1, pp.1-48.

[53] Kobo, H. I., Abu-Mahfouz, A. M., and Hancke, G. P., (2017), "A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements", IEEE Access, Vol. 5, pp.1872-1899.

[54] Yetgin, H., Cheung, K. T. K., El-Hajjar, M., and Hanzo, L. H., (2017), "A Survey of Network Lifetime Maximization Techniques in Wireless Sensor Networks", IEEE Communications Surveys & Tutorials, Vol. 19, No. 2, pp.828-854.

[55] Hema, N., and Kant, K., (2013), "Optimization of sensor deployment in WSN for precision irrigation using spatial arrangement of permanent crop", In Contemporary Computing (IC3), 2013 Sixth International Conference on (pp. 455-460), IEEE.

[56] Sharma, S., Bansal, R. K., & Bansal, S., (2013), "Issues and challenges in wireless sensor networks", In *Machine Intelligence and Research Advancement (ICMIRA), 2013 International Conference on* (pp. 58-62), IEEE

[57] Padmavathi, D. G., and Shanmugapriya, M., (2009), "A survey of attacks, security mechanisms and challenges in wireless sensor networks", *arXiv preprint arXiv:0909.0576*.

[58] Kumar, R. (2014). A survey on data aggregation and clustering schemes in underwater sensor networks. *International Journal of Grid and Distributed Computing*, *7*(6), 29-52.

[59] Nokhanji, N., and Hanapi, Z. M., (2014), "A survey on cluster-based routing protocols in wireless sensor networks", *Journal of Applied Sciences*, Vol. 14, No. 18, pp.2011-2022.

[60] Mulligan, R., and Ammari, H. M., (2010), "Coverage in wireless sensor networks: A survey", *Network Protocols and Algorithms*, Vol. 2, No. 2, pp.27-53.

[61] Pagar, A. R., and Mehetre, D. C., (2015), "A Survey on energy efficient sleep scheduling in wireless sensor network", *International Journal*, Vol. 5, No. 1.

[62] Sirsikar, S., and Anavatti, S., (2015), "Issues of data aggregation methods in wireless sensor network: A survey", *Procedia Computer Science*, Vol. 49, pp.194-201.

[63] Medina-Salgado, B., Sanchez-DelaCruz, E., Pozos-Parra, P., & Sierra, J. E. (2022). Urban traffic flow prediction techniques: A review. Sustainable Computing: Informatics and Systems, 100739.

[64] Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

[65] Korstanje, J. (2021). *Advanced forecasting with Python*. United States: Apress.

[66] Wang, X. (2022). Research on the prediction of per capita coal consumption based on the ARIMA–BP combined model. Energy Reports, 8, 285-294.

[67] Kurawarwala AA, Matsuo H (1998) Product growth models for medium-term forecasting of short life cycle products. Technol Forecast Soc Change 57(3):169–196.

[68] Miller DM, Williams D (2003) Shrinkage estimators of time series seasonal factors and their effect on forecasting accuracy. Int J Forecast 19(4):669–684

[69] Jain G, Mallick B (2017) A study of time series models arima and ets. Available at SSRN 2898968

[70] Abdelghafar, S., Darwish, A., & Ali, A. (2023, March). Short-Term Forecasting of GDP Growth for the Petroleum Exporting Countries Based on ARIMA Model. In The International Conference on Artificial Intelligence and Computer Vision (pp. 399-406). Cham: Springer Nature Switzerland.

[71] Fildes, R., Ma, S., & Kolassa, S. (2022). Retail forecasting: Research and practice. International Journal of Forecasting, 38(4), 1283-1318.

[72] Li, M. W., Xu, D. Y., Geng, J., & Hong, W. C. (2022). A ship motion forecasting approach based on empirical mode decomposition method hybrid deep learning network and quantum butterfly optimization algorithm. Nonlinear dynamics, 107(3), 2447-2467.

[73] Xia, F. L., Jarad, F., Hashemi, M. S., & Riaz, M. B. (2022). A reduction technique to solve the generalized nonlinear dispersive mK (m, n) equation with new local derivative. Results in Physics, 38, 105512.

[74] Hasegawa, A., Ishihara, T., Thomas, M. A., & Pan, T. (2022). Noise reduction profile: A new method for evaluation of noise reduction techniques in CT. Medical physics, 49(1), 186-200.

[75] Chen, R., Yang, D., & Zhang, C. H. (2022). Factor models for high-dimensional tensor time series. Journal of the American Statistical Association, 117(537), 94-116.

[76] Wang, D., Zheng, Y., Lian, H., & Li, G. (2022). High-dimensional vector autoregressive time series modeling via tensor decomposition. Journal of the American Statistical Association, 117(539), 1338-1356.

[77] Zaini, N. A., Ean, L. W., Ahmed, A. N., & Malek, M. A. (2022). A systematic literature review of deep learning neural network for time series air quality forecasting. Environmental Science and Pollution Research, 1-33.

[78] Chen, X., & Güttel, S. (2023). An efficient aggregation method for the symbolic representation of temporal data. ACM Transactions on Knowledge Discovery from Data, 17(1), 1-22.

[79] Lui, J. H., Nguyen, N. D., Grutzner, S. M., Darmanis, S., Peixoto, D., Wagner, M. J., ... & Luo, L. (2021). Differential encoding in prefrontal cortex projection neuron classes across cognitive tasks. Cell, 184(2), 489-506.

[80] Al-Qurabat, M., & Kadhum, A. (2021). A lightweight Huffman-based differential encoding lossless compression technique in IoT for smart agriculture. International Journal of Computing and Digital System.

[81] Jin, S. W., & Lee, I. (2021). Differential encoding of place value between the dorsal and intermediate hippocampus. Current Biology, 31(14), 3053-3072.

[82] Kagita, M. K., Thilakarathne, N., Bojja, G. R., & Kaosar, M. (2021). A lossless compression technique for Huffman-based differential encoding in IoT for smart agriculture. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 29(Supp. 02), 317-332.

[83] Azar, J., Makhoul, A., Barhamgi, M., & Couturier, R. (2019). An energy efficient IoT data compression approach for edge machine learning. Future Generation Computer Systems, 96, 168-175.

[84] Hua, S., Wang, C., Lam, H. K., & Wen, S. (2023). An incremental learning method with hybrid data over/down-sampling for sEMG-based gesture classification. Biomedical Signal Processing and Control, 83, 104613.

[85] Lim, Y., Kwon, J., & Oh, H. S. (2021). Principal component analysis in the wavelet domain. Pattern Recognition, 119, 108096.

[86] Azar, J., Makhoul, A., Couturier, R., & Demerjian, J. (2020). Robust IoT time series classification with data compression and deep learning. Neurocomputing, 398, 222-234.

[87] Liang, T., Glossner, J., Wang, L., Shi, S., & Zhang, X. (2021). Pruning and quantization for deep neural network acceleration: A survey. Neurocomputing, 461, 370-403.

[88] Nguyen, L., & Nguyen, H. T. (2020). Mobility based network lifetime in wireless sensor networks: A review. Computer networks, 174, 107236.

[89] Jain, K., Kumar, A., & Singh, A. (2023). 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, 34(1), e4674.

[90] Wahyono, I. D., Jong, G. J., Asfani, K., Afandi, A. N., & Fadlika, I. (2019, October). New algorithm to determine prediction accuracy on wireless sensor networks. In 2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE) (Vol. 6, pp. 144-147). IEEE.

[91] Alrabea, A., Alzubi, O. A., & Alzubi, J. A. (2019). A task-based model for minimizing energy consumption in WSNs. Energy Systems, 1-18.

[92] Kho, E. P., Chua, S. N. D., Lim, S. F., Lau, L. C., & Gani, M. T. N. (2022). Development of young sago palm environmental monitoring system with wireless sensor networks. Computers and Electronics in Agriculture, 193, 106723.

[93] Thakur, D., Kumar, Y., Kumar, A., & Singh, P. K. (2019). Applicability of wireless sensor networks in precision agriculture: A review. Wireless Personal Communications, 107, 471-512.

[94] Nurgaliyev, M., Saymbetov, A., Yashchyshyn, Y., Kuttybay, N., & Tukymbekov, D. (2020). Prediction of energy consumption for LoRa based wireless sensors network. Wireless Networks, 26, 3507-3520.

[95] Fu, T. C., (2011), "A review on time series data mining", *Engineering Applications of Artificial Intelligence*, Vol. 24, No. 1, pp.164-181.

[96] Cassisi, C., Montalto, P., Aliotta, M., Cannata, A., and Pulvirenti, A., (2012), "Similarity measures and dimensionality reduction techniques for time series data mining", In *Advances in data mining knowledge discovery and applications*. InTech.

[97] Wang, C., Ma, H., He, Y., and Xiong, S., (2012), "Adaptive approximate data collection for wireless sensor networks", IEEE Transactions on Parallel and Distributed Systems, Vol. 23, No. 6, pp.1004-1016.

[98] Heinzelman, W. R., Chandrakasan, A., and Balakrishnan, H., (2000), "Energy-efficient communication protocol for wireless microsensor networks", In *System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on (pp. 10-pp), IEEE.*

[99] Box GE, Jenkins GM, Reinsel GC, Ljung GM (2015) Time series analysis: forecasting and control. John Wiley & Sons.

[100] Saad, G., Harb, H., Abou Jaoude, C., & Jaber, A. (2019, October). A distributed round-based prediction model for hierarchical large-scale sensor networks. In 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) (pp. 1-6). IEEE.

[101] Jain, K., Agarwal, A., & Kumar, A. (2021). A novel data prediction technique based on correlation for data reduction in sensor networks. In Proceedings of International Conference on Artificial Intelligence and Applications: ICAIA 2020 (pp. 595-606). Springer Singapore.

[102] Mehrani, M., Attarzadeh, I., & Hosseinzadeh, M. (2020). Sampling rate prediction of biosensors in wireless body area networks using deep-learning methods. Simulation Modelling Practice and Theory, 105, 102101.

[103] Kök, İ., & Özdemir, S. (2020). Deepmdp: A novel deep-learning-based missing data prediction protocol for iot. IEEE Internet of Things Journal, 8(1), 232-243.

[104] Jain, K., & Kumar, A. (2020). An energy-efficient prediction model for data aggregation in sensor network. Journal of Ambient Intelligence and Humanized Computing, 11, 5205-5216.

APPENDICES

APPENDICES

Appendices A: Published Papers

Received: 8 March 2022 | Revised: 31 May 2022 | Accepted: 13 June 2022
DOI: 10.1002/dac.5282

RESEARCH ARTICLE

WILEY

Distributed energy-efficient data reduction approach based on prediction and compression to reduce data transmission in IoT networks

Ahmed Mohammed Hussein[1,2] | **Ali Kadhum Idrees**[2] | **Raphael Couturier**[3]

[1]Department of Information Networks, College of Information Technology, University of Babylon, Babylon, Iraq
[2]Department of Computer Science, University of Babylon, Babylon, Iraq
[3]FEMTO-ST Institute/CNRS, Universite Bourgogne Franche-Comte, Belfort, France

Correspondence
Ali Kadhum Idrees, Department of Computer Science, University of Babylon, Babylon, Iraq.
Email: ali.idrees@uobabylon.edu.iq

Summary
In the modern world, it will be necessary to deploy a large number of sensor devices to sense everything around us in order to detect changes, risks, and hazards and to mitigate them. This increasing number of sensor devices represents an essential data provider in the Internet of Things (IoT). The devices generate and transmit a huge amounts of data which requires a large amount of storage and high processing power to come real-time processing and speed up the network. It also leads to an increase in high energy consumption. Thus, it is important to remove redundant data to reduce the data transmission before sending it to the gateway while maintaining a good level of data quality. In this paper, a distributed energy-efficient data reduction (DEDaR) approach based on prediction and compression to minimize the data transmission in IoT Networks is proposed. The DEDaR is used in periods to make decision. In each period, the autoregressive prediction (ARP) is used to predict the data of the next period and make a decision on whether to send the data of the current period to the gateway or not. In the case of data transmission, the redundant data are eliminated using an efficient compression approach based on adaptive piecewise constant approximation (APCA), symbolic aggregate approximation (SAX), and finally fixed code dictionary (FCD) based on Huffman encoding. The simulation results based on real-sensed data show that the proposed DEDaR approach outperforms the other recent methods in terms of data reduction percentage, transmitted data size, energy consumption, and data accuracy.

KEYWORDS
data compression, data reduction, IoT, network lifetime, prediction, sensor networks

1 | INTRODUCTION

The fundamental aspect of the Internet of Things (IoT) is to allow the communication of virtual and physical things with each other.[1] IoT systems include embedded intelligence, wireless sensor networks, and cloud computing. Sensors, cameras, radio frequency identifier (RFID), and other devices are used by IoT systems to gather environmental data.[2] These systems are capable of providing sophisticated services such as remote management, online analytics, and real-

The Journal of Supercomputing
https://doi.org/10.1007/s11227-023-05317-w

A distributed prediction–compression-based mechanism for energy saving in IoT networks

Ahmed Mohammed Hussein[1] · **Ali Kadhum Idrees**[1] · **Raphaël Couturier**[1]

Accepted: 16 April 2023

Abstract
Nowadays, the number of Internet of things (IoT) devices has rapidly increased due to their increasing use in different real-world applications. The sensor devices represent the basic element of the IoT network because they gather data from various environments and situations, while the sink node serves as the network's brain because it processes the data and makes decisions. However, the large amount of data that the sensor devices gather and send to the gateway toward the sink, on the one hand, causes the sensor's limited energy to be depleted and, on the other hand, makes it more difficult to achieve the decisions using these data at the sink. Therefore, before sending data to the gateway, it is important to get rid of any duplicate data while maintaining a high level of data quality. In this paper, a distributed prediction–compression-based mechanism (DiPCoM) for saving power in IoT networks is suggested. DiPCoM makes periodic decisions on sending the data to the gateway. It uses the autoregressive integrated moving average prediction method in each period to predict the next period's data and decide whether the current data should be sent to the gateway. When the decision is made to send the data to the gateway, an effective compression approach is used by DiPCoM to get rid of the duplicate data. It combines different data transmission reduction techniques such as adaptive piecewise constant approximation, differential encoding, symbolic aggregate approximation, and Lempel–Ziv–Welch. Simulation results based on real-world data show that the DiPCoM method is better than other techniques in terms of energy consumption, data reduction ratio, transferred data size, and data accuracy.

Keywords IoT · Sensor networks · Prediction · Data reduction · Data compression · Network lifetime

✉ Ali Kadhum Idrees
ali.idrees@uobabylon.edu.iq

[1] Department of Information Networks, College of Information Technology, University of Babylon, Babylon, Iraq

Published online: 04 May 2023

الخلاصة

في العالم الحديث، سيكون من الضروري نشر عدد كبير من أجهزة الاستشعار لاستشعار كل شيء حولنا من أجل اكتشاف التغيرات والمخاطر والأخطار والتخفيف منها. يمثل هذا العدد المتزايد من أجهزة الاستشعار مزودًا أساسيًا للبيانات في إنترنت الأشياء (IoT). تقوم الأجهزة بتوليد ونقل كمية هائلة من البيانات التي تتطلب قدرًا كبيرًا من التخزين وقدرة معالجة عالية لتأتي في المعالجة في الوقت الفعلي وتسريع الشبكة. كما أنه يؤدي إلى زيادة في استهلاك الطاقة العالية. وبالتالي، من المهم إزالة البيانات الزائدة لتقليل نقل البيانات قبل إرسالها إلى البوابة مع الحفاظ على مستوى جيد من جودة البيانات.

يعمل هذا البحث على مستويين: الأول هو مستوى عقدة الاستشعار (SN) والثاني هو مستوى بوابة الضباب .(GW) اقترح مستوى SN أربعة أساليب للتنبؤ بالبيانات ومعالجتها موفرة للطاقة لتقليل البيانات الزائدة عن الحاجة وتوفير الطاقة مع الحفاظ على جودة مناسبة للبيانات المستلمة على المستوى التالي من الشبكة. استخدم النهج الأول) تقليل البيانات الموزعة الموفرة للطاقة ((DEDaR) التنبؤ بالانحدار التلقائي (AR) وأساليب ضغط هوفمان. استخدم النهج الثاني) آلية قائمة على الضغط الموزع ((DiPCoM) تنبؤ ARIMAوأساليب ضغط .LZW الطريقة الثالثة) تقنيات التنبؤ والضغط المتكاملة للبيانات ((IDaPCoT) تستخدم أساليب التنبؤ AR وضغط .LZW النهج الرابع) التنبؤ والتشفير القائم على تقليل البيانات الموفر للطاقة ((EDaRePE) يستخدم تنبؤ ARIMA وأساليب ضغط هوفمان. استخدمت جميع الأساليب على مستوى SN تقنيات التقريب الثابت المجزأ (APCA) والتقريب التجميعي الرمزي (SAX) لتقليل البيانات.

على مستوىFW ، قمنا بتصميم تقنية جديدة لتقليل البيانات الموفرة للطاقة من مستويين لشبكات إنترنت الأشياء (TEDaReT) لإزالة التكرارات بين بيانات أجهزة الاستشعار قبل إرسالها إلى السحابة. يتم استخدام TEDaReT لإزالة التكرارات بين بيانات

أجهزة الاستشعار التي تم الحصول عليها من مستوى عقدة الاستشعار عن طريق تحديد أوجه التشابه بينها، مما يؤدي إلى تقليل بيانات أجهزة الاستشعار قبل إرسالها إلى السحابة.

يتم استخدام جهاز محاكاة مخصص قائم على لغة بايثون لتقييم الأساليب المقترحة من خلال تجارب المحاكاة باستخدام بيانات حقيقية تم جمعها من عقد الاستشعار المستخدمة في مختبر أبحاث إنتل بيركلي.

أظهرت نتائج المحاكاة كفاءة في مستوىSN ، أي أن الأساليب المقترحة زادت نسبة تخفيض البيانات بنسبة 93.14%، 99.71%، 99.72%، و97.3% على التوالي مقارنة مع الأساليب الأخرى، وفي عدد القراءات المرسلة. يصل تخفيض النفقات العامة إلى 93.44% و96.05% و96.54 و93.9% على التوالي. في الطاقة المستهلكة وصلت إلى 0.0010%، 0.000209%، 0.00019%، و0.000212% على التوالي، وفي مع الحفاظ على دقة البيانات المرسلة تصل إلى 99.33، 99.73%، 99.79%، و99.737% على التوالي. ونتيجة لذلك، اعتمادا على النتائج التي تم الحصول عليها يمكننا القول أن نهج IDaPCoTهو أفضل نهج مقارنة بين الأساليب الأربعة المقترحة.

أظهرت نتائج المحاكاة كفاءة على مستوى FG، أي أن الطريقة المقترحة تقلل من عدد البيانات الزائدة للحساسات بنسبة تصل إلى 26.081% مقارنة بالطرق الأخرى.

وزارة التعليم العالي والبحث العلمي
جامعـــــــــة بابـــــــــــل
كليـــــة تكنولوجيا المعلومـــــات
قسم شبكـــــات المعلومـــــات

التنبؤات المدركة للطاقة وأساليب المعالجة لتقليل تكلفة الاتصال في شبكات انترنيت الأشياء

أطروحة مقدمة الى مجلس كلية تكنولوجيا المعلومات في جامعة بابل وهي
جزء من متطلبات الحصول على شهادة الدكتوراه فلسفة في تكنولوجيا المعلومات / شبكات المعلومات

مقدمة من قبل

أحمد محمد حسين الغزالي

باشراف

الاستاذ الدكتور: علي كاظم ادريس السعدي

الاستاذ الدكتور: رافائيل كوتوريبه

أيلول 2023 م

1445 هـ