Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Department of Information Networks

AN ENHANCED LZW COMPERSSION ALGORITHM TO IMPROVE THE PERFORMANCE OF WIRELESS SENSOR NETWORK

A Thesis

Submitted to the Council of the College of Information Technology for the Postgraduate Studies/University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology/Information Networks

By

Zahraa Mazin Bahlool Salih

Supervised by

Asst. Prof. Saif Mahmoud Khalaf Hussein Alalak

2022 A.D **1444 A.H**

Supervisor Certification

I certify that the thesis entitled (**AN ENHANCED LZW COMPERSSION ALGORITHM TO IMPROVE THE PERFORMANCE OF WIRELESS SENSOR NETWORK**) was prepared under my supervision Asst. Prof. Dr. Saif Alalak at the department of Information Networks/ College of Information Technology/the University of Babylon as partial fulfillment of the requirements of the degree of master's in information technology-Information Networks.

Signature:

Supervisor Name: Asst. Prof. Dr. Saif Mahmoud Alalak

Date: / /2022

The Head of the Department Certification

Given the available recommendations, I forward the thesis entitled "**AN ENHANCED LZW COMPERSSION ALGORITHM TO IMPROVE THE PERFORMANCE OF WIRELESS SENSOR NETWORK**" for debate by the examination committee.

Signature:

Prof. Dr. Saad Talib Hasson

Head of Information Networks Department

Date: / /2022

Certification of the Examination Committee

We, the undersigned, certify that (**Zahraa Mazin Bahlool Sahih**) candidate for the degree of Master in Information Technology-Information Networks, has presented his dissertation of the following title (**AN ENHANCED LZW COMPERSSION ALGORITHM TO IMPROVE THE PERFORMANCE OF WIRELESS SENSOR NETWORK**) as it appears on the title page and front cover of the dissertation that the said dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on (23 / 10 / 2022) in our opinion, it is adequate with (Very Good) standing as a thesis.

Signature:
Name: Mahdi Abd Salman
Title: Asst. Prof.
Date: / 11 / 2022
(Chairman)

Signature:
Name: Alaa Taima Abdul Kazem
Title: Asst. Prof.
Date: / 11 / 2022
(Member)

Signature:
Name: Aladdin Abbas Abdel Hassan
Title: Lecturer
Date: / 11 / 2022
(Member)

Signature:
Name: Saif Mahmoud Khalaf Alalak
Title: Asst. Prof.
Date: / 11 / 2022
(Member and Supervisor)

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature:
Name: Dr. Hussein Atiya Lafta
Title: Professor
Date: / 11 / 2022
(Dean of Collage of Information Technology)

Dedication

To whom I will never pay for their love and care

My Dear father (Mazin Bahlool Saleh)

My lovely mother (Zainab Kareem Jader)

My Husband (Ali Ayyed)

To My sisters and brothers

To My Friends for Their Help and Support

I Sincerely Dedicate My Thesis

Acknowledgements

First of all, I would like to express my deepest thanks to "ALLAH” the most gracious and most merciful, for inspiring me with strength and patience to perform this work.

Special thanks to my family for their prayed and support, to my supervisor prof. DR.Saif Alalak for his high moral and his scientific encouragement for me through my work.

Sincere thanks to my friends for their support throughout the process of researching and writing this thesis.

Abstract

Wireless Sensor Network has many limitations including network performance. These limitations also reduce overall system throughput. The difficulty of replacing the power supply when it runs out is prompting researchers to develop schemes to reduce WSN uptime. Because the uncompressed data transmission takes a lot of time, so the power supply runs out quickly. The research question is how to reduce the amount of data transmitted to improve network performance?

The work will include a dictionary compression method, Lempel Ziv Welch (LZW). One problem with the dictionary method is that the token size is fixed. Furthermore, for example, two bytes are wasted when one byte needs to represent a token (since the token size must be represented by three bytes). The LZW dictionary method is not very useful with little data, because it loses many bytes when storing small-sized symbols.

The proposed work suggests the use of a dynamic size symbol where symbols are categorized according to their size into (one byte, two byte, three byte).

The main idea of the proposed work is to increase the repetitive data to increase the compression ratio.

To increase redundant data, the work suggests keeping the increased amount of read data rather than keeping the whole real data. Because climate reading data changes very slowly, so that the amount of change is frequent.

The work was carried out experimentally. Three experiments were performed. Each experiment includes four tests. The first experiment is performed on data collected in one hour. The second experiment runs on data collected in two hours. The third experiment is performed on data collected at three hours.

The tests for each experiment are, Data compression by LZW test, Data compression by reducing the value of data (DVM) by LZW test, Data compression using the Variable Index (VI)-LZW test, and finally Data compression using DVM and VI-LZW test.

Experimental Result showed that DVM and VI-LZW provide better data compression as 13,392 to 814 bytes were compressed, better compression ratio of 6.1, data transfer time of 1,409 miliseconds and system throghput of 0.591 and 93.92% saving percentage.

Table of Contents

List of Figures

List of Tables

List of Algorithm

List of Abbreviations

Abbreviation	**Description**
ADC	*Analog-to-Digital Converter*
BS	*Base-Station*
BPCS	*Bus Priority Control System*
DAC	*Digital-to-Analog Converter*
DVM	*Data Value Minimizing*
DSSS	*Direct Sequence Spread Spectrum*
FHSS	*Frequency Hopping Spread Spectrum*
GPS	*Global Positioning System*
GMs	*General Motors*
LZW	*Lempel Zev Welch*
MEMS	*Micro-electro-mechanical-systems*
MEMO	*Multimedia Environment for Mobiles*
PANs	*Personal Area Networks*
PDAs	*Personal Digital Assistants*
TL	*Turbo-Like*
VI- LZW	*Variable Index LZW*

CHAPTER ONE

INTRODUCTION

1.1 Overview

Simply a WSN is a network of sensors that communicates with each other in different ways. They are sensing , processing , and sending data to the base station and receiving it are all things that must be taken into account with regard to network performance, WSN consist of a group of devices called motes. Each mote consists of a processor, internal memory, RAM and ROM, a wireless transceiver, a power source, and one or more sensors. And perhaps in some of them the Global Positioning System (GPS) because determining the location is important for receiving data. The processor usually runs an operating system that supports network protocols [1].

In WSN, time is consumed by receiving and sending as well as processing data, while the time required in data processing is much less than that required in data transmission, so reducing sensor working hours has become critical. There are many techniques that lead to preserving the life of the network through reducing the amount of data sent and received (data compression). [2]

The main challenge today in this field is to improve performance of the sensor network. For this reason, This work, aims to develop an algorithm for data compression.

Dictionary based compression algorithms are based on a dictionary instead of a statistical model. A dictionary is a set of possible words of a language, and is stored in a table like structure and used the indexes of entries to represent larger and repeating dictionary words. Dictionary is created dynamically is created dynamically in the compression process and no need to transfer it with the encoded message for decompressing. In the decompression process, the same dictionary is created dynamically. Therefore, this algorithm is an adaptive compression algorithm.[3]

The Lempel Ziv Walch (LZW) algorithm is a dictionary algorithm. Where a dictionary is used in the process of storing words and in the process of compressing them, indexes are used. The LZW algorithm is a lossless algorithm. [4]

1.2 Thesis Problem

WSN has many limitations including network performance. These limitations also reduce system throughput in general. The difficulty of replacing the power supply when it runs out prompts researchers to develop schemes to reduce the uptime of the WSN because it takes a lot of time to transmit uncompressed data, so the power source runs out quickly. The research question is how to reduce the amount of data transmitted to improve network performance? .

One problem with the dictionary method is that the token size is fixed. Furthermore, for example, 2 bytes are wasted when 1 byte needs to represent a token (since the token size must be represented in 3 bytes). The dictionary method is not very useful with little data, because it loses many bytes when storing small-sized tokens.

1.3 Thesis Objectives

The main objective of the thesis is to develop an algorithm to improve system performance, which is one of the dictionary algorithms. This aim can be reached even more with the following thesis objectives:

1- Improving the work of sensors by reducing the time required to send data.

.

2- Improving node throughput in WSN through transmitting less size data through data compression. .

3- Improving system throughput of WSN when nodes transmit messages fast.
.

4- Enable the LZW algorithm to compress small data by using dynamic token size.

1.4 Related Studies

Suha et al. [2] he proposed a set of techniques to save energy in a network (WSN) of DRCT The DRCT includes two compression stages, a loss SAX Quantization stage that reduces dynamic range of the sensor data readings, after that a lossless LZW compression to compress the output of the loss quantization. OMNeT++ simulator along with a real sensory data gathered at Intel Lab is used to show the performance of the proposed method.

S. Pushpalatha, and K.S Shivaprakasha [3] explain and compare the different compression methods to improve the performance of the wireless sensor network, where efforts are focused on using algorithms to compress and aggregate data using different network topologies to reduce the amount of data transmitted using different compression techniques. In this paper there is no specific winner and according to the type of application the algorithm is chosen. .

Kodituwakku SR, and Amarasinghe US [4] they propose and compare a set of lossless compression algorithms and test the performance of lossless compression algorithms, run-length encryption algorithm, Hoffman encryption algorithm, Shannon Fano algorithm, adaptive Hoffman encryption algorithm, cipher algorithm, and LZW algorithm implemented and tested using a set of text files. Performance is evaluated by calculating compression ratio, compression time, compression factor, saving ratio and entropy. The algorithms are all good except for the Run length encoding algorithm. Also, the LZW algorithm is not good in the case of large files, as it needs very huge dictionaries for the decompression and compression process. .

S. Shanmugasundaram and R. Lourdusamy [5] there are lot of data compression algorithms which are available to compress files of different formats. This paper provides a survey of different basic lossless data compression algorithms. Experimental results and comparisons of the lossless compression algorithms using Statistical compression techniques and Dictionary based compression techniques were performed on text data. Lossy algorithms achieve better compression effectiveness than lossless algorithms, but lossy compression is limited to audio, images, and video, where some loss is acceptable. The question of the better technique of the two, “lossless” or “lossy” is pointless as each has its own uses with lossless techniques better in some cases and lossy technique better in others..

Mamta Rani and Vikram Singh [6] a review of a group of lossless compression algorithms (bit reduction algorithm, Huffman encoding, run length encoding, Shannon Fano encoding, arithmetic encoding, LZW, Burrows-Wheeler transform), on English words in terms of compression ratio and saving percentage where different compression levels appeared. Huffman and Shannon Fano's coding are very powerful over LZW.

S. Ganesh and N. Shubha G [7] propose Packet level data compression algorithm. Using this algorithm a better compression ratio has been achieved when compared to the previously proposed algorithm. This paper also discusses various standard compression Algorithms. (Delta encoding, Huffman coding, Run length encoding and Arithmetic coding).

Mo yuanbin et al. [8] explain a new algorithm based on adaptive Huffman code. Temperature data is collected from the real world by means of sensor nodes where the temperature change is slow. The principle of the Adaptive Huffman code and the detailed procedure of the new algorithm are introduced. Unlike the predecessors to encode the data as a whole, the algorithm encodes the elementary characters in the difference value respectively. Based on the algorithm, compression experiments are carried out with Ambient Temperature Data, and the compression properties under the precision of 0.1, 0.01 and the mixed precision are analyzed. The results show that the algorithm can achieve the high compression ratio at the precision 0.1, and a little lower compression ratio at 0.01. On the condition of mixed precision the compression properties are also between the two of them.

Cui and W. [9] in this paper, a new LZW compression algorithm increases throughput and improves compression ratio. That partitions conventional single large dictionary into a dictionary set that consists of several small address space dictionaries. As doing so the dictionary set not only has small lookup time but also can operate in parallel.Simulation results show that the proposed algorithm has better compression ratio for image data than conventional LZW algorithm and DLZW (dynamic LZW) algorithm, has competitive performance for text data with DLZW algorithm.

Simrandeep kaur and V.Sulochana Verma [10] He proposed an improved scheme for compressed data by using fewer bits per symbol in the dictionary than its ASCII.In order to get better compression rate, the proposed dictionary based LZW algorithm can replace their codes with 5 bits instead of 7 bits ASCII code. LZW algorithm is evaluated by finite state machine technique. Accurate simulation results are obtained using Xilinx tools which show an improvement in lossless data compression scheme by reducing storage space to 60.25% and increasing the compression rate by 30.3%.

Table 1.1 Summary of Related Works

Year	Algorithm	Description	Result
2020	LZW & SAX Quantization	• Suggest a set of of techniques to save energy in a network (WSN) DRCT The DRCT includes two compression stages. • A loss SAX Quantization stage that reduces dynamic range of the sensor data readings • After that a lossless LZW compression to compress the output of the loss quantization.	The simulation experiments illustrate that the proposed DRCT technique provides a better performance than the other techniques
2020	Bit reduction algorithm, Huffman encoding, Run length encoding, Shannon Fano encoding, Arithmetic encoding, LZW, Burrows-Wheeler transform	• Explain and compare the different compression methods to improve the performance of the wireless sensor network. • Where efforts are focused on using algorithms to compress and aggregate data using different network topologies to reduce the amount of data transmitted using different compression techniques.	In this paper there is no specific winner and according to the type of application the algorithm is chosen.
2016	Bit reduction algorithm, Huffman encoding, Run length encoding, Shannon Fano encoding, Arithmetic encoding, LZW, Burrows-Wheeler transform	• A review of a group of lossless compression algorithms (bit reduction algorithm, Huffman encoding, run length encoding, Shannon Fano encoding, arithmetic encoding, LZW, Burrows-Wheeler transform). • Use English words in terms of compression ratio and saving percentage where different compression levels appeared.	Huffman and Shannon Fano's coding are very powerful over LZW.

Year	Algorithm	Description	Result
2015	Delta encoding, Huffman coding, Run length encoding and Arithmetic coding	• Propose Packet level data compression algorithm.	Using this algorithm a better compression ratio has been achieved when compared
2012	LZW algorithm	• He proposed an improved scheme for compressed data by using fewer bits per symbol in the dictionary than its ASCII. • In order to get better compression rate, the proposed dictionary based LZW algorithm can replace their codes with 5 bits instead of 7 bits ASCII code.	show an improvement in lossless data compression scheme by reducing storage space to 60.25% and increasing the compression rate by 30.3%.
2011	Shannon-Fano Coding, Huffman coding, Adaptive Huffman coding, Run Length Encoding and Arithmetic coding are considered. Lempel Ziv(LZ77, LZSS, LZH and LZB, LZ78, LZW and LZFG)	• There are lot of data compression algorithms which are available to compress files of different formats. • Lossy algorithms achieve better compression algorithms	lossy effectiveness than lossless compression is limited to audio, images, and video, where some loss is acceptable.
2011	Huffman code	• Explain a new algorithm based on adaptive Huffman code. • Temperature data is collected from the real world by means of sensor nodes where the temperature change is slow.	The results show that the algorithm can achieve the high compression ratio at the precision 0.1, and a little lower compression ratio at 0.01.

Year	Algorithm	Description	Result
2010	Run-length encryption algorithm, Hoffman encryption algorithm, Shannon Fano algorithm, adaptive Hoffman encryption algorithm, Cipher algorithm, and LZW	• They propose and compare a set of lossless compression algorithms and test the performance of lossless compression algorithms, implemented and tested using a set of text files.	The algorithms are all good except for the Run length encoding algorithm. Also, the LZW algorithm is not good in the case of large files, as it needs very huge dictionaries for the decompression and compression process.
2007	LZW algorithm, DLZW	• New LZW compression algorithm increases throughput and improves compression ratio. • That partitions conventional single large dictionary into a dictionary set that consists of several small address space dictionaries.	results show that the proposed algorithm has better compression ratio for image data than conventional LZW algorithm and DLZW (dynamic LZW) algorithm, has competitive performance for text data with DLZW algorithm.

1.5 Thsis contributions

The contributions made by the thesis are:

1- The data has been preprocessed so that the first value is fixed and the difference between the current and previous value will be the new value, so representing the new value will need fewer bits.

2- Make the token number of the compression algorithm dynamic by classifying tokens into five classes (A, B, C, D, and E), where the size of each token is 1, 2, 3, 4, and 5 bytes respectively.

1.6 Thesis Structure

In the second chapter, thesis provides an explanation of the most important types of WSN and the most important applications used in their field, as well as the most important types of topology, WSN architecture, its component layers, and finally data compression (LZW) in particular. In the third chapter idea of the proposed work is explained. The fourth chapter includes research method, results and discussed. In the fifth chapter the conclusions and future work are written.

CHAPTER TWO

THEORETICAL BACKGROUND

2.1 Introduction

A WSN is simply a network of wireless sensors that communicate with each other in various wireless ways. WSN has done a wide range of research for long-term physical environmental monitoring where network performance is the main concern, so the effort is focused on reducing the amount of data and thus reducing network uptime. Sensor count, data processing, sending and receiving to the base-station is all things to consider in terms of network performance. In this chapter, we have mentioned an overview of WSN and WSN applications such as technologies, layers, and architecture. Moreover, we mentioned the Zigbee protocol and the specifications of raspberry pi. . . .

2.2 Wireless Sensor Network

It is a group of devices called nodes that collect information from the environment and deliver it to a node through wireless links. Maybe this group data passes several hops until it reaches the sink. It can connect to another network through the Internet, and these nodes can be mobile or fixed, and they can contain a GPS system to know their location. This is the traditional type of WSNs with a single, as it suffers from difficulty in expanding and increasing the size of the network and thus increasing the volume of data, which this leads to difficulty in routing, that is why multiple sink networks are used in which it is very easy to add new and scalable nodes [11]. With a wide range of consumer and military applications, WSNs are becoming a new kind of wireless network. A WSN is a wireless network of sensor devices that are used to monitor physical or environmental factors in an autonomous manner. In a WSN, sensors are linked in a network and interact with one another to share data and information. All the data from the environment is sent to a base-station via these nodes that collect information such as temperature. A wired network or an alert or some other action may be triggered by the latter, depending on the data type and volume being monitored [12]. A WSN is comprised of sensors that collect and transmit data to a base-station or sink node. While WSN is still in its infancy, nodes are transmitting their detected data straight to the base-station. For this reason, the nodes that are the farthest away from the base-station usually perish first, since the power is equal to the square of the distance between the sensor and the base-station[13]. Network protocol designers later adopted multihop routing, which transfers data from a source node through a series of intermediate nodes before arriving at the destination node (the base-station). Because the transmission distance is much less than the direct communication distance between the sender sensor and the base-station, this multihop routing helps to reduce the power consumption of the sender sensor. One further issue has been identified in multihop routing, in which the nodes closest to the base-station die first, the larger number of received packets from other nodes and the increased number of retransmission procedures to the base-station As a result, power consumption for nodes close to the base-station is much greater than for other nodes . Because of their cheap cost and ease of deployment, wireless transmission technologies are starting to play an increasingly significant part in our everyday lives. This is due to the rising capabilities of IT equipment, the advancement of downsizing in electronic systems, and the creation of specialized software and information transfer methods. [14]

2.2.1 Nodes in a WSN

Nodes in a WSN are tiny and affordable, and they may be placed in a fixed position or dynamically distributed to monitor an area. Due to their small stature, they are constrained in many ways. A WSN's primary role is to keep an eye on things. A WSN may observe three types of monitoring: Entity monitoring includes monitoring civil structures like bridges, tunnels, highways, and buildings, as well as monitoring the human body's organs; area monitoring includes monitoring environmental area alarms; and area-entity monitoring includes monitoring vehicles on the highway and tracking the movement of an object. The strength of WSNs is not in the individual sensor nodes, but in the network as a whole, which is made up of many linked nodes [15]. As a result, in order to attain high levels of dependability, WSNs are anticipated to contain a large number of nodes and be self-configuring. It may anticipate a large number of nodes in a WSN since wireless sensor nodes are often affordable. When sensors are connected to one another, a multi-hop protocol is used to provide reliable communication. Base-stations or sinks are places where information and data are halted. In order to disseminate the data collected by the sensor network for further processing, a sink or base-station is often used [16].

2.2.2 Base-stations

Base-stations, are more equipped than normal nodes since they are required to do compound processing. The leader of a cluster should be the primary point of contact for all members of the group. This is followed by a step forward to the next group head, until it reaches its final goal, which is known as the sink or base-station. Sinks are supposed to execute more jobs than typical sensor nodes, which is why they feature more modern processors like PCs/laptops with higher RAM memory, secondary storage, battery, and computing capacity. It's important to keep in mind that the power consumption of sensor networks is highly dependent on the way nodes communicate with one another. As a solution to this issue, the network has included aggregation points, which minimize communication traffic between nodes and save energy. Collection points are nodes that gather data from other nodes nearby, process it, and then forward the filtered data to the next hop. There are "group/cluster heads" for each group of sensor nodes, and each group has a "group head." [17]

When not in use, the nodes may be put into sleep mode and then awoken when required as an alternative energy-saving strategy. When WSNs first emerged, they were primarily used for military purposes [18]. Today, however, they can be seen being used in a wide range of non-military settings such as tracking pollution levels in cities and towns as well as industrial processes. Wireless gadgets improve health care by making patient monitoring less intrusive. Wireless sensors are a cost-effective way to monitor system health and reduce energy consumption in utility applications. For remote monitoring, wireless networks may be used with fixed networks and systems to save wiring costs and enable new types of testing and measuring applications [19] .

2.2.3 Remote monitoring

Remote monitoring can be used for a wide range of purposes, including environmental monitoring, building and structural monitoring, process monitoring, machine monitoring, habitat monitoring, intelligent transportation systems, air traffic control, traffic surveillance, and video surveillance, among others. A radio transceiver, a microcontroller, and a power supply are commonly found in each node in a WSN (typically a battery). According on the network's size, as well as the level of functionality and complexity demanded of each node, sensor nodes may cost anywhere from hundreds of dollars to a quarter of a dollar. Resources such as energy, memory, processing power, and throughput are constrained by sensor node size and cost. Because each sensor in the network enables a multi-hop routing system, sensor networks often comprise a wireless ad-hoc computer network [20].

WSNs include several components that are becoming smaller and less costly, such as sensors, signal converters like A/D and D/A converters, processing units and communication devices. A power supply is also a crucial component of a WSN. This necessitates stringent power consumption requirements since the sensor node has to be dependable and able to operate unattended for lengthy periods of time, which may be years. Power harvesting systems and battery types, as well as electrical design strategies for low power consumption, should all be taken into account while developing a WSN power sources. Small sensor nodes and networks are currently being developed by the manufacturers of these devices [21] . As a further example, commercial off-the-shelf PDAs (personal digital assistants) or pocket computers provide amazing computational capability in a compact size. Sensor nodes may simply be built using these kinds of devices. It is now possible for wireless LANs like IEEE 802.11 to compete with wired networks in terms of performance. The IEEE 802.15 standard, which provides requirements for personal area networks (PANs) and may be used for WSNs, is also available today [22].

Furthermore, developments in semiconductor technology enable us to have greater chip capacity and greater processing capabilities. There is a decrease in energy/bit needs for both computer and communication systems because to this advancement. Micro-electro-mechanical-systems(MEMS) technology is projected to generate more powerful and adaptable sensors in the near future, CMOS bipolar transistors and other IC process sequences are used to create the electronics, whereas MEMO technology merges mechanical components, sensors, actuators, and electronics on a single silicon substrate. The mechanical and electromechanical devices are constructed by adding additional structural layers to the micromechanical parts that are well-suited for micromachining [23].

2.3 Application of WSNs

Today's WSN applications are either in use or may be in their early stages of development. They are categorized according to their uses into six categories as show in figure 2.1 Health, Military, Plants, Animals, Environmental, Industrial, and Urban. [24]

It is important to note that the application affects the form of technology that is chosen for use, so the most important technologies will be discussed in the next section. [11]

Figure 2.1 Categories of Applications of WSNs. [24]

2.3.1 Military Applications

The base field is not the first area in human use for WSNs. Smart dust [25] is one of the most important inventions that were reached in the early nineties in the development of sensor nodes that spy despite their small size. It is essential that the network be self-organizing, resilient, and secure in the military scenario as show in figure 2.2) [26].

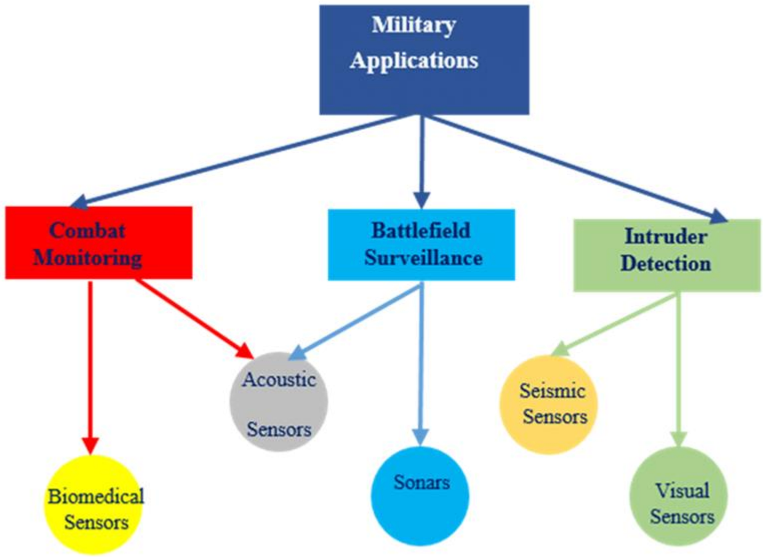

Figure 2.2 The Military Applications of WSNs [24]

2.3.2 Health Application

WSNs use medical sensors in hospitals or at home to monitor patients. One of the important applications of WSNs in the health field is the wearable patient monitor that allows monitoring of the patient's vital condition from measuring blood pressure, heart rate, home assistance systems and patient monitoring systems in the hospital [24]. As show in figure 2.3 the most important vital applications in the Healthy field.

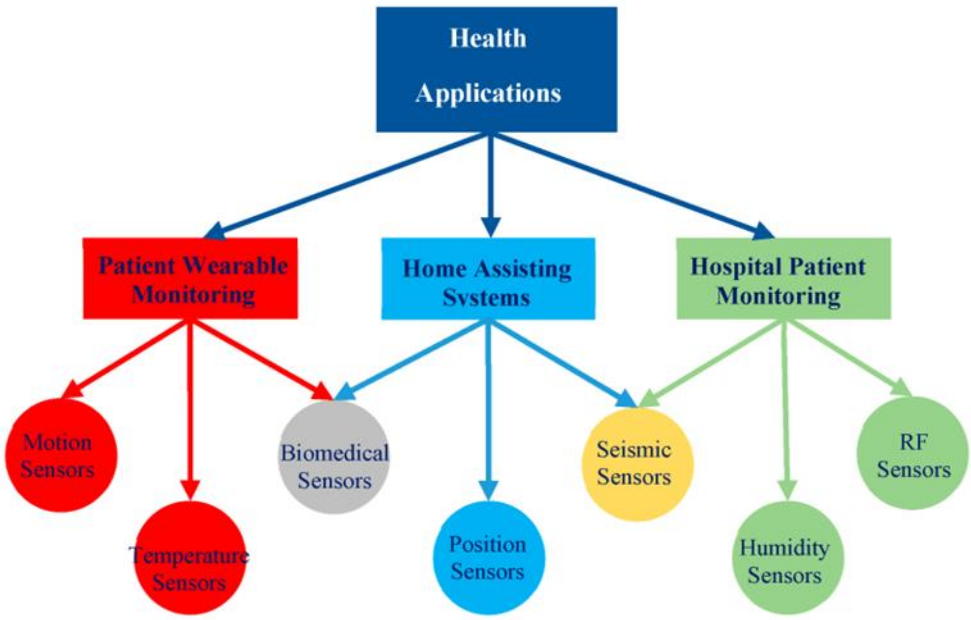

Figure 2.3 Health Applications of WSNs. [24]

2.3.3 Environmental Applications

WSNs may also be used to monitor the environment as show in figure 2.4. In such cases, a large number of low-cost sensor nodes that can be quickly spread around the area is necessary. Examples of WSN applications include forest fire detection, flood detection, soil moisture detection, microclimate mapping and solar radiation mapping, as well as river monitoring and forecasting. Soil moisture measurements are taken more often during rainy periods and less frequently during dry periods, allowing the system to identify itself. This method aids in extending the useful life of the system [27].

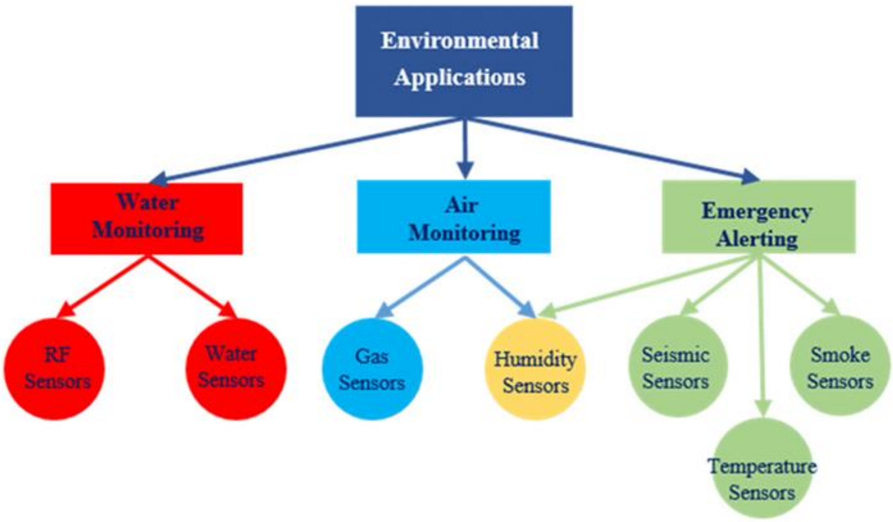

Figure 2.4 Environmental Applications in WSN. [24]

2.3.4 Flora and Fauna Applications

Monitoring plants and animals is as important to every country as monitoring crops and livestock [24]. As show in figure 2.5 the most important vital applications in the flora and fauna applications of WSNs field.

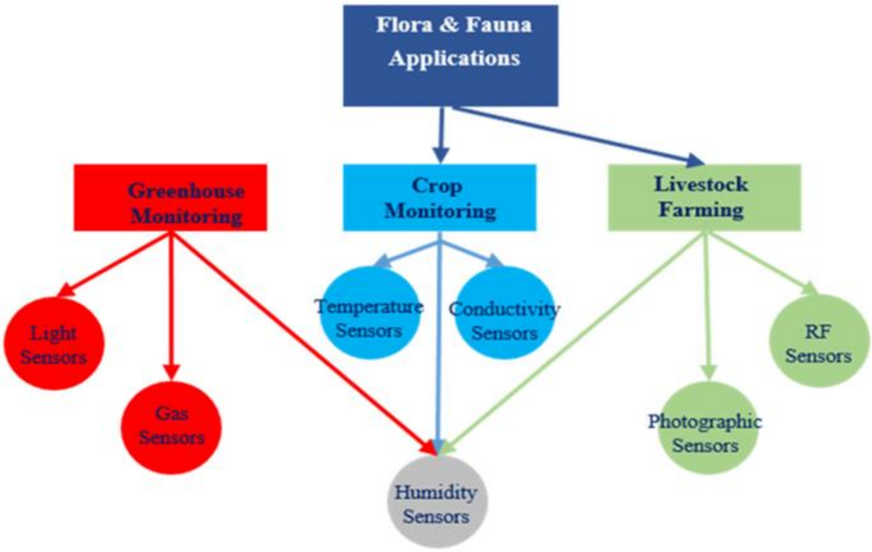

Figure 2.5 Flora and Fauna Applications of WSN. [24]

2.3.5 Industrial Applications

WSN networks are used in many areas of industry, for example, monitoring the health of machines remotely by measuring pressure and temperature; General Motors (GM) is one of the companies that used these sensors to monitor machinery and manufacturing equipment. [28]

2.3.6 Urban Applications

WSNs provide this type of applications such as smart cities, smart homes, and transportation systems [24]. As show in figure 2.6 the most important vital applications in the urban applications of WSNs field.

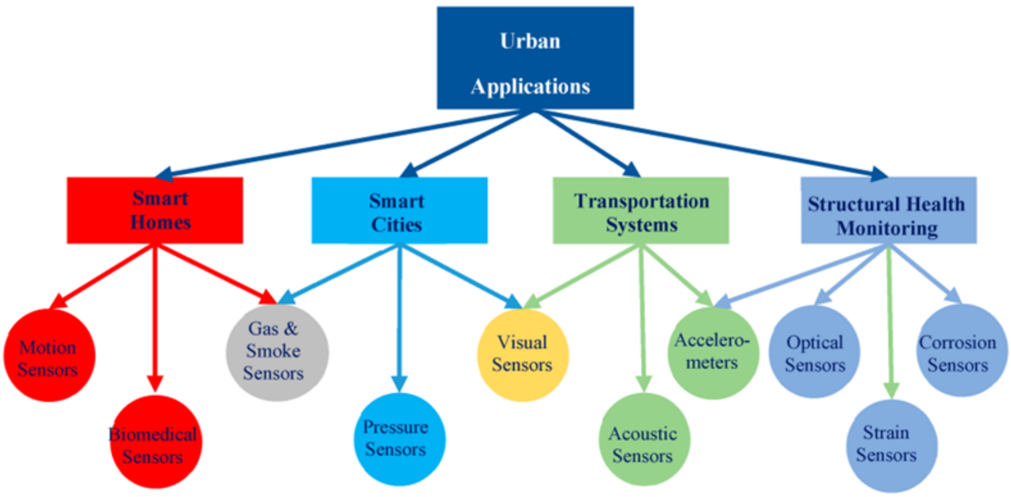

Figure 2.6 Urban Applications in WSNs. [24]

2.4 WSN Topologies

Traditional network topologies as show in figure 2.7 have been reimagined by the creation and implementation of WSNs. It is possible to build wireless sensor networks in a variety of ways(Bus Topology, Tree Topology, Star Topology, Ring Topology, Mesh Topology, Circular Topology, Grid Topology)[29] .

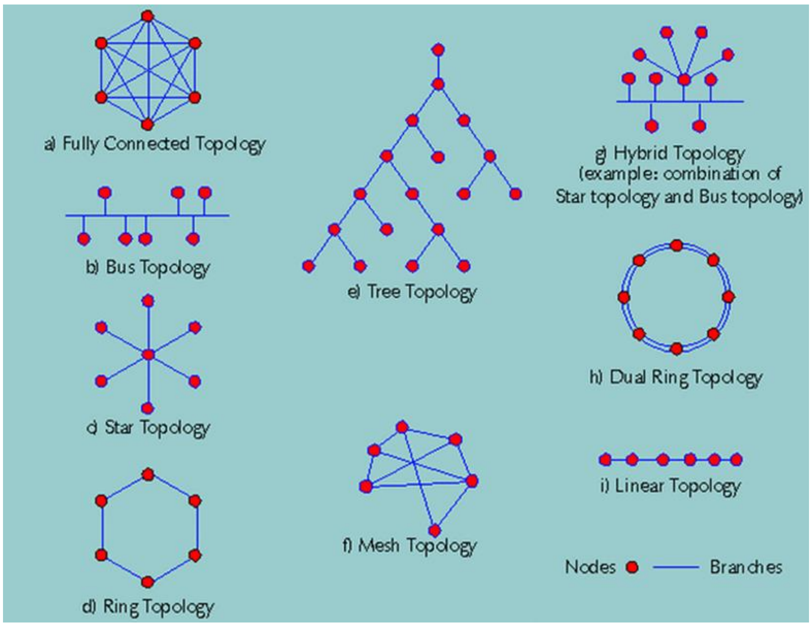

Figure 2.7 WSN Topologies. [30]

2.5 WSN Layers

The architecture of WSNs follows the common OSI model. In a WSN, there are five layers: the application layer, the transport layer, the network layer, the data link mode and the physical layer. And also three cross layers: the energy management plane, the mobility management plane, and the task management plane, as shown in as show in figure 2.8. [31, 32]

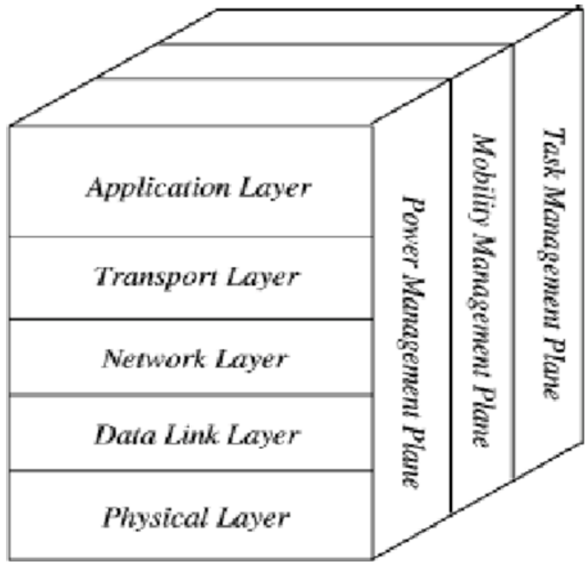

Figure 2.8 WSN Layers. [32]

3.5.1 Cross layers [31, 32]

These layers work to increase the efficiency of the overall sensor network:

- Power management plane: Manages the power used in nodes for processing, sensing, and communication.
- Mobility management plane: It is responsible for the movement of the sensor and also tracking the neighbors.
- Task management plane: It detects running and discontinued nodes and also organizes sensing tasks.

3.5.2 OSI layers

•Transport layer: One of the most important things this layer does is reliability and congestion avoidance, there are protocols designed to provide this functionality (TCP, UDP, SCTP, DCCP, SPX). This layer is important when the system connects to another network, where it receives data from the upper layers and divides it into small units and then sends it to the network layer, and to make sure that all the pieces have been received at the other end, it detects and recovers the loss using (ACK, NACK, and Sequence number). UDP is used to communicate from the sink to the node to avoid overhead and also the limited memory of the sensor, TCP or UDP from the node to the sink. [31, 32]

•Network layer: Routing is the main function of this layer. There are many challenges in this layer depending on the application, the most important of these challenges are limited power and memory, as well as buffers, and the sensor does not have a global identifier (ID). The routing protocol assigns reliable and redundant paths according to a scale called a metric. There are many protocols (flat routing such as direct diffusion and hierarchal routing e.g., LEACH or can be divided into time driven, event driven and query driven). [31,32]

- Data link layer: It is responsible for error detection and correction, and for multiplexing data transmissions, data frame detection, MAC, and ensure reliability of point–point or point– multipoint. [31,32]
- Physical layer: It provides a physical interface to transfer bits through a physical medium, Responsible for generating, carrier frequencies, frequency selection, signal detection, signals modulation and data encryption. [31,32]
- Application layer: Responsible for traffic management and provide software for different applications that translate the data in an understandable form or send queries to obtain certain information. [31,32]

2.6 WSN architecture

Sensor nodes and the base station must interact through networking in order to monitor the activity of sensors. WSN's second component plays a crucial part in this network design. A large number of sensor nodes are placed in a large region in order to jointly monitor a physical environment. As a result, the networking of these sensor nodes is just as critical. Sensor nodes in a WSN communicate with each other and with a Base-Station. Wireless communication is used in this case via WSN. Wireless sensor networks are the name of this kind of network as show in figure 2.9 [33].

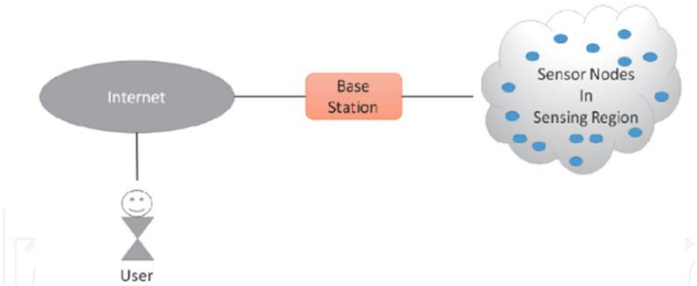

Figure 2.9 WSN Architecture[33]

There are two major facets to this network's design. An example of one is known as a single-hop network and another as a multi-hop network Single-hop network design is as show in figure 2.10. Each sensor node is linked to the base station in a single-hop network design. It may also be used for long-distance transmission.

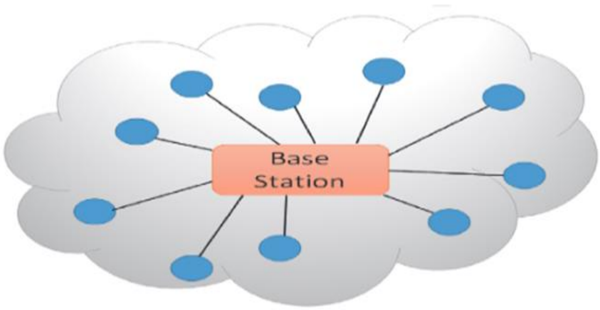

Figure 2.10 Single-hope Network Designs[34]

Long-distance transmission necessitates a large increase in energy usage [34]. As a result, data gathering and calculation are affected. Although a single-hop network's power consumption is considerable, a multi-hop network's power consumption is much lower. Because of this, multi-hop network design is often utilized to reduce power consumption. The multi-hop network design is shown in figure 2.11. In a multi-hop network design, the sensor node and the base station each have an intermediary node, which decreases the burden on the one single connection. Intermediate nodes may be used in this scenario. As a result, this network design is more efficient than a single-hop network [34].

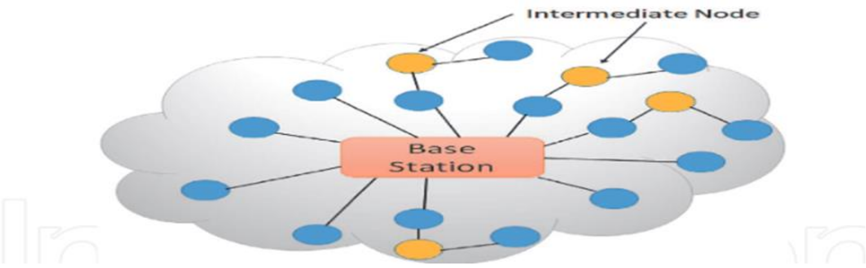

Figure 2.11 Multi-hop Network Designs[34]

There are two ways to construct a multi-hop network architecture. Flat and hierarchical network architectures are available. The data may then be sent to the base station via the cluster heads. Sensors and analog-to-digital converters are the most common components of a sensing unit. For example, sensors may monitor ambient parameters like temperature, light, and sound. Analog signals from the sensor are converted to digital signals by an analog-to-digital converter (ADC) before being sent into a computer processing unit, which processes the data. Sensors should be tiny in order to provide the best results [35]. .

The sensor node gets its intelligence mostly from the processing unit. Microprocessors are used to run communication protocols, operate sensors, and perform signal processing on sensor data collected from several sensors. Sleep, idle, off, and active are the four basic processing states of a microprocessor, and each has its own subcategory. In idle mode, the CPU is still active but other peripherals are active, but in sleep mode, most internal devices and the CPU are switched on and can only be activated by some external event [36]. .

When running in idle mode, the transceiver consumes a significant amount of power, which is almost equivalent to the amount of power used when working in reception mode. When the radio is not sending or receiving data, shutting it down fully rather than setting it in an idle state is necessary because of the significant power consumption. WSN's sleep mode is one of the most energy-efficient of the system's four phases [37]. .

The battery powers the sensor node as a whole. It has a significant impact on the sensor node's lifespan. Keeping an eye on how much power is being pulled from a battery is essential. Because of this, the battery capacity of the sensor node stays constrained. For a network with thousands of embedded nodes, battery replacement is not an option. As a result, the lifespan of a sensor network is heavily influenced by its energy usage [38].

2.7 Data Compression

The widespread use of computing and the Internet led to an explosion in the amount of data transmitted, which led to the emergence of the need to compress data. The compression ratio is an important feature in data compression, as it reduces the file size, making it easier to store and transfer over the Internet. There are two types of compression, "lossless" and "lossy" for data compression. Lossless data compression is used when it is necessary to retrieve data after compression exactly identical to the data before compression, such as text files, where the loss of one character leads to making the data misleading, and also video, image and audio files must be lossless. While compression with loss is considered not to be necessary for the stored data to be perfect. Each method has its uses, Lossless compression is better in some cases, and in other cases lossly compression is better. In order to obtain a better result, the two methods are used, as well as the contents of the file in terms of repetition in the data, which greatly affects the compression ratio [42]. Sensor nodes in a WSN use the most energy communicating data. A direct transfer of the original measured data isn't feasible because to the restricted power, computational power, storage space, and communication bandwidth available on nodes in general. As it turns out, exchanging computation for communication may help conserve energy. One study found that executing 3000,000 instructions consumes the same amount of energy as delivering a thousand bits over a distance of 100 meters via radio. Therefore, compressing data before to transmission is an efficient technique to maximize node resources and decrease network energy usage. Lossy and lossless compression is two types of data compression. Sensor node resources are a major determinant of WSN data compression. For this reason, WSN-specific data compression algorithms must concentrate on low computation complexity and great energy efficiency [39]. .

Data compression for WSN has been the subject of several studies in recent years. As one of the hot issues nowadays, wavelet-based data compression is a lossy compression. To prepare for inquiries, important measurable data must be preserved in the area without any loss. The raw data may need to be sent to a higher level node or a sink node for WSN real-time monitoring. In order to fulfill the storage and wireless transmission needs of WSN, energy-efficient lossless compression algorithms are required. Huffman coding, LZW coding, and run-length coding are all lossless data compression methods that may be used in WSNs. In terms of coding systems, the LZW algorithm is the most notable one. There is no need for the vocabulary to be delivered with the code at the same time, but rather, it is progressively created at the decoder. There are no assumptions made about the qualities of the source, which makes it flexible. Even though the technique is simple to put into practice and has excellent latency, memory, and compression effects for WSN, there is room for improvement in terms of coding efficiency [40].

2.7.1 Data Compression Techniques

Can be categorized as show in figure 2.12 to data quality, coding schemes, data type and application suitability [42].

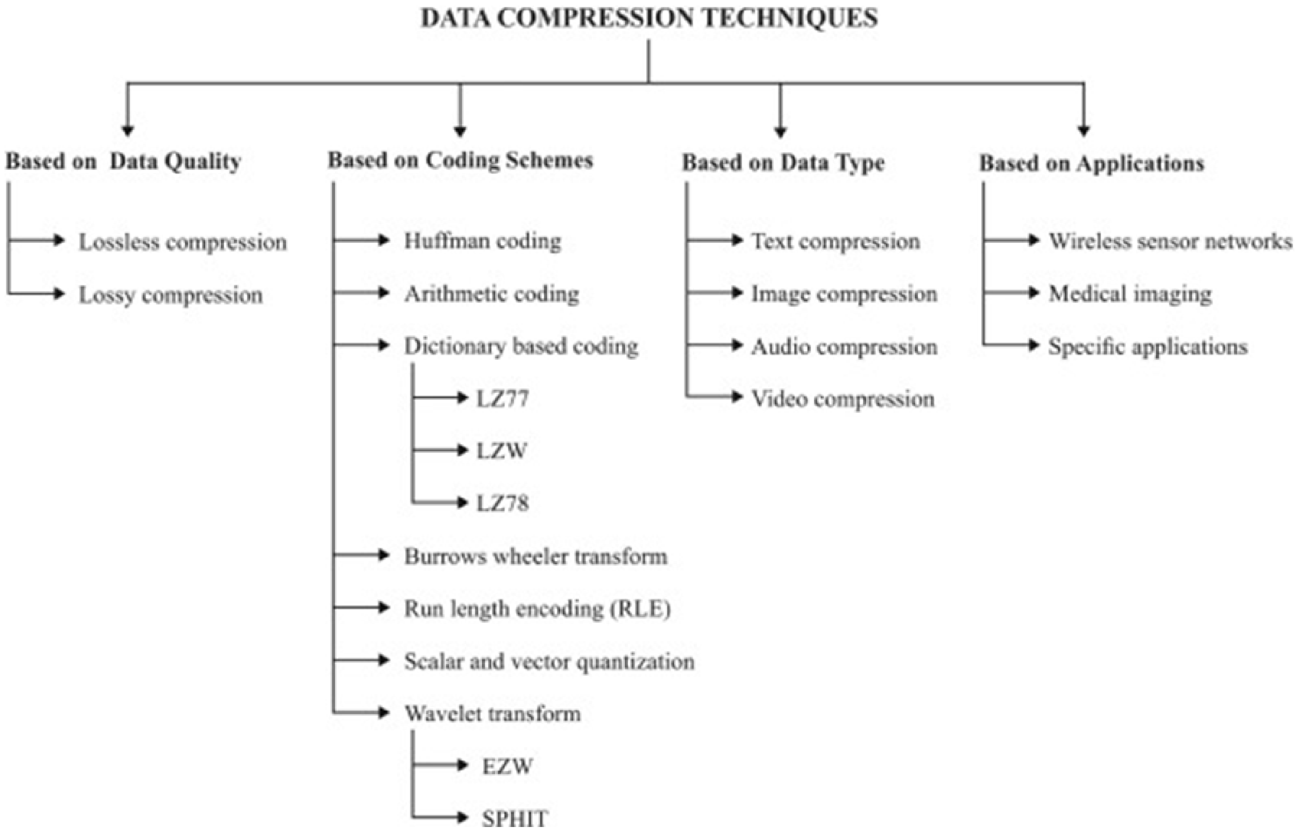

Figure 2.12 Data Compression Techniques. [42]

I. Based on Data Quality Techniques

Generally, data compression techniques affect data quality based on application criteria. For example, in the case of using data compression techniques for the purposes of messaging or browsing the Internet, the quality of the data is not taken into account, but in the case of text compression, the quality of the data is very important, as losing one character leads to a complete change of meaning. Data compression techniques can be divided into lossless and lossy compression. Lossless data compression indicates that information cannot be lost, meaning that the data after compression must be exactly the same as the data after compression. It is used in text, medical imaging, law forensics, military imagery, satellite imaging. Whereas it is preferable to use lossy data compression when there is no need in that the data after compression is exactly the same as the data before compression, It is used to compress medical images, hyper spectral images, videos and so on [42].

II. Based on Coding Schemes

In this section, the popular coding techniques such as Huffman coding, Arithmetic coding, LZ coding, Burrows-wheeler transform (BWT) coding, RLE, transform coding, predictive coding, dictionary based methods, fractal compression, Scalar and vector quantization which are considered as the ancestors in the field of Data Comperission [42].

III. Based on The Data Type

Generally, DC has been employed to compress text, image, audio, video and some application specific data. In this section most of the important and recent compression techniques under each datatype and the classification hierarchy as show in figure 2.13 [42].

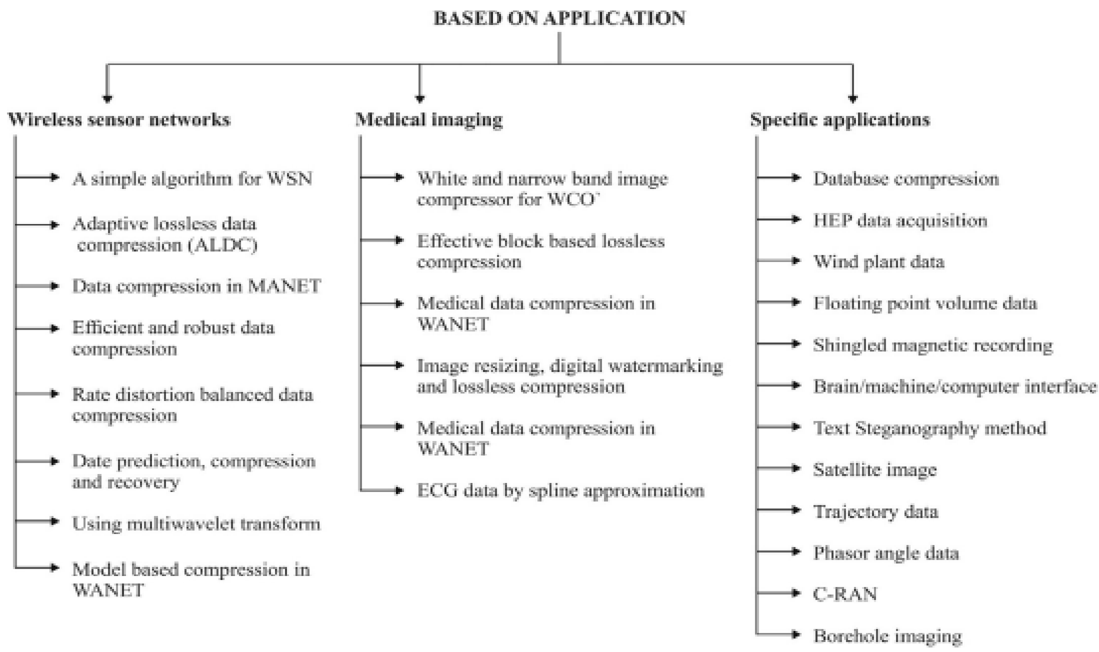

Figure 2.13 Data Compression Techniques based on Application. [42]

IV. Based on Application

Although some of the common compression techniques can be applicable to preferred applications, the concentration has been on the technique rather than the application. However, there are certain techniques where it is impossible to separate the technique from the application . This is due to the fact that several techniques depend upon the nature of data involved in the application. the classification hierarchy as show in figuer 2.14 [42].

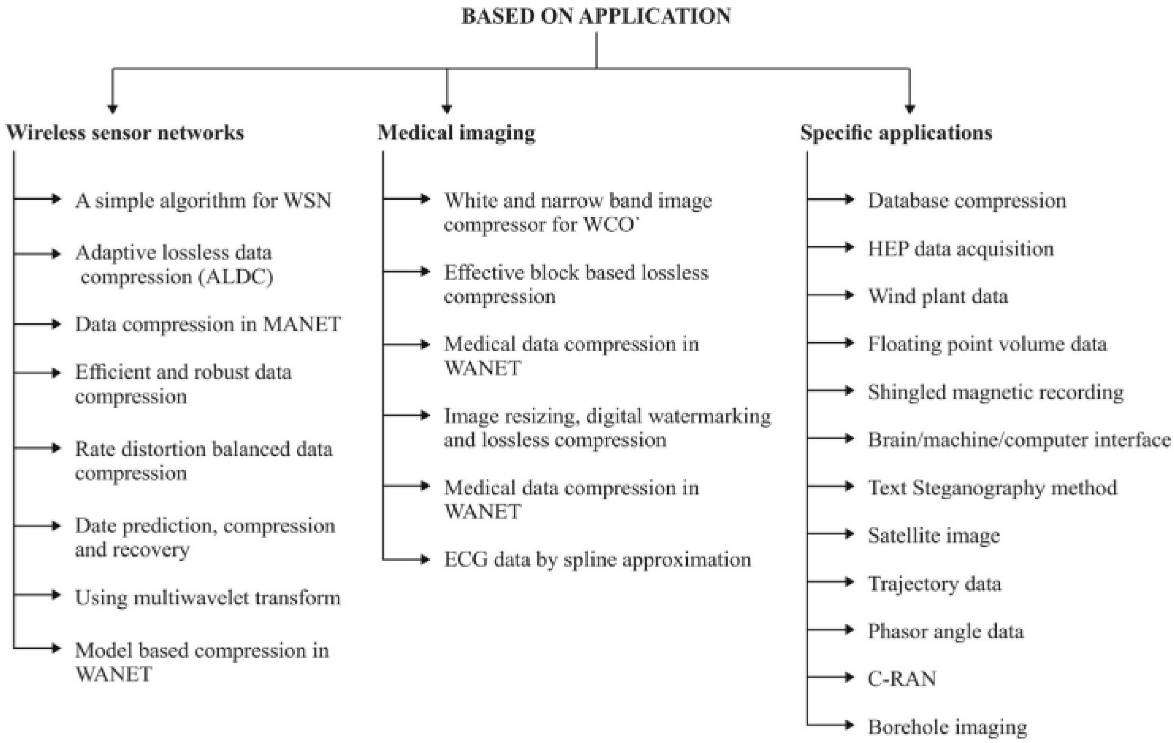

Figure 2.14 Data Compression Techniques based on Application. [42]

2.7.2 LZW Compression

In 1980 Terry Welch invented the LZW algorithm which is one of the most important and popular compression algorithms which is one of the powerful compression techniques that provide high lossless compression efficiency. The LZW algorithm depends on a dictionary that is dynamicly is created. The LZW algorithm is simple as the dictionary is not transmitted during the transmission process and both sender and receiver will get the same dictionary dynamically. In the LZW algorithm, all the individual items must be initially added to the dictionary, which does not work with few data [43]. .

A dictionary lookup-based method, LZW compression uses the prefix string to search up the rest of the string. For example if the dictionary contains the character string OJK, which consists of the character string OJ and a single letter K, then OJ must be there as well. When the character string occurs again during encoding, a number that represents it may be substituted and the number can be output. The dictionary in question isn't pre-built; rather, it is generated on-the-fly from raw data. When the data is decoded, the raw dictionary is retrieved. During the compression and decompression processes, the dictionary may be constructed appropriately and then discarded. In a nutshell, an LZW algorithm adaptively builds a dictionary of non-repeating substrings from the input stream. Each substring is then mapped to a unique output code word. Because of this, it accomplishes the goal of compressing data. Text files should be compressed using the LZW technique in three steps [44]:

- Make a dictionary entry for every single character.
- Get the prefix string w by reading the first character.
- Output the dictionary and exit if the following character K indicates that the file is at its conclusion. Then, if wK is already in the dictionary, then w= wK; if it is not, then w=K. Output the string's matching address if w contains just one character; else, output the string's full address. Observe and Recur (c).

In spite of the ease with which the LZW technique is implemented, the encoding mode has several drawbacks. LZW's drawback is that all single characters need to be entered into the dictionary at the beginning, but they don't need to be encoded or decoded. There are so few terms in the dictionary at this early stage that it does not help much to compression, making it inapplicable to a little quantity of data. The dictionary, on the other hand, has a limited storage capacity, and its efficiency will be reduced if the data is too vast. As a result, the LZW compression technique is inefficient when dealing with huge datasets [45].

While larger substrings occur less often, the dictionary's substring length is unrestricted. It adds to dictionary size and does nothing to enhance compression ratio, especially for material with a low repetition ratio .Only text files can benefit from the LZW method. Sensor nodes embedded with ultralow power microcontrollers like the MSP430F series have limited storing and calculating power and this constraint will put more strain on their storage or calculating capabilities. The encoding process has been improved as a result of the following changes [46]:

- All single characters will be removed from the vocabulary at the outset, allowing the dictionary to be compressed while still being properly decoded.
- The dictionary capacity should be set to the proper value. It is appropriate to store the dictionary in nodes with an address space of two bytes since nodes have a limited amount of memory.
- In order to save memory and ensure excellent performance, it is recommended that substrings in the dictionary be kept to a minimum. The only way to know for sure how long something will be is to put it to the test. Using the difference approach for numerical data reduces the range of possible data duplication and reduces the dictionary size.
- In order to transform non-document files into text files, do preprocessing for each circumstance.

Limitations of LZW [1]:

LZW It has many limitations that greatly affect WSN Which, we will explain in the following points:

- LZW algorithm is only suitable for text files.
- Initially, when creating the dictionary, all individual letters must be placed.

- A fixed token is used to store index of character in dictionary.

This algorithm has two inputs s and ch (see in algorithm 2.1), data is compressed by reading first character(s) and second character(ch), if s + ch is in dictionary, index is taken for compression purpose, but if it doesn't exist, it is added to it new index in the dictionary, This index is compressed with a fixed token as show flowchart in figure 2.15.

Algorithm 2.1 Original LZW

Input : string s ;character ch.

Output: encode s to output file .

```
s ← empty string
while (there is still data to be read)
    ch ← read a character
    if (dictionary contains s+ch)
        s ← s+ch
    else
        encode s to output file
        add s+ch to dictionary
        s ← ch
    end
end
end
return s
```

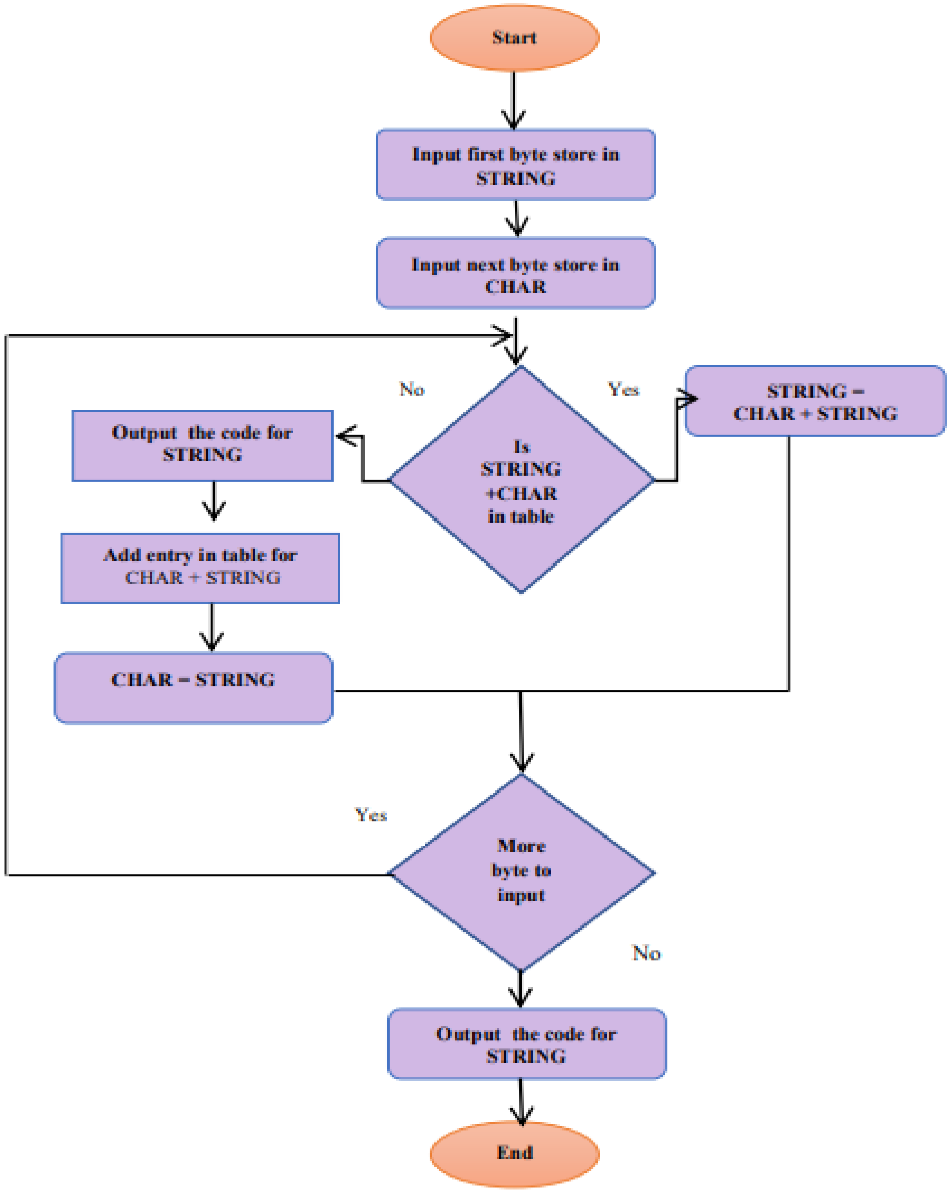

Figure 2.15 Flowchart of the Original Algorithm [1]

2.8 Raspberry pi

It was developed by the Raspberry Pi Foundation (raspberrypi.org) and was first released in 2012. Several generations of it have been produced that can be classified into (The Raspberry Pi A, B and Zero). Each unit consists of a central processing unit (CPU), onchip graphics processing unit (GPU), internal memory, a 5V DC power input, a dedicated place to connect the camera, in addition to the general-purpose I/O ports, Ethernet and wireless

(Bluetooth and WiFi) connectivity are also available. Raspberry with these specifications is a small size computer with lower costs.

There is a new addition to our popular Raspberry Pi family of computers - the Raspberry Pi 4 Model B Compared to the previous-generation Raspberry Pi 3 Model B+, it delivers ground-breaking advances in processing speed, multimedia capability, memory, and networking, while maintaining backwards compatibility and comparable power consumption. When it comes to desktop performance, the Raspberry Pi 4 Model B is on par with entry-level x86 PCs [47]. .

Dual-display support up to 4K through a pair of micro-HDMI ports, hardware video decoding at up to 4K p60, up to 4GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth, USB 3.0, and PoE capabilities are all included in this product's essential characteristics (via a separate PoE HAT add-on) as show in figure 2.16. It's possible to integrate the dual-band wireless LAN and Bluetooth into final products with much less time and expense by achieving modular compliance certification [48].

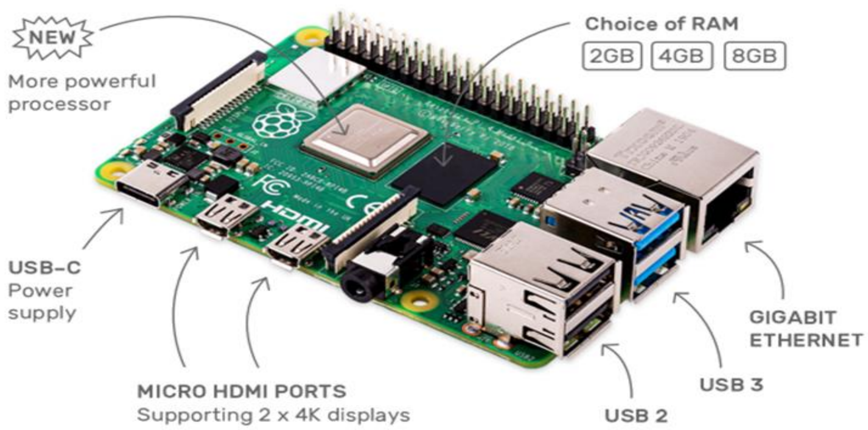

Figure 2.16 Raspberry pi 4 Model B[49]

2.9 Zigbee protocol

From this point of view, engineers began to think of a new technology that would be a little energy consuming and cheap to manufacture as show in figure 2.17. The engineers saw the movement of bees when they met each other. They could perform complicated and difficult tasks with each bee sharing its simple card. They also noted that bees use their own language and dance to get to know each other. From here, the new technical engineers called ZigBee, the bees' language or the language used by the bees to talk to one another, have been invented to do a certain task. As part of the ZigBee Alliance, a wireless communication standard that is both inexpensive and power efficient has been created. Consumer electronics, home and building automation, industrial controls, PC peripherals, medical sensor applications, and toys and games will all use ZigBee-based solutions [50].

Figure 2.17 Zigbee Protocol. [50]

Power required for ZigBee is relatively little. In most circumstances it takes 1mW (or less power). But nonetheless it delivers range up to 150 meters in outside which is done by the technology called direct sequence spread spectrum (DSSS). Also DSSS uses less electricity compared to Frequency Hopping Spread Spectrum (FHSS). It operates in the 868 MHz (Europe), 915 MHz (North America and Australia) and 2.4 GHz (available globally) ISM band with up to 20kbps, 40kbps and 250kbps data rate accordingly . Because these wave bands are distinct from the bands of existing mainstream wireless networks, Wireless Fidelity (Wi-Fi), Bluetooth, Wireless USB etc. Mutual interferences between them will not occur, thus, this assures the system will not disrupt other wireless networks and will not be harmed as well. The IEEE 802.15.4 standard utilizes 64-bit and 16-bit short addresses to allow potentially more than 65,000 nodes per network. ZigBee network can include up to 653356 devices, the distance between ZigBee devices may be up to 50 meters, and each node can relay data to other nodes. This gives capabilities of constructing a very vast network with spans considerable distances [50].

The experiments of the research used Raspberry pi Arduino Sheild Xbee Expansion Hat for zigbee connectivity as (see in figure 2.18).

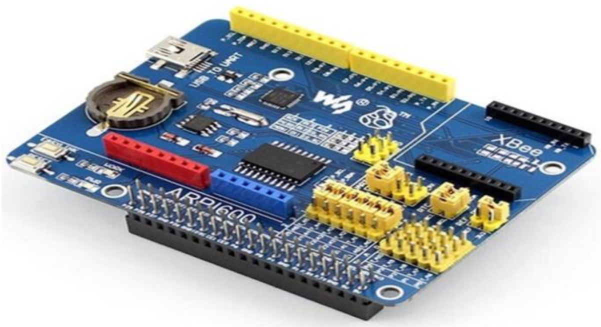

Figure2.18 Raspberry pi Arduino Sheild Xbee Expansion. [52]

2.10 Temperature and Humidity sensors

It is an economical peripheral device produced by D-Robotics UK (www.droboticsonline.com), DHT11 sensor sitting on a breadboard as show in figure 2.19. It can measure relative humidity, which ranges between 20% and 90%, and temperature from 0 to 50 degrees Celsius. [53]

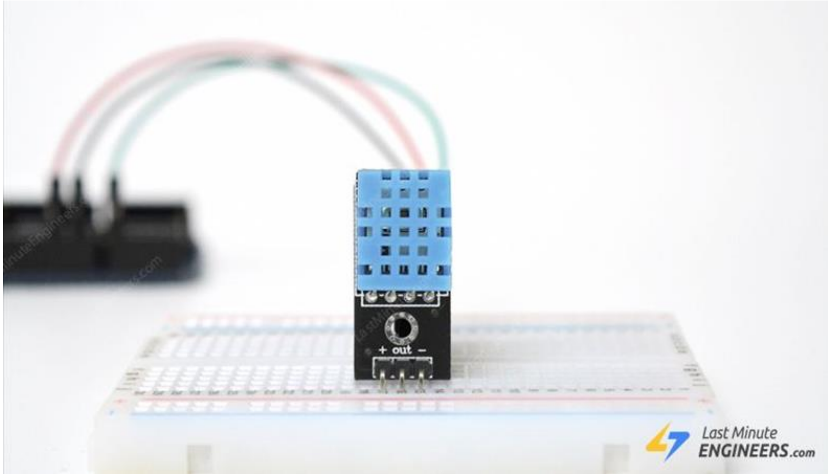

Figure 2.19 DHT11 Sensors. [54]

2.11 Rain Sensors

It is one of the switches used to detect rain as show in figure 2.20. When rain is detected, it issues an alarm. The pin of this sensor is shown below. This sensor includes four pins which include the following: [55]

Pin1 (VCC): It is a 5V DC pin , Pin2 (GND): it is a GND (ground) pin , Pin3 (DO): It is a low/ high output pin, Pin4 (AO): It is an analog output pin

Figure 2.20 Rain Sensors. [56]

2.12 Performance Metrics

Three metrics are computed to evaluate the proposed work compression ratio, Throughput, Saving Percentage.

- **Compression ratio**, first we need to know the "ratio", the ratio is a comparison of two numbers, so the compression ratio is the size of the data before and after compression [4,6].
  Through the following mathematical eq.

$$\textbf{Compression Ratio}= \frac{\text{size after compression}}{\text{size befor compression}} \times 100 \quad \ldots\ldots\ldots..(2\text{-}1)$$

- **Throughput** is also calculated**,** It is the amount of data that the system can process during a certain period of time, so it is possible to measure the speed of data transfer between two systems [4,6].

$$\textbf{Throughput} = \frac{\text{The amount of data transmitted}}{\text{time}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(2\text{-}2)$$

- **Saving Percentage** calculates the shrinkage of the source data as a percentage [4,6].

$$\textbf{Saving percentage} = \frac{\text{size befor compression} - \text{size after compression}}{\text{size befor compression}} \,\%(2\text{-}3)$$

CHAPTER THREE

PROPOSED WORK

3.1 Introduction

This chapter, proposes three data compression algorithms in WSNs with the purpose of reducing the amount of data transmitted. Thus increasing the efficiency of network performance. The proposed method for the algorithms that have been worked on is explained in Section 3.2. As well as pre-processing the data collected by the sensors from reality in Data Value Minimizing (DVM) and compressing it with the original LZW algorithm as shown in Section 3.2.1. In Section 3.2.2. The Variable Index LZW algorithm (VI-LZW) is discussed. As for Section 3.2.3, the data was pre-processed (DVM) and compressed using the Variable Index LZW algorithm.

3.2 Proposed work

Most of the sensed data in WSN is related to the weather, which is including heat, humidity, atmosphere pressure …etc. The data would be represented as integer number. It means it needs one signed byte for binary representation. To address the problem of data representing, a proposed scheme does not deal with the actual value, it computes the difference between previous value and new value. The difference between two values would be less than both of them, so that when representing a new value, it would need less bits. An example to illustrate the idea is demonstrated in example.

Example:

When the temperature of somewhere at time (t1, t2, …, t7) are (25, 26,27, 27, 27, 28, 27) respectively, then the binary representation for degrees is as shown in third column in Table 3.1.

The difference between new and old value staring from original is shown in fourth column, and binary representation is shown in fifth column.

Table 3.1 Example data representation

Time	Data in Decimal	Binary (byte)	Difference	Binary (Signed byte)
t0	0	00000000	0	00000000
t1	25	00011001	25	00011001
t2	26	00011010	+1	00000001
t3	27	00011011	+1	00000001
t4	27	00011011	0	00000000
t5	27	00011011	0	00000000
t6	28	00011100	+1	00000001
t7	27	00011011	-1	11111111

From example, it is possible to say that the difference between original and new value could be represented by less than one byte. To complete the addressing of problem, we propose to use one of the dictionary compressions schemes like LZW (Lempel Ziv Welch) to compress the computed values. It is expected to get a high compression ratio because most of the computed values are with high similarity. To address the problem of fixed size token in LZW, propose to use a dynamic size token. The proposed work, classifies tokens into five classes (A, B, C, D, and E), where the size of each token is 1, 2, 3, 4, and 5 bytes respectively. Each category is differentiated from other basing on the first byte. Furthermore, the left most bits gives token category, as shown in Table 3.2. The Most Significant Bit (MSB) of token belongs to class A is (1). LMBs of token belongs to class B are (0,1). LMBs of token belongs to class C are (0,0,1). LMBs of token belongs to class D are (0,0,0,1). LMBs of token belongs to class E are (0,0,0,0,1).

Table 3.2 LZW token categories

Token Class	Binary Representation	Token Size
A	1,-,-,-,-,-,-	1 byte
B	0,1,-,-,-,-,-,- -,-,-,-,-,-,-,-	2 bytes
C	0,0,1,-,-,-,-,- -,-,-,-,-,-,-,- -,-,-,-,-,-,-,-	3 bytes
D	0,0,0,1,-,-,-,- -,-,-,-,-,-,-,- -,-,-,-,-,-,-,- -,-,-,-,- ,-,-,-	4 bytes
E	0,0,0,0,1,-,-,- -,-,-,-,-,-,-,- -,-,-,-,-,-,-,- -,-,-,-,- ,-,-,- -,-,-,-,-,-,-,-	5 bytes

When the compression ratio is high it means a WSN system uses less messages for transferring more data than original state.

3.2.1 Data Value Minimizing (DVM) & Original LZW Algorithm

In this algorithm, the data is first processed by storing the first value in buffer (nval), and then subtracting the current value (cval) from the first value (fval) for the purpose of increasing the number of frequent iterations and thus compressing them using the original LZW algorithm is more efficient. Two entries are used during the compression process in the algorithm: s and ch, so s is the first value and ch is the second value. It checks whether s+ch is in the dictionary. If yes, the index is taken, but if not, a new index is added to it in the dictionary as shown in Algorithm 3.1.

Algorithm 3.1 Data Value Minimizing & Original LZW Algorithm

Input : string st.

Output: encode st to output.

```
Begin
x ← length(st)
i ← 1
nval ←  st[i]
Buffer[i] ← nval
While (i < x)
    fval ← st[i]
    cval ← st[i+1]
    nval ← fval-cval
    Buffer[i+1] ← nval
    Fval ← cval
    i ← i+1
end
s ← empty string
while (Buffer is not empty)
    ch ← read from Buffer
    if (dictionary contains s+ch)
        s ← s+ch
        else
            encode s to output file
            add s+ch to dictionary
            s ← ch
        end
    end
end
return s
```

3.2.2 Variable Index LZW (VI-LZW)

In the original LZW algorithm, a fixed index size is used in the compression and decompression process, so the Variable Index LZW (VI-LZW) algorithm, which makes the index size automatically determined according to the size of the data.

At the beginning there are two entries which are (data and ch), the data is compressed by reading the first character(data) and the second one (ch), if data+ch is present in the dictionary, the index is taken as output, but if it does not exist, a string is added to the dictionary. In this new indexed location represented with a dynamic token size according to the number of bits of the index. The indexes are classified into four classes: A, B, C and D. The size of index of class A, B, C , D is one, two, three, and four bytes respectively as show in algorithm 3.2.

Algorithm 3.2 Variable Index LZW (VI-LZW)Algorithm

Input : string data , character ch.
Output: encode s to output file.

```
s ← empty string
i ← 0
while (there is still data to be read)
    ch ← read a character
    if (dictionary contains s+ch)
        s ← s+ch
        else
            index ← table(s)
            x ← class(index)
            index ← (x,index)
            Buffer[i] ← (index)
            i ← i+1
            add s+ch to dictionary
            s ←ch
        end
    end
end
return Buffer
```

3.2.3 Data Value Minimizing & Variable Index LZW Algorithm (DVM VI-LZW)

In the last section, work is done on both the algorithm (VI-LZW) and DVM in order to obtain the best result of the compression process.

This scheme is merging the two previous mentioned algorithms(VI) and (DVM). So that in the current scheme both VI and DVM algorithms are implemented. At the beginning the DVM algorithm would be executed to increase the frequent of data that is reducing data randomness. When data randomness is decreased, the compression ratio would be increased.

In the next stage, the VI algorithm is implemented inside the LZW improve its work specially with small size data.

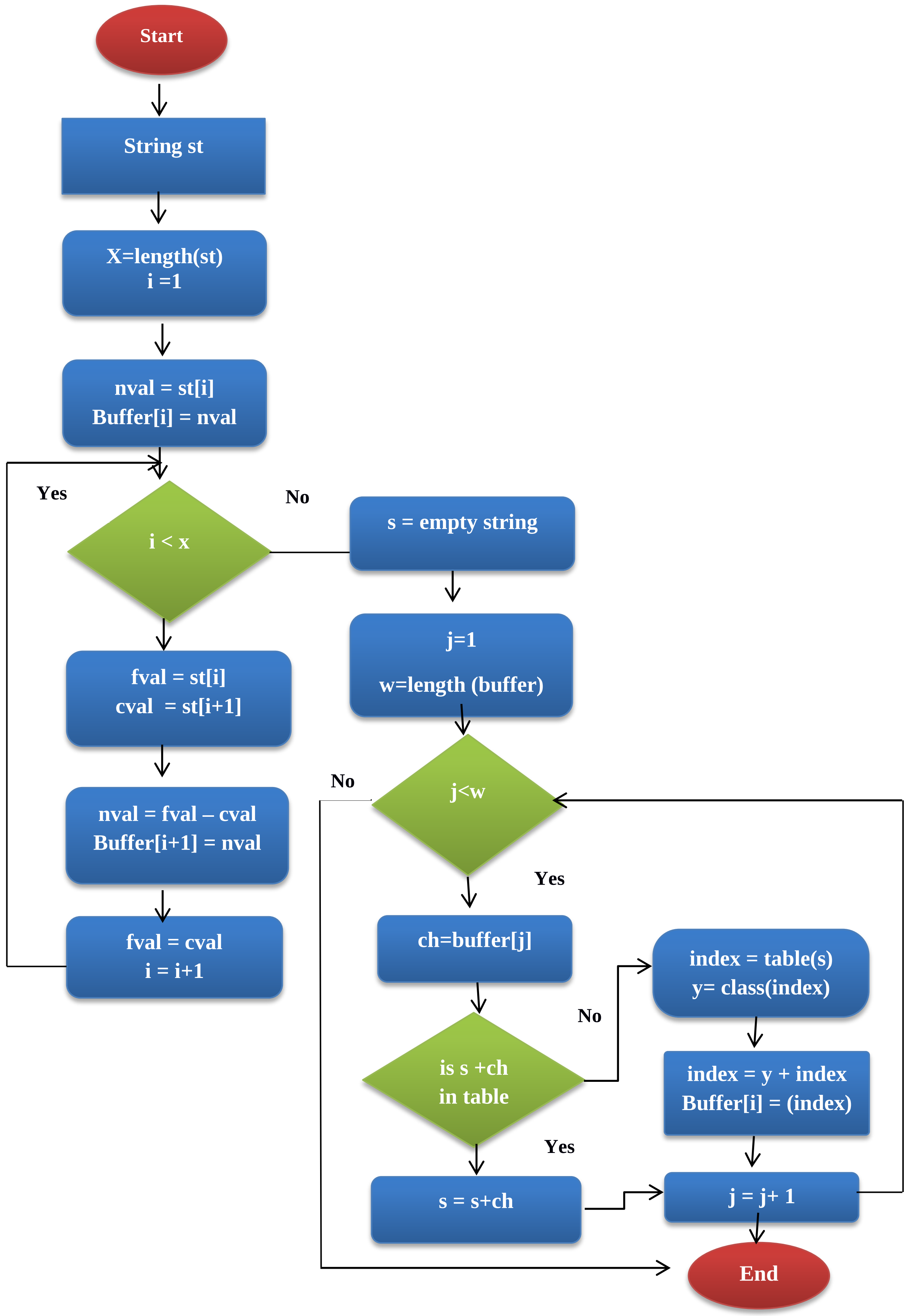

Figure 3.1 flowchart of DVM & VI-LZW

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

The proposed work is tested by collecting data from reality. Three experiments are carried out for an hour, two hours and three hours. Four tests are executed in each experiment. The data size and the time needed for transmission were compared, and through the time and size difference, the performance were calculated.

4. 2 Methodology of Research

This section includes four sub-sections. They are Material, Network installation, Experiments, and Metrics.

4.2.1 Material

The proposed work is being implemented by collecting real data from the sensor network in Al-Hilla, Al-Moallem District, at 10 am on 10/3/2022. Work is carried out with the following **materials:**

1- Two Raspberry pi 4 models B deployed with (4GB), two Xbee Shield, breadboard, temperature, humidity and rain sensors.

2- Ethernet cable, USB cable, mouse, keyboard, HDMI cable, mini HDMI cable.

4.2.2 Network Installation

- **Software**

1- Install and set up Raspberry Pi OS for two Raspberry pi 4 models B (4GB).

2- Install VNC viewer on the laptop for the purpose of accessing and controlling the raspberry

3- Install NetBeans program for the purpose of programming the algorithms.

❖ **Hardware**

- **Sender**

1- Install Xbee Shield on Raspberry.

2- Connect the sensors to the raspberry transmitter, where a wire was connected to pin 5V and another to GND pin, the humidity and temperature sensor was connected to pin P1, while the rain sensor was connected to pin P4.

3- Connect the Ethernet cable between the laptop and the Raspberry.

4- Connecting a USB cable for the purpose of supplying power to the Raspberry.

As show in figure 4.1 the data reading in sendr raspberry.

test 1 - Notepad

File Edit Format View Help


```
[[21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21
21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21,
43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,
43, 43, 43, 43, 43, 43, 43, 43, 43], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
[[21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21
21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21,
43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,
43, 43, 43, 43, 43, 43, 43, 43, 43], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
[[21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21
21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21,
43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43, 43,
43, 43, 43, 43, 43, 43, 43, 43, 43], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
```


Figure 4.1 Sender Raspberry File

- **Receiver (Sink)**

1- Install Xbee Shield on Raspberry.

2- Connecting a USB cable for the purpose of supplying power to the Raspberry.

3- Connect display, mouse and keyboard to Raspberry for the purpose of controlling it.

4- Connect mini HDMI from TV to raspberry.

As show in figure 4.2 the data reading in receiver raspberry.

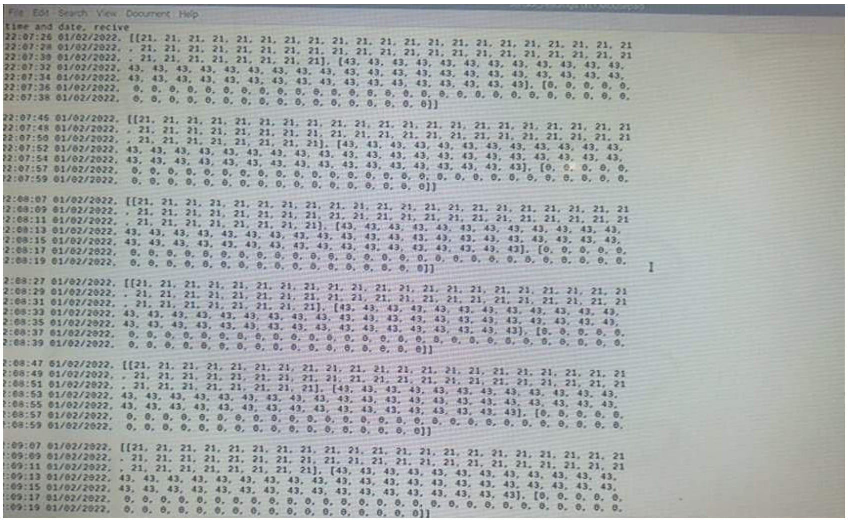

Figure 4.2 Receiver Raspberry File

4.2.3 Experiments

<u>Three Experiments is implemented:</u>

1- The data read every hour (3600 seconds). Each value is stored in one byte (128_-127) for each of the humidity, temperature and rain (3 bytes), it sent (3,600) bytes every hour. The first clock contains 24 readings (each reading is 150 values) so 24 * 150 = 3600, stored in text file and sends it to another raspberry.

2- The second clock contains 48 readings (each reading is 150 values) so 48 * 150 = 7,200, stored in text file and sends it to another raspberry.

3- And third one that contains 72 readings (each reading is 150 values) for this 72 * 150 = 10,800, stored in text file and sends it to another raspberry.

<u>Five tests are conducted over the three hours:</u>

1- First, the data is sent without any compression in order to compare the results after compression.

2- Compressed and send the data using the traditional LZW algorithm see the difference from the first sending.

3- The difference between the previous value and the new value was calculated using DVM code and stored in new text files, then compressed with the original LZW and sent to the sink.

4- In the fourth test, the original data is compressed and sends it using the proposed algorithm (VI-LZW).

5- The last test was performed on new files whose data had been modified using DVM code and compresses it with the proposed algorithm (VI-LZW).

4.3 Results

This section and its sub-section includes the computed result of the experiments. Each experiment has five tables. Tables from 1 to 5 have data compression size, compression ratio, time taken to transmit, throughput and saving percentage respectively.

<u>4.3.1 First Experiment</u>

The size of the data in the this experiment is (13,392) byte which contains 24 readings.

Table 4.1 Compressed Data Size [first experiment]

Test	**Size(Byte)**
Original Data	13,392
Original LZW	1,334
DVM and Original LZW	834
VI-LZW	1,301
DVM VI-LZW	814

Five tests are done on this experiment; first, the data is sent in its original size, and in the second test, the data is compressed with the original LZW algorithm to become its size (1,334) bytes, In the third test, the data is pre-processed, where the first value is subtracted from the second using a DVM algorithm and then compressed with the original LZW algorithm. The size of the data becomes (834) bytes. In the third, the data is compressed by an (VI-LZW) algorithm and the size of the data becomes (1,301) bytes. In the last test, the processed data is compressed with a DVM algorithm by (VI-LZW) algorithm and the size of the data (814) bytes. Through the above results, we notice the best results when using the DVM code because the climate readings change in them slowly, so there is a large repetition of the read data and the amount of pressure increases when using the DVM code with the VI-LZW algorithm due to the division of the index into classes.

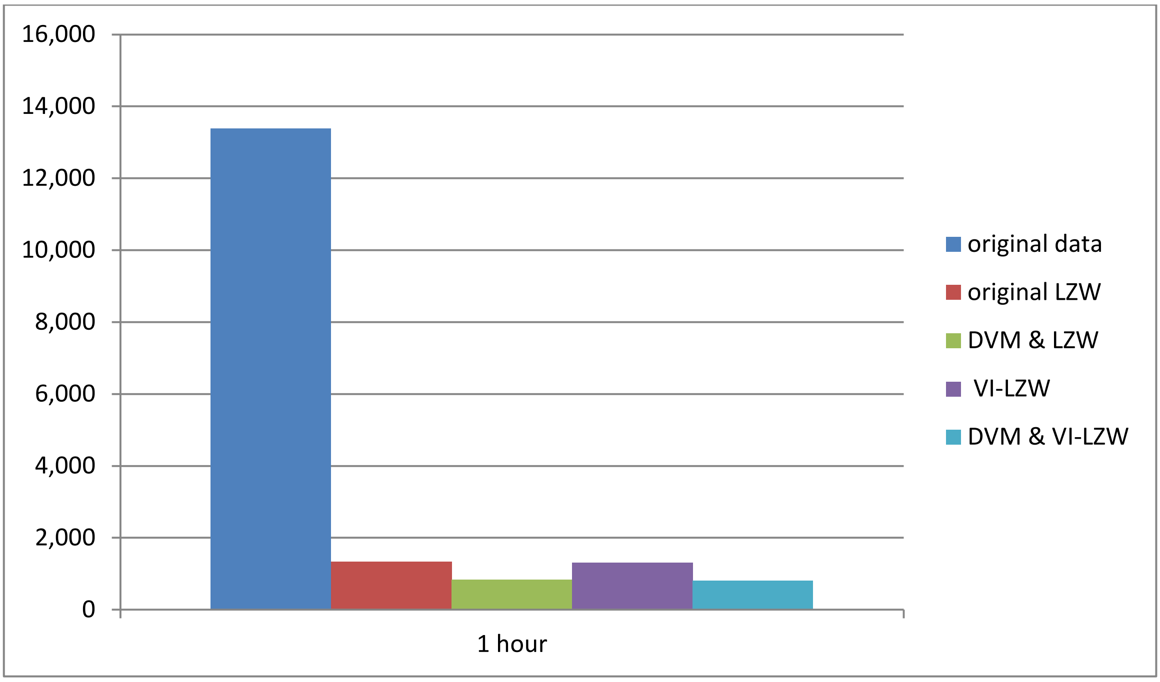

Figure 4.3 Compressed Data Size for First Experiment

Table 4.2 Compression Ratio[first experiment]

Test	Rate
Original LZW	9.9
DVM Original LZW	6.2
VI-LZW	9.7
DVM and VI-LZW	6.1

In the table 4.2, the comperession ratio is calculated for the first experiment, where it notes that the pressure ratio in the first test is (9.9), while in the second, the ratio is (6.2), and in the third the pressure ratio is (9.7), and in the last was highest ratio, which is (6.1). We note that the amount of uncompressed data is minimal when using the DFM code and VI-LZW .

Table 4.3 Transmission Time[first experiment]

Test	Time(ms)
Original Data	24,000
Original LZW	2,420
DVM and Original LZW	1,429
VI-LZW	2,400
DVM VI-LZW	1,409

It notices from the previous table 4.3 that the shortest time to send was in the last test, as it only took 1,429 ms. In the table 4.4, the result in the DVM and VI-LZW (0.591B/s). …. .

Table 4.4 System Throughput [first experiment]

Test	Throughput(B/s)
Original Data	0.558
Original LZW	0.551
DVM and Original LZW	0.569
VI-LZW	0.542
DVM VI-LZW	0.591

Through the clear results in the table 4.5, it notices that the last row, which is the DVM and VI-LZW, is considered the best results compared to the other tests.

Table 4.5 Saving Percentage [first experiment]

Test	Percentage
Original LZW	90.03%
DVM Original LZW	93.77%
VI-LZW	90.29%
DVM and VI-LZW	93.92%

4.3.2 Second Experiment

The size of the data in the this experiment is (26,784) byte which contains 48 readings.

Table 4.6 Compressed Data Size[second experiment]

Test	Size(Byte)
Original Data	26,784
Original LZW	2,566
DVM and Original LZW	1,749
VI-LZW	2,494
DVM VI-LZW	1,687

Five tests are done on this experiment; first, the data is sent in its original size, and in the second test, the data is compressed with the original LZW algorithm to become its size (2,566) bytes, In the third test, the data is pre-processed, where the first value is subtracted from the second using a DVM algorithm and then compressed with the original LZW algorithm. The size of the data becomes (1,749) bytes. In the third, the data is compressed by an (VI-LZW) algorithm and the size of the data becomes (2,494) bytes. In the last test, the processed data is compressed with a DVM algorithm by (VI-LZW) algorithm to be the size of the data (1,687) bytes. Through the above results, we notice the best results when using the DVM code because the climate readings change in them slowly, so there is a large repetition of the read data and the amount of pressure increases when using the DVM code with the VI-LZW algorithm due to the division of the index into classes.

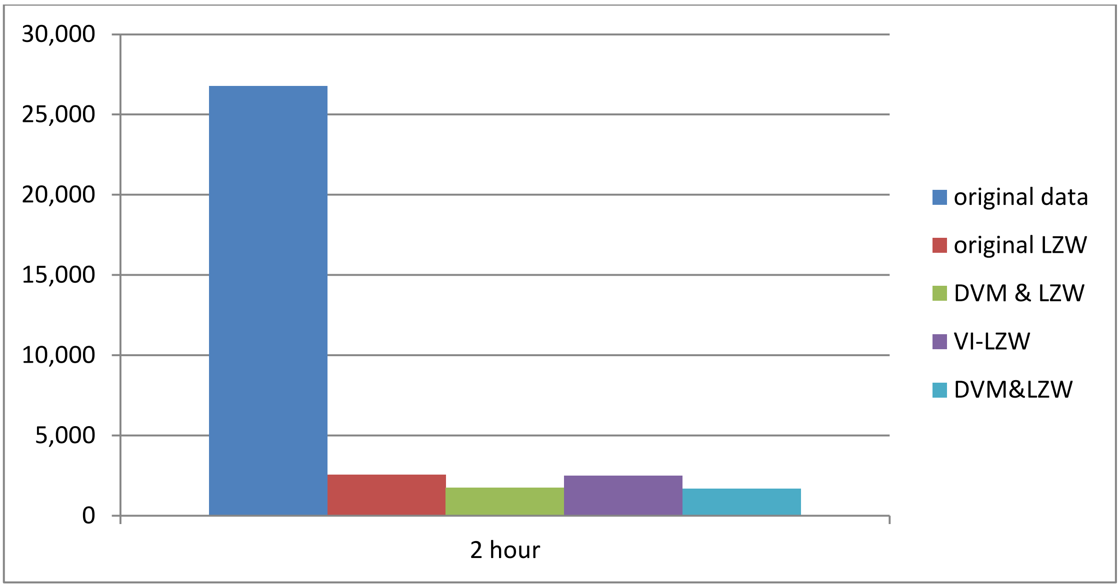

Figure 4.4 Compressed Data Size for Second Experiment

Table 4.7 Compression Ratio[second experiment]

Test	Rate
Original LZW	9.6
DVM and Original LZW	6.5
VI-LZW	9.3
DVM VI-LZW	6.3

In the table 4.7, the pressure ratio is calculated for the first experiment, where it notes that the pressure ratio in the first test is (9.6), while in the second, the ratio is (6.5), and in the third the pressure ratio is (9.3), and in the last was highest ratio, which is (6.3). We note that the amount of uncompressed data is minimal when using the DFM code and VI-LZW .

Table 4.8 Transmission Time [second experiment]

Test	Time(ms)
Original Data	48,000
Original LZW	5,429
DVM and Original LZW	3,100
VI-LZW	4,492
DVM VI-LZW	3,040

It notices from the table 4.8 that the shortest time to send was in the last test, as it only took 3,040 ms.

In the table 4.9, the highest result is in the DVM and VI-LZW test (0.554B/s).

Table 4.9 System Throughput[second experiment]

Test	Throughput(B/s)
Original Data	0.558
Original LZW	0.472
DVM and Original LZW	0.564
VI-LZW	0.555
DVM VI-LZW	0.554

Through the clear results in the table 4.10, it notices that the last row, which is the DVM and VI-LZW, is considered the best results compared to the other tests. .

Table 4.10 Saving Percentage[second experiment]

Test	Percentage
Original LZW	90.42%
DVM and Original LZW	93.47%
VI-LZW	90.69%
DVM VI-LZW	93.70%

4.3.3 Third Experiment

The size of the data in the this experiment is (40,176) byte which contains 72 readings.

Table 4.11 Compressed Data Size[third experiment]

Test	Size(Byte)
Original Data	40,176
Original LZW	2,520
DVM and Original LZW	1,722
VI-LZW	2,498
DVM VI-LZW	1,705

Five tests are done on this experiment; first, the data is sent in its original size, and in the second test, the data is compressed with the original LZW algorithm to become its size (2,520) bytes, In the third test, the data is pre-processed, where the first value is subtracted from the second using a DVM algorithm and then compressed with the original LZW algorithm. The size of the data becomes (1,722) bytes. In the third, the data is compressed by an (VI-LZW) algorithm and the size of the data becomes (2,498) bytes. In the last test, the processed data is compressed with a DVM algorithm by (VI-LZW) algorithm to be the size of the data (1,705) bytes. Through the above results, we notice the best results when using the DVM code because the climate readings change in them slowly, so there is a large repetition of the read data and the amount of pressure increases when using the DVM code with the VI-LZW algorithm due to the division of the index into classes.

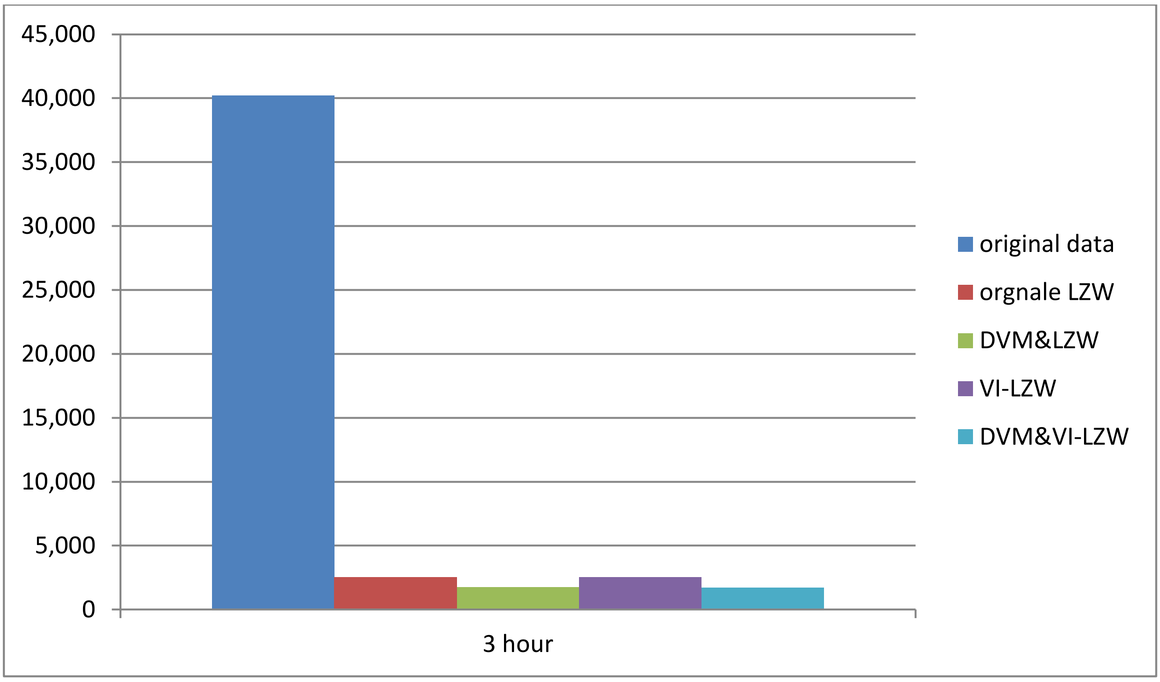

Figure 4.5 Compressed Data Size for Third Experiment

Table 4.12 Compression Ratio[third experiment]

Test	Rate
Original LZW	9.6
DVM and Original LZW	4.3
VI-LZW	6.5
DVM VI-LZW	6.3

In the table 4.12, the pressure ratio is calculated for the first experiment, where it notes that the pressure ratio in the first test is (9.6), while in the second, the ratio is (4.3), and in the third the pressure ratio is (6.5), and in the last was highest ratio, which is (6.3). We note that the amount of uncompressed data is minimal when using the DFM code and VI-LZW .

Table 4.13 Transmission Time[third experiment]

Test	Time(ms)
Original Data	72,000
Original LZW	4,570
DVM and Original LZW	3,120
VI-LZW	4,429
DVM VI-LZW	3,090

It notices from the table 4.13 that the shortest time to send was in the last test, as it only took 3,090 ms.

In the table 4.14, the highest result is in the DVM and VI-LZW test (0.551B/s). .

Table 4.14 System Throughput[third experiment]

Test	Throughput(B/s)
Original Data	0.558
Original LZW	0.551
DVM and Original LZW	0.551
VI-LZW	0.564
DVM VI-LZW	0.551

Through the clear results in the table 4.15, It notices that the last row, which is the DVM and VI-LZW, is considered the best results compared to the other test.

Table 4.15 Saving Percentage[third experiment]

Test	Percentage
Original LZW	93.72%
DVM and Original LZW	95.71%
VI-LZW	93.78%
DVM VI-LZW	95.75%

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

This study aims to develop the LZW algorithm to improve system performance.

1. The compression ratio depends on the frequent of the data.
2. Making the token size dynamically according to the data size increases the compression ratio.
3. When the number of frequent in the data is increased by subtracting the next value from the first for the data collected from reality by the sensors, the compression ratio in the proposed algorithm and also the original became very high.
4. When the size of data is big the problem of fixed token size doesn't appear.

5.2 Future Work

1. Use other schemes like Huffman for testing instead of LZW.
2. Use one value for representing data when the climate conditions is stable for some time in preprocessing operation.
3. Repeat the experiments with other conditions in terms of location and degree.

REFERENCES

[1] Guy, C. (2006, October). Wireless sensor networks. In Sixth international symposium on instrumentation and control technology: Signal analysis, measurement theory, photo-electronic technology, and artificial intelligence (Vol. 6357, p. 63571I).*International Society for Optics and Photonics.*

[2] S. A. Abdulzahra, A. K. M. Al-Qurabat, et al. (2020, March). Data reduction based on compression technique for big data in IoT. *In 2020 international conference on emerging smart computing and informatics (ESCI) (pp. 103-108).* IEEE.

[3] S. Pushpalatha, K.S. Shivaprakasha, (2020). Energy-efficient communication using data aggregation and data compression techniques in wireless sensor networks: A survey. *In Advances in communication, signal processing, VLSI, and embedded systems (pp. 161-179*). Springer, Singapore.

[4] Kodituwakku, S. R., & Amarasinghe, U. S. (2010). Comparison of lossless data compression algorithms for text data. *Indian journal of computer science and engineering, 1(4), 416-425.*

[5] Shanmugasundaram, S., & Lourdusamy, R. (2011). A comparative study of text compression algorithms. International Journal of Wisdom Based Computing, 1(3), 68-76.

[6] Mamta Rani and Vikram Singh (June 20 Neha Sharma 16), A SURVEY ON LOSSLESS TEXT DATA COMPRESSION TECHNIQUES. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 5, Issue 6.*

[7] Jancy, S., & Jayakumar, C. (2015). Packet level data compression techniques for wireless sensor networks. *Journal of Theoretical and Applied Information Technology* J. Theor. Appl. Inf. Technol, 75(1).

[8] Yuanbin, M., Yubing, Q., Jizhong, L., & Yanxia, L. (2011, March). A data compression algorithm based on adaptive Huffman code for wireless sensor networks. In 2011 Fourth International Conference on Intelligent Computation Technology and Automation (Vol. 1, pp. 3-6). IEEE.

[9] Cui, W. (2007, November). New LZW data compression algorithm and its FPGA implementation. In *Proc. 26th Picture Coding Symposium (PCS 2007).*

[10] Kaur, S. (2012). Design and Implementation of Lzw data compression algorithm. International Journal of Information Sciences and Techniques (IJIST) Vol, 2.

[11] Buratti, C., Conti, A., Dardari, D., & Verdone, R. (2009). An overview on wireless sensor networks technology and evolution. Sensors, 9(9), 6869-6896.

[12] Steenkiste, P., Sicker, D., Minden, G., & Raychaudhuri, D. (2009, March). Future directions in cognitive radio network research. In *NSF workshop report* (Vol. 4, No. 1, pp. 1-2).

[13] Akleek, F. A., Alquraan, R., & Shurman, M. (2020, December). Enhancement of WSN Network Lifetime. In *2020 32nd International Conference on Microelectronics (ICM)* (pp. 1-5). IEEE.

[14] Marciniak, T., Bujnowski, S., Marciniak, B., & Lutowski, Z. (2019). Method of limiting the emissivity of WSN networks. *Turkish Journal of Electrical Engineering & Computer Sciences*, *27*(5), 3470-3485.

[15] Gupta, S., Patel, H. D., Shukla, S. K., & Gupta, R. (2005). Design issues for networked embedded systems. In *Embedded Systems: Handbook* (pp. 29-1). CRC Press.

[16] Fischione, C. (2014). An introduction to wireless sensor networks. *Royal Institute of technology*. *Draft*, *version*, *1*.

[17] Pal, S. K., & Misra, S. (2016). *Soft computing applications in sensor*

networks. Chapman and Hall/CRC.

[18] "Wireless Sensor Networks Formation: Approaches", [Online]. Available: https://www.hindawi.com/journals/js/2016/2081902/

[19] "machine learning capabilities, limitations and implications " [Online]. Available: https://apo.org.au/node/208196 .

[20] Olakanmi, O. O., & Dada, A. (2020). Wireless sensor networks (WSNs): Security and privacy issues and solutions. In *Wireless mesh networks-security, architectures and protocols* (p. 13). IntechOpen.

[21] Sohraby, K., Minoli, D., & Znati, T. (2007). Basic wireless sensor technology, Book “Wireless Sensor Networks: Technology, Protocols, and Applications”.

[22] Benhaddou, D., & Al-Fuqaha, A. (2015). *Wireless Sensor and Mobile Ad-Hoc Networks* (p. 253). by Driss Benhaddou and Ala Al-Fuqaha, Book, Wireless Sensor and Mobile Ad-Hoc Networks.

[23] Matin, M. A., & Islam, M. M. (2012). Overview of wireless sensor network. *Wireless sensor networks-technology and protocols*, *1*(3).

[24] Kandris, D., Nakas, C., Vomvas, D., & Koulouras, G. (2020). *Applications of wireless sensor networks: an up-to-date survey. Applied System Innovation, 3(1), 14.*

[25] Warneke, B., Last, M., Liebowitz, B., elt. (2001). *Smart dust: Communicating with a cubic-millimeter computer. Computer, 34(1), 44-51.*

[26] Debessonet, “Artificial Intelligence Application in the Law,” High Tech. L.J. 329 (1986), vol. 1, no. 2, 2014.

[27] World Health Organization. (2012). Environmental monitoring of clean rooms in vaccine manufacturing facilities. *Switzerland*, *201*(1), 27.

[28] Bal, M. (2014, June). Industrial applications of collaborative wireless sensor networks: A survey. *In 2014 IEEE 23rd international symposium on industrial electronics (ISIE) (pp. 1463-1468). IEEE.*

[29] Wu, Z., Chu, H., Pan, Y., & Yang, X. (2006, December). Bus priority control system based on wireless sensor network (WSN) and zigbee. In *2006 IEEE International Conference on Vehicular Electronics and Safety* (pp. 148-151). IEEE.

[30] "Network Topologies", [Online]. Available: http://kingofnetworking.weebly.com/network-topologies.html

[31] Kaur, K., Kaur, P., & Singh, E. S. (2014). Wireless Sensor Network: Architecture, Design Issues and Applications. International Journal of Scientific Engineering and Research (IJSER), 2(11).

[32] Alkhatib, A. A. A., & Baicher, G. S. (2012, April). Wireless sensor network architecture. In 2012 International conference on computer networks and communication systems (CNCS 2012).

[33] Khanafer, M., Guennoun, M., & Mouftah, H. T. (2009, December). WSN architectures for intelligent transportation systems. In *2009 3rd International Conference on New Technologies, Mobility and Security* (pp. 1-8). IEEE.

[34] He, T., Hai, Z., & Shi-liang, S. (2017). WSN Architecture Design Based on Software Defined Networks, Advances in Engineering Research, volume 119, 2nd International Conference on Automatic Control and Information Engineering (ICACIE).

[35] Mohammad, N., Muhammad, S., Bashar, A., & Khan, M. A. (2017, November). Design and modeling of energy efficient wsn architecture for tactical applications. In *2017 Military Communications and Information Systems Conference (MilCIS)* (pp. 1-6). IEEE.

[36] Moussa, N., & El Alaoui, A. E. B. (2022). DACOR: A distributed ACO-based routing protocol for mitigating the hot spot problem in fog-enabled WSN architecture. *International Journal of Communication Systems*, e5008.

[37] Alkhatib, A. A. A., & Baicher, G. S. (2012, April). Wireless sensor

network architecture. In 2012 International conference on computer networks and communication systems (CNCS 2012). Singapore: IACSIT Press.

[38] Liu, Y., Wu, Q., Zhao, T., Tie, Y., Bai, F., & Jin, M. (2019). An improved energy-efficient routing protocol for wireless sensor networks. Sensors, 19(20), 4579.

[39] You-Chiun Wang, Data Compression Techniques in Wireless Sensor Networks ,*University, Hsin-Chu, Taiwan.*

[40] Shrestha, A., & Xing, L. (2007, May). A performance comparison of different topologies for wireless sensor networks. In 2007 IEEE Conference on Technologies for Homeland Security (pp. 280-285). IEEE.

[41] Yan-li, Z., Xiao-ping, F., Shao-qiang, L., & Zhe-yuan, X. (2010, July). Improved LZW algorithm of lossless data compression for WSN. In *2010 3rd International Conference on Computer Science and Information Technology* (Vol. 4, pp. 523-527). IEEE.

[42] Jayasankar, U., Thirumal, V., & Ponnurangam, D. (2021). A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications. Journal of King Saud University-Computer and Information Sciences, 33(2), 119-140.

[43] Jiang, P., & Li, S. Q. (2010). A data compression algorithm for wireless sensor networks based on an optimal order estimation model and distributed coding. *Sensors*, *10*(10), 9065-9083.

[44] Al-Laham, M., & El Emary, I. M. (2007). Comparative study between various algorithms of data compression techniques. IJCSNS, 7(4), 281.

[45] Wang, C. M., Yen, C. C., Yang, W. Y., & Wang, J. S. (2016, May). Tree-structured linear approximation for data compression over wsns. In *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)* (pp. 43-51). IEEE.

[46] Papaioannou, T. G., Riahi, M., & Aberer, K. (2011, June). Towards online multi-model approximation of time series. In *2011 IEEE 12th International Conference on Mobile Data Management* (Vol. 1, pp. 33-38). IEEE.

[47] Sathe, S., Papaioannou, T. G., Jeung, H., & Aberer, K. (2013). A survey of model-based sensor data acquisition and management. In *Managing and mining sensor data* (pp. 9-50). Springer, Boston, MA.

[48] Upton, Eben (24 June 2019). "Raspberry Pi 4 on sale now from $35". Raspberry Pi Foundation.

[49] U. M. Netravati, “Raspberry Pi 4,” Raspberry Pi Found., no. May, p. 1129409, 2020.

[50] Heile, B., & Chairman, Z. (2006). Alliance. Wireless Control That Simply Works, Wireless Sensors and Control Networks: Enabling New Opportunities with ZigBee.

[51] Z. Alliance, “ZigBee Specificarion,” *Stand. Oct*, 2015.

[52] "Raspberry Pi Arduino Shield Xbee Expansion Hat ", [Online]. Available: https://www.ardunic.com/product/Raspberry-Pi-Arduino-Shield-Xbee-Expansion-Hat . Access in 1/6/2020 at 3:15 pm

[53] Gay, W. (2018). DHT11 sensor. In *Advanced Raspberry Pi* (pp. 399-418). Apress, Berkeley, CA.

[54] "In-Depth: Interface DHT11 Module With Arduino", [Online]. Available: https://lastminuteengineers.com/dht11-module-arduino-tutorial/

[55] "Rain Sensor : Working, Pin Configuration and Applications", [Online]. Available : https://www.elprocus.com/rain-sensor-working-and-its-applications/

[56] "TOLAKO Rain Sensor Module Humidity Raindrop Weather Detection Module for Arduino", [Online]. Available:
https://www.makerlab-electronics.com/product/product-details-of-raindrops

detection-sensor-rain-water-weather-module-humidity-cable-for-arduino/

الخلاصة

شبكة المستشعرات اللاسلكية لها العديد من القيود بما في ذلك أداء الشبكة. تقلل هذه القيود أيضًا إنتاجية الشبكة الإجمالية. تدفع صعوبة استبدال مصدر الطاقة عند نفادها الباحثين إلى تطوير مخططات لتقليل وقت تشغيل WSN ,لأن نقل البيانات غير المضغوطة يستغرق الكثير من الوقت ، لذلك ينفد مصدر الطاقة بسرعة. سؤال البحث هو كيف يتم تقليل كمية البيانات المنقولة لتحسين أداء الشبكة؟

سيتضمن العمل طريقة ضغط القاموس Lempel Ziv Welch (LZW) ، تتمثل إحدى مشكلات طريقة القاموس في أن حجم الرمز المميز ثابت. علاوة على ذلك ، على سبيل المثال ، يتم إهدار 2 بايت عندما يحتاج بايت واحد إلى تمثيل رمز مميز (حيث يجب تمثيل حجم الرمز المميز بثلاثة بايت). طريقة قاموس LZW ليست مفيدة جدًا مع القليل من البيانات ، لأنها تفقد العديد من وحدات البايت عند تخزين الرموز صغيرة الحجم.

يقترح العمل المقترح استخدام رمز حجم ديناميكي حيث يتم تصنيف الرموز وفقًا لحجمها إلى (بايت واحد ، اثنان بايت ، ثلاثة بايت).

الفكرة الرئيسية للعمل المقترح هي زيادة البيانات المتكررة لزيادة نسبة الضغط.

لزيادة البيانات الزائدة عن الحاجة ، يقترح العمل الاحتفاظ بالكمية المتزايدة من البيانات المقروءة بدلاً من الاحتفاظ بالبيانات الحقيقية بأكملها. لأن بيانات قراءة المناخ تتغير ببطء شديد ، بحيث يكون مقدار التغيير متكررًا.

تم تنفيذ العمل تجريبيا. تم إجراء ثلاث تجارب. تتضمن كل تجربة أربعة اختبارات. يتم إجراء التجربة الأولى على البيانات التي تم جمعها في ساعة واحدة. تعمل التجربة الثانية على البيانات التي تم جمعها في ساعتين. يتم إجراء التجربة الثالثة على البيانات التي تم جمعها في ثلاث ساعات.

الاختبارات الخاصة بكل تجربة هي اولا ضغط البيانات باختبار LZW ، ضغط البيانات بتقليل قيمة البيانات (DVM) بأختبار LZW ـ البيانات باستخدام اختبار مؤشر المتغير (VI-LZW) البيانات باستخدام اختبار (DVM) و (VI-LZW).

اظهرت النتيجة التجريبية كيف أن (DVM) و (VI-LZW) يوفران ضغط بيانات أفضل للبيانات حيث تم ضغط 13,392 إلى 814 بايت ، ونسبة ضغط أفضل تبلغ 6.1 ، ووقت نقل البيانات 1,409 ملي ثانية وإنتاجية النظام 0.591 و 93.92٪ نسبة توفير.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة بابل كلية تكنولوجيا المعلومات

قسم شبكات المعلومات

خوارزمية الضغط LZW المعدلة لتحسين إداء شبكة المستشعرات اللاسلكية

رسالة

مقدمة إلى مجلس كلية تكنولوجيا المعلومات في جامعة بابل والتي هي جزء من
متطلبات الحصول على درجة الماجستير في تكنولوجيا المعلومات - شبكات المعلومات

زهراء مازن بهلول صالح

بأشراف

أ.م.د سيف محمود خلف حسين العلاق

2022م 1444هـ