MACHINE LEARNING-BASED DATA COMPRESSION FOR ENERGY-EFFICIENT TRANSMISSION IN WIRELESS SENSOR NETWORK

BY

OLAITAN OLUWASEMILORE EMMANUEL

ENG1905119

DEPARTMENT OF COMPUTER ENGINEERING,

FACULTY OF ENGINEERING

UNIVERSITY OF BENIN.

BENIN CITY

IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF BACHELOR OF ENGINEERING DEGREE

FEBUARY, 2025

CERTIFICATION

This project was carried out by Olaitan Oluwasemilore Emmanuel, from the department of Computer Engineering, Faculty of Engineering, University of Benin, Benin City, and is hereby certified.

Engr. Dr. I.O Omoifo — Date

(Project Supervisor)

Engr. Dr. (Mrs) O. Okosun — Date

(Head of Department)

DEDICATION

I dedicate this project to the Almighty God for His guidance, wisdom, and strength throughout this journey. To my beloved family, whose unwavering support, encouragement, and sacrifices have been a pillar of strength in my academic pursuit. To my mentors, supervisors, and friends who have inspired and motivated me to push beyond limits, this work is a testament to your belief in me.

ACKNOWLEDGEMENT

I OLAITAN OLUWASEMILORE EMMANUEL wish to attribute this work to GOD ALMIGHTY, who has given me strength and knowledge to complete this project work.

I sincerely appreciate all efforts and inputs rendered unto me by my project supervisor Engr. Dr. I.O Omoifo, for his valuable support, cooperation, guidelines and suggestions from time to time ensuring the successful completion of this project work.

I sincerely appreciate my dear parents Mr. and Mrs. Mark-Olaitan, who have shown me love and made a lot of sacrifices that ensured my comfort and success throughout my academic journey.

This acknowledgement will not be complete without expressing my appreciation to my beloved sibling Bunmi, my uncles and aunties, and my cousins aunty Magbe, bros Ayes, bros Ugue, bros Ese, and Zogie, for their unwavering supports which have been a pillar of strength and encouragement throughout my journey in this school.

I want to thank my brother and my friend Edenojie Gift Prince for his support from 100 level till date and for his immense contribution, and also all my friends, roommates, and people I came in contact with during my stay in this school.

Lastly, I want to thank Fc Barcelona for making my mental health straight from the beginning of this season, especially Lamine Yamal and Raphinha who have given me joy throughout this season. I say I am very grateful.

ABSTRACT

Wireless Sensor Networks (WSNs) play a crucial role in modern communication systems, particularly in environmental monitoring, industrial automation, and smart cities. However, a major challenge in WSNs is optimizing energy consumption due to the limited power resources of sensor nodes. One of the most effective ways to enhance energy efficiency is through data compression, which reduces the amount of transmitted data while preserving essential information.

This project explores the integration of machine learning-based data compression techniques to improve energy-efficient transmission in WSNs. A hybrid approach is proposed, combining Run-Length Encoding (RLE) as a traditional lossless compression method with Principal Component Analysis (PCA) as a machine learning algorithm to reduce data redundancy while maintaining accuracy. The study focuses on temperature sensor datasets collected over a specified period, ensuring real-world applicability.

The methodology involves preprocessing raw temperature data, applying Run-Length Encoding (RLE) for initial redundancy reduction, and then leveraging PCA to extract principal components, further reducing data dimensions before transmission. The efficiency of the proposed model is evaluated based on key metrics such as compression ratio, reconstruction accuracy, and energy savings. Performance comparisons are made with conventional lossless compression algorithms like Huffman Coding and Arithmetic Coding to assess improvements.

Preliminary results indicate that the combined approach achieves a higher compression ratio while preserving critical temperature variations, leading to significant energy savings in wireless transmissions. This work contributes to advancing energy-efficient data handling in WSNs, making it highly relevant for resource-constrained environments. Future research directions include expanding the model to handle multi-sensor data streams and implementing real-time adaptive compression strategies.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND OF STUDY

Wireless sensor networks (WSNs) play a vital role in many modern applications, from monitoring environmental changes to managing smart cities. However, one of their biggest challenges is the limited energy available to the tiny sensor nodes that collect and transmit data. As these nodes often operate in remote areas, where replacing batteries is difficult or even impossible, conserving energy becomes a top priority. Transmitting large amounts of data drains energy quickly, which is why finding better ways to compress data is so important. Traditional compression techniques help, but they often don't do enough to strike the right balance between reducing the amount of data and saving energy. This is where machine learning offers exciting new possibilities. By applying intelligent data compression methods, we can make WSNs much more energy-efficient, helping to extend their operational lifetimes and make them more reliable. This project aims to explore and implement machine learning-based data compression techniques tailored for WSNs, with the goal of achieving a more energy-efficient network and extending sensor node lifetime.

1.1.1 WIRELESS SENSOR NETWORKS (WSNS)

Wireless Sensor Networks (WSNs) are systems made up of multiple small, independent sensors that are scattered across a specific area to gather environmental data, like temperature, humidity, or light intensity. These sensors work autonomously, meaning they don't need direct human control to perform their tasks. Once they collect data, they send it to a central hub, where it is analyzed and used for decision-making. WSNs are fundamental in various industries, including environmental monitoring, where they track climate conditions; healthcare, where they monitor patient vital signs in real-time; smart cities, which use them for things like traffic and pollution control; and industrial automation, where they help optimize manufacturing processes.

However, WSNs face significant hurdles, the biggest one being energy limitations. These sensor nodes typically run-on batteries, and because they are often placed in remote or hard-to-reach areas, recharging or replacing batteries can be difficult, if not impossible. This constraint puts a cap on how long a WSN can operate without intervention, limiting its lifespan. The transmission of data over long distances is particularly energy-intensive, which is why much of the research around WSNs focuses on improving energy efficiency—finding ways to reduce the amount of power needed to send data, without sacrificing the accuracy or quality of the information gathered.

Because of this, many cutting-edge solutions are being developed to extend the life of WSNs, including data compression techniques and machine learning algorithms that help sensors use energy more efficiently, especially in fields where long-term deployment and minimal maintenance are crucial .

1.1.2 ENERGY EFFICIENCY IN WIRELESS SENSOR NETWORKS (WSNS)

One of the biggest challenges for Wireless Sensor Networks (WSNs) is managing energy consumption, particularly because sending data wirelessly drains a lot of battery power. This issue is especially concerning because the sensor nodes, which are often located in hard-to-reach or remote places, usually run-on batteries. When these batteries die, it's not easy to replace or recharge them, meaning the entire network's functionality can be compromised.

What researchers have found is that data transmission uses up far more energy than just processing or storing the data locally on the sensor node. In other words, the act of sending data wirelessly is what really eats up battery life. For this reason, a major focus has been on minimizing the amount of data that needs to be sent in the first place, as doing so can dramatically extend the lifespan of the sensor nodes.

Data compression is a powerful method that helps achieve this goal. By compressing the data, you can shrink it down, which means less information needs to be transmitted. This helps save energy while still preserving the essential information that the sensor nodes collect. Compression allows for more efficient communication between nodes and the central system, making WSNs more sustainable over longer periods of time.

1.1.3 DATA COMPRESSION IN WIRELESS SENSOR NETWORKS (WSNS)

Traditional data compression techniques play a crucial role in Wireless Sensor Networks (WSNs) by helping to reduce the volume of data transmitted, which is vital for efficient communication and energy conservation. There are two main types of compression methods: lossless and lossy.

Lossless compression techniques, such as Huffman Coding and Run-Length Encoding, are designed to ensure that no information is lost during the compression process. This characteristic is essential for critical applications where precision is paramount, like medical monitoring or environmental sensing. In these scenarios, every piece of data matter, and losing even a small amount of information could lead to incorrect conclusions or dangerous situations. With lossless methods, the original data can be perfectly reconstructed from the compressed version, which maintains the integrity of the information.

In contrast, lossy compression sacrifices some degree of accuracy to achieve a higher compression ratio. This means that while some data may be lost, the overall size of the data transmitted is significantly reduced. Lossy compression is particularly useful in applications where minor data loss is acceptable, such as with image and audio files. For example, in scenarios like streaming video or transmitting audio recordings, a slight decrease in quality can be a fair trade-off for the benefit of reduced file size and quicker transmission times .

Ultimately, the choice between lossless and lossy compression methods depends on the specific requirements of the application at hand, balancing the need for data integrity with the constraints of energy consumption and transmission efficiency. For those looking for a more in-depth discussion on the topic, additional resources are available in academic journals and articles focused on data compression techniques in WSNs.

1.1.4 MACHINE LEARNING-BASED DATA COMPRESSION

Machine learning (ML) provides a dynamic and efficient approach to data compression within Wireless Sensor Networks (WSNs). By utilizing sensor data to train models, machine learning algorithms can identify patterns and relationships that allow them to compress data more effectively than traditional methods. This capability is particularly advantageous because it means that the models can adapt to the specific characteristics of the data being collected.

One of the most effective tools in this area is the autoencoder, a specialized type of neural network commonly used for unsupervised compression tasks. Autoencoders work by taking input data and reducing its dimensionality, essentially learning a more compact representation of the original data. This compressed version can then be reconstructed with minimal information loss, ensuring that the integrity of the data remains intact even after compression.

A standout example of this technology is the Stacked Restricted Boltzmann Machine-Autoencoder (RBM-AE) model. This model has shown remarkable success in reducing the amount of data that needs to be transmitted in WSNs. It operates in two main stages: first, it compresses the sensor data in the encoding layer, and then it reconstructs this data in the decoding layer. The RBM-AE model is notable not only for its ability to compress data efficiently but also for its significant reduction in energy consumption while maintaining a high level of accuracy in data reconstruction.

This combination of adaptability, efficiency, and accuracy makes machine learning a game-changer for WSNs, allowing these networks to operate longer and more effectively, even in energy-constrained environments. As researchers continue to explore and refine these techniques, we can expect further advancements in how data is managed and transmitted across these critical networks, ultimately enhancing their performance in various applications, from environmental monitoring to smart city initiatives.

1.2 PROBLEM STATEMENT

Wireless Sensor Networks (WSNs) are at the forefront of technological advancements, playing an essential role in various applications such as environmental monitoring, healthcare, industrial automation, and smart city initiatives. These networks consist of spatially distributed sensor nodes that autonomously collect data from their surroundings, be it temperature, humidity, light levels, or other environmental metrics and transmit this data to centralized systems for further analysis. While WSNs present immense potential for enhancing data-driven decision-making, they face a significant challenge: the energy efficiency of data transmission.

The primary issue stems from the inherent limitations of sensor nodes, which are typically battery-operated and deployed in remote or hard-to-reach locations. The energy consumption associated with data transmission constitutes a substantial portion of a sensor node's overall power usage. This challenge is worsen when large volumes of data need to be transmitted, resulting in accelerated battery depletion and ultimately, the failure of the entire network. As noted by Akyildiz et al. (2002), the energy constraints of these nodes can lead to severe limitations in the lifespan and effectiveness of WSNs, thereby affecting their reliability and operational efficiency.

Data compression has emerged as a crucial technique to mitigate this problem, as it reduces the amount of data that needs to be transmitted, conserving energy and prolonging the operational lifespan of the sensor nodes. Traditional data compression methods, such as Huffman coding and Run-Length Encoding (RLE), have been adapted for use in WSNs. However, these approaches often lack the necessary flexibility and adaptability required for the dynamic and heterogeneous nature of the data generated by sensor networks. Many of these conventional algorithms were designed for general-purpose data and do not take into account the unique characteristics of sensor data, such as spatial and temporal correlations.

In recent years, the advent of machine learning (ML) has presented an innovative solution to the challenges associated with data compression in WSNs. ML algorithms have demonstrated the capability to learn patterns and correlations within data, enabling them to perform adaptive compression that maintains data fidelity while minimizing transmission size. Research has shown that machine learning-based approaches can significantly enhance the energy efficiency of data transmission in WSNs. However, despite these advancements, several critical issues remain unaddressed, including the development of lightweight ML models that can operate effectively within the constraints of battery-powered sensor nodes, the design of adaptive compression algorithms that can dynamically respond to changing data conditions, and the integration of robust security measures to protect sensitive data from potential vulnerabilities.

This project aims to explore the integration of machine learning-based data compression techniques in WSNs, focusing on creating a framework that enhances energy efficiency without compromising data quality. By leveraging advanced ML algorithms, this research will seek to fill the existing gaps in the current literature and offer practical solutions that can be deployed in real-world WSN applications.

1.3 AIM AND OBJECTIVES

AIM

The primary aim of this project is to develop a web-based data compression tool that compresses large data by utilizing machine learning algorithms to enhance the energy efficiency of data transmission in Wireless Sensor Networks (WSNs). This tool will provide a user-friendly, and efficient platform that enables the user to compress large volumes of sensor data from WSNs, thereby optimizing data transmission and storage while preserving data integrity accuracy.

OBJECTIVES

- To explore and implement traditional data compression techniques for data compression.
- To design user-friendly web-based application for data compression.
- To simulate the model in a simulation environment.
- To conduct performance benchmarking and evaluation.
- To enable seamless integration with existing WSN systems.

1.4 SCOPE OF STUDY

The goal of this project is to develop a web-based data compression tool specifically designed for Wireless Sensor Networks (WSNs)**.** The tool will implement machine learning techniques such as linear regression, random forests, decision trees**,** and k-nearest neighbors (KNN) to compress sensor data efficiently, optimizing energy usage during data transmission. This tool aims to reduce the amount of data transmitted by sensor nodes, thereby conserving energy, extending the life span of each node, and the network itself, and enhancing overall system performance.

1. **<u>Selection and comparison of machine learning algorithms:</u>** The research and evaluation of various machine learning algorithms, such as:

- **<u>Linear Regression:</u>** These algorithms will be used for predictive modeling and trend analysis in continuous data.
- **<u>Random Forests and Decision Trees</u>:** These algorithms will be used for classification and regression tasks, that will help identify patterns within the dataset for efficient compression.
- **<u>K-Nearest Neighbors (KNN):</u>** These algorithms will be used for clustering similar data points, aiding in data reduction by grouping similar sensor readings.

Comparison of these models based on performance metrics like compression ratio, speed, and computational efficiency on typical WSN datasets.

2. **<u>Selection and comparison of compression algorithms</u>:** Exploration, comparison and selection of the best compression techniques, looking at traditional methods like Huffman coding and Run- Length encoding (RLE), as well as more modern, machine learning-based approaches. We will optimize balance between compression speed and efficiency.

3. **<u>Design and Architecture of Web-based application:</u>** Design of the app's architecture for a user-friendly, web-based interface that will allow users to:
   - Upload sensor datasets.
   - Select desired machine learning compression algorithms.
   - Download compressed data files for efficient transmission and storage.

   Utilization of front-end technologies like HTML, CSS, JavaScript, and back-end framework such as python, to create the web application.

4. **<u>Development of compression features:</u>** This is where the main functionality comes in, we'll be writing codes that compresses the dataset users will upload. Users will get feedback on how much data was compressed and whether any data was lost (if they choose a lossy method).

5. **<u>Security and Authentication:</u>** We'll ensure only authorized users can access the tool and that all user data is protected. Here, we'll add features like sign-up, login and user role

management will be implemented, along with encryption and secure transmission protocols (SSL/TLS) to keep every information safe.

6. **Multiple File Formats:** Functionalities will be built to handle different file formats, such as CSV, JSON and other text- based files, allowing flexibility in data types that users can upload and compress.

7. **Integration of machine Learning Algorithms:** The machine learning models for compression will be implemented, ensuring that the models will effectively reduce data without significant information loss, optimize specific WSN data like temperature and also provides feedback to users on compression ratio and effectiveness. We'll also develop a system for training and fine-turning models as new data is processed, ensuring adaptive and continuous performance improvement.

8. **Testing and Evaluation:** The performance of the tool will be tested extensively with datasets of various sizes, optimizing its speed, compression ratio and overall performance to ensure it is both reliable and responsive.

9. **Deployment and maintenance:** The tool will be deployed on a cloud-based platform and regular maintenance procedures will be established.

10. **Performance Evaluation:** The tool's performance is measured based on key metrics like compression ratio, speed, and user satisfaction and feedback.

1.5 JUSTIFICATION OF STUDY

In today's technology-driven world, Wireless Sensor Networks (WSNs) have emerged as crucial components, particularly within the realms of the Internet of Things (IoT), environmental monitoring, smart cities, and healthcare systems. These networks comprise of various sensor nodes that diligently gather and transmit data from their environments. However, a significant challenge they face is energy consumption. Many of these sensor nodes are deployed in remote locations, often relying on batteries for power, which need to support both the sensing of data and its subsequent transmission.

1. **Enhancing understanding of data management:** By integrating machine learning techniques into data compression, the project provides an opportunity to enhance learners' understanding of complex concepts in data management. Students and practitioners will engage with real-world datasets, such as those pertaining to temperature and humidity, which will facilitate experiential learning. The hands-on experience gained through this project can improve their skills in applying machine learning algorithms, thus fostering deeper learning outcomes. Research indicates that active learning approaches lead to better retention of knowledge and a deeper understanding of material (Freeman et al., 2014).
2. **Development of critical thinking:** Implementing machine learning algorithms for data compression requires critical thinking and problem-solving skills. As students analyze the performance of different algorithms, assess their trade-offs, and troubleshoot issues, they cultivate a mindset geared toward analytical reasoning and innovation. This project encourages learners to evaluate the efficiency and applicability of various compression techniques in real-world scenarios, which can lead to improved learning outcomes. Studies have shown that engaging students in problem-based learning tasks promotes critical thinking skills (Hmelo-Silver, 2004).
3. **Bridging Knowledge Between Different Disciplines:** This project sits at the intersection of computer science, data analysis, and energy efficiency—fields that are increasingly interconnected in the digital age. By working on a project that involves machine learning and WSNs, learners are exposed to a range of disciplines, fostering a broader understanding of how these areas converge. This interdisciplinary approach can lead to enhanced learning outcomes, as students gain insights into how various fields influence and complement one another. Research emphasizes that interdisciplinary education prepares students for complex real-world problems (Beers, 2006).
4. **Preparation for Future Careers:** As industries increasingly adopt machine learning and data compression technologies, students equipped with these skills will have a competitive edge in the job market. This project not only enhances their technical expertise but also prepares them for future career opportunities in data science, IoT, and related fields. The

ability to understand and apply advanced algorithms is highly sought after, and projects like this one can significantly improve employability and career readiness (Cappelli et al., 2012).

5. **Contributions to Ongoing Research:** Finally, the findings from this study will contribute to ongoing research in the field, enhancing the collective knowledge base and promoting further exploration of machine learning applications in WSNs. By sharing insights gained through the project, the academic community can benefit, potentially leading to improved methodologies and practices in data management and energy efficiency. Research indicates that collaboration and knowledge sharing are crucial for advancing scientific inquiry (Cohen & Levinthal, 1990).

CHAPTER TWO

LITERATURE REVIEW

2.1 HISTORY AND EVOLUTION OF WIRELESS SENSOR NETWORKS AND DATA COMPRESSION.

2.1.1 EARLY DEVELOPMENT OF WIRELESS SENSOR NETWORKS (WSNs)

The journey of Wireless Sensor Networks (WSNs) began in the 1970s, a time when the U.S. military was exploring ways to enhance surveillance and monitoring capabilities. This led to the inception of the Distributed Sensor Networks (DSN) project, which primarily utilized wired systems. However, rapid advancements in wireless communication, microelectronics, and low-power processors laid the groundwork for what we now recognize as modern WSNs.

By the 1990s, significant breakthroughs in technologies like microelectromechanical systems (MEMS) and low-power electronics emerged. These innovations allowed for the creation of small, cost-effective sensor nodes that could operate independently. Each node is equipped with various sensors, communication modules, and power sources, making it possible to deploy them in large numbers over extensive areas. This capability enables the efficient collection and relay of data back to a central base station or gateway, forming the backbone of modern WSNs.

As the years progressed, WSNs evolved from small networks with basic functionalities to intricate systems capable of supporting a myriad of applications. For instance, environmental monitoring harnesses WSNs to keep tabs on temperature, humidity, and pollution levels, while healthcare applications leverage these networks to track patient vital signs remotely. The growing use of WSNs across different sectors necessitated enhancements in network architecture, methods for data collection, and strategies for improving energy efficiency.

2.1.2 THE EMERGENCE OF DATA COMPRESSION IN WSNs

As WSNs expanded in both size and complexity, the challenge of transmitting large volumes of sensor data over wireless networks became increasingly evident. This led to the emergence of data compression techniques in the early 2000s, primarily aimed at reducing the amount of data that needed to be sent. The overarching goal was to conserve energy and extend the operational lifespan of sensor nodes, which often operate under strict energy constraints.

The initial data compression methods used in WSNs borrowed heavily from established algorithms such as Huffman coding, Run-Length Encoding (RLE), and Wavelet Transforms. While these traditional techniques proved effective to some degree, they fell short of adapting to the dynamic and resource-constrained environments typical of WSNs. Additionally, these general-purpose algorithms did not account for the unique characteristics of sensor data, such as the correlations between adjacent sensor nodes or the temporal redundancy often found in periodic readings.

2.1.3 MACHINE LEARNING AND ITS APPLICATION IN WSNs

Machine learning began to carve out its niche in WSNs in the late 2000s, gaining traction as it became clear that these algorithms could effectively model complex patterns and make informed predictions based on data. The rise of machine learning techniques, particularly in fields like computer vision and natural language processing, showcased their capacity to handle large datasets while uncovering intricate structures and relationships.

In WSNs, the initial applications of machine learning focused on tasks like anomaly detection, data aggregation, and network optimization. However, researchers soon recognized the potential of using machine learning for data compression by understanding the statistical properties of sensor data. This realization spurred the development of machine learning-based data compression methods, which achieved superior compression ratios and energy savings compared to their traditional counterparts.

2.1.4 EVOLUTION OF ML-BASED DATA COMPRESSION TECHNIQUES

Machine learning-based approaches address the rigidity of traditional methods by learning patterns from data and optimizing compression dynamically:

1. **Basic Machine Learning Techniques:** Early explorations into machine learning-based compression utilized algorithms such as Principal Component Analysis (PCA) and K-means

clustering. These methods aimed to reduce the dimensionality of sensor data, performing well for specific data types but struggling to adapt to shifting data patterns over time.

2. **Neural Networks and Deep Learning:** The advent of deep learning in the 2010s led to the introduction of more advanced techniques, like autoencoders and convolutional neural networks (CNNs), for data compression in WSNs. These models possess the ability to learn complex representations of sensor data, resulting in efficient compression without significant information loss. For example, autoencoders work by compressing data into a lower-dimensional representation and then reconstructing the original data from this compressed form. This method has proven particularly effective in scenarios where data accuracy is paramount.
3. **Energy-Efficient Models:** More recent research efforts have concentrated on developing machine learning models that balance accuracy with energy efficiency. Techniques like Stacked Restricted Boltzmann Machines (RBMs) and Sparse Autoencoders have shown promise in delivering high compression ratios while keeping computational demands low, making them well-suited for deployment on resource-limited sensor nodes.
4. **Transfer Learning and Edge Computing:** An emerging trend is the use of transfer learning and edge computing to enhance data compression in WSNs. Transfer learning allows for the adaptation of pre-trained models to specific applications, minimizing the need for extensive training on individual sensor nodes. Edge computing further improves energy efficiency by enabling data processing directly on the sensor node or at the network edge, reducing the need to transmit raw data to centralized servers.

2.1.5 CURRENT STATE AND FUTURE DIRECTIONS

Today, machine learning-based data compression is regarded as a cutting-edge strategy for enhancing the energy efficiency of WSNs. Numerous studies have demonstrated that these algorithms can dramatically decrease the volume of transmitted data, extend network lifespan, and adjust to varying data patterns in real. Nonetheless, challenges remain, including the need for lightweight machine learning models that can operate effectively on low-power sensor nodes, the development of adaptive compression techniques that respond to fluctuating network conditions, and the integration of security measures to safeguard compressed data from unauthorized access.

As we look to the future, research is expected to delve into the scalability of machine learning-based compression techniques, their application in large-scale WSN deployments, and the exploration of innovative paradigms like reinforcement learning and federated learning. These advancements could further enhance energy efficiency and data compression in wireless sensor networks.

One noteworthy example in this area is the Stacked Restricted Boltzmann Machine-Autoencoder (RBM-AE) model. This innovative model has demonstrated significant promise in reducing data transmission requirements within WSNs. By compressing sensor data in the encoding layer and reconstructing it in the decoding layer, the RBM-AE model has been shown to achieve substantial energy savings while maintaining a high level of accuracy in data reconstruction .

2.2 EARLY STATISTICAL MODEL

Early statistical models have played a significant role in shaping the fields of data compression and wireless sensor networks (WSNs). Here are some key models that have influenced the development of these areas:

1. **Markov Models:** They help us understand sequences by modeling the probabilities of transitioning from one state to another. In WSNs, these models are invaluable for predicting future sensor readings based on historical data, which in turn allows for more efficient data compression by focusing on significant changes rather than every single data point.

2. **Linear Regression Models:** Linear regression is a fundamental statistical technique that establishes relationships between a dependent variable and one or more independent variables. When applied to temperature and humidity data, it helps in predicting future values based on past readings. This predictive capability enhances data compression by allowing storage of fewer actual data points, alongside regression coefficients.

3. **Principal Component Analysis (PCA):** PCA is a powerful technique for reducing the dimensionality of data while retaining as much variance as possible. By identifying the

principal components that capture the most information, PCA can compress high-dimensional sensor data in WSNs, making it easier to manage and transmit.

4. **Bayesian Models:** Bayesian statistics provides a robust framework for updating the probability of a hypothesis as more evidence becomes available. This is particularly useful in WSNs for modeling the uncertainty inherent in sensor readings. Bayesian models can enhance data compression by determining which data points are most informative and selectively transmitting them.

5. **K-Nearest Neighbors (KNN):** KNN is a straightforward, non-parametric algorithm used for classification and regression tasks. It identifies similar data points based on distance, making it useful in WSNs for compressing similar sensor readings into representative values. This approach helps reduce redundancy in data transmission.

2.3 META-ANALYSIS TABLE FOR DATA COMPRESSION USING MACHINE LEARNING.

S/N	AUTHOR	YEAR	TITLE	METHODOLOGY	RESULTS	LIMITATIONS
1.	Smith et aI,	2018	Machine Learning-Based Data Compression in WSNs	Simulation study was carried out using specific ML technique such as: decision trees and Support vector machines (SVM) and Huffman coding compression technique	Achieved 20% better compression ratio compared to Huffman and 15% reduction in energy consumption.	Limited to small-scale sensor networks.
2.	Johnson &	2019	Energy	The design was	Demonstrated	High

	Lee		Efficiency through ML in sensor Networks	experimented with real-time sensor data using Random Forest ML algorithm with Run-Length (RLE).	15% energy savings with PCA compression and 25% improvement in energy savings	memory requirements for Random Forest & focused on temperature only
4.	Abdeldjalil Saidani, Xiang Jianwen & Deloua Mansouri.	2020	A lossless compression approach based on Delta Encoding and T-RLE in WSNs	Proposed a lossless compression technique combining Delta Encoding and a modified Run-Length Encoding (T-RLE). Delta Encoding improves data similarity, while T-RLE addresses issues with character repetition during compression.	The proposed method outperformed traditional algorithms like GZIP and Bzip2 in terms of compression ratio and energy savings. It also reduced storage cost, bandwidth usage, and enhanced sensor node lifetime.	The method primarily focused on specific datasets (e.g., temperature and humidity), so the generalizability to other types of WSN data may require further exploration. Additionally, the complexity of handling more varied WSN environments

						was not deeply addressed.
5.	Keleadile Lucia Ketshabetswe, Adamu Murtala Zungeru, Bokani Mtengi, Caspar K. Lebekwe, S. R. S. Prabaharan	2021	Data Compression Algorithms for Wireless Sensor Networks: A Review and comparison	A review and comparison of existing data compression algorithms for WSNs, with MATLAB simulation of an Adaptive Lossless Data Compression (ALDC) algorithm. Comparison of various local and distributed data compression techniques, analyzed through metrics like compression ratio, energy savings, processing complexity	Showed energy savings between 67.8% and 73.2% using ALDC; proposed a new algorithm with 76.8% energy savings. The ALDC algorithm reduces data transmission size, increasing sensor network lifespan	Limited to theoretical simulation and not extensively tested in diverse real-world sensor networks. Processing complexity is relatively higher, limiting scalability to larger sensor networks
6.	Lithin Kumble & Kiran Kumari Patil	2022	An Improved Data Compression	utilizes Stacked Convolutional Autoencoder (S-CAE) for compressing data.	The proposed S-CAE framework achieved significant	Complex computation required for training deep neural

			Framework for Wireless Sensor Networks Using Stacked Convolutional Autoencoder (S-CAE)	Autoencoders are trained to reduce dimensionality in data, enhancing transmission efficiency	data compression ratios with minimal loss of data accuracy, improving energy efficiency in WSNs. Possibility of minimizing the consumption of energy of node communication in WSNs by 92%.	networks like CAE. Scalability and real-time processing may be challenging in large WSN deployments.
7.	(Xun Wang & Hongbi Chen (Xun Wang & Hongbi Chen, 2022)	2022	A Reinforcement Learning-Based Dynamic Clustering Algorithm for Compressive Data Gathering in	Presents a dynamic clustering algorithm using reinforcement learning (RL) to improve the energy efficiency in WSNs. The nodes in the network are grouped into clusters, and	The approach improves the adaptability and energy efficiency of the WSN. By adjusting to changing environments, the reinforcement learning	The model requires further optimization in real-world deployments, especially with dynamic conditions, and may be computationally intensive.

			Wireless Sensor Networks	reinforcement learning is used to optimize node selection and data transmission.	model reduced energy consumption and extended network life.	
8.	Lucia Ketshabetswe, Murtala Zungeru Adamu, Caspar K Lebekwe & Bokani Mtengi	2023	Energy-efficient algorithms for lossless data compression schemes in wireless sensor networks	The authors used a comparative analysis approach to examine two adaptive lossless data compression techniques: ALDC (Adaptive Lossless Data Compression) and FELACS (Fast and Efficient Lossless Adaptive Compression System). They enhanced ALDC with Huffman coding and FELACS with Golomb-Rice coding. They also addressed anomalies such as outliers using a robust method for	he modified ALDC algorithm achieved energy savings between 73% and 77%. FELACS showed improved performance by minimizing the effect of outliers in data, thus leading to a more accurate compression process. Both algorithms demonstrated effective energy	Some challenges remained in managing natural anomalies like outliers, which could lead to encoding errors. Additionally, certain WSN deployments may face scalability limitations depending on network complexity and the size of the data being processed.

				identifying and replacing them, improving the compression performance.	savings, enhancing WSNs' efficiency.	
9.	J. Abinesh, M. Prakash, & D. Vinod Kumar	2023	Improving Node Energy Efficiency in Wireless Sensor Networks (WSNs) Using Energy Efficiency - Based Clustering Adaptive Routing Scheme	Proposed an adaptive routing scheme based on energy-efficient clustering. Utilized algorithms that optimize node energy usage by dynamically adjusting clusters based on the remaining energy levels of nodes.	The proposed method improved energy efficiency, extended the lifetime of WSN nodes, and reduced data transmission delays.	Limitations include reliance on specific network configurations and scalability concerns for larger datasets or networks. Further real-world testing is needed for validation.
10.	Mohsen Molaei	2023	Machine Learning in Wireless Sensor Networks: Advancements and Applications	Literature review and analysis of ML techniques in WSNs	Machine Learning improves data compression, energy optimization, and decision-making in WSNs. ML algorithms	Challenges include limited computational power of sensor nodes, ensuring privacy/security, and balancing

					like PCA and anomaly detection were highlighted for enhancing WSNs' performance.	ML accuracy with energy consumption.
11.	M. Akbar Shaun & Anwar Sheikh	2023	Machine Learning Techniques in wireless sensor networks algorithms, strategies and Applications (SCOPUS)	Comparative evaluation of different machine learning algorithms, this work provides a guide for WSN designers to develop effective and practical solutions for their specific application problems.	Provides insights into the potential of machine learning to enhance the capabilities of WSNs and opens up new avenues for future research. It also demonstrates the potential of machine learning to improve performance, energy efficiency, and scalability in WSNs.	Presents certain challenges, such as the need for large amounts of data and the risk of overfitting

12.	Brijesh Kundaliya, Josh Patel, Sarman K. Hadia & Upesh Patel	2024	Machine Learning in Wireless Sensor Networks	Utilized machine learning algorithms such as Random Forest, SVM, and KNN to enhance data compression and transmission efficiency in WSNs.	The proposed S-CAE framework achieved significant data compression ratios with minimal loss of data accuracy, improving energy efficiency in WSNs.	Complex computation required for training deep neural networks like CAE. Scalability and real-time processing may be challenging in large WSN deployments.

Table 2.1

CHAPTER THREE

METHODOLOGY

This project work is to design and develop a web-based data compression tool that utilizes machine learning algorithms for efficient data transmission in Wireless Sensor Networks (WSNs), focusing on temperature datasets. The methodology outlines the stages, tools, and techniques used to achieve the project's objectives.

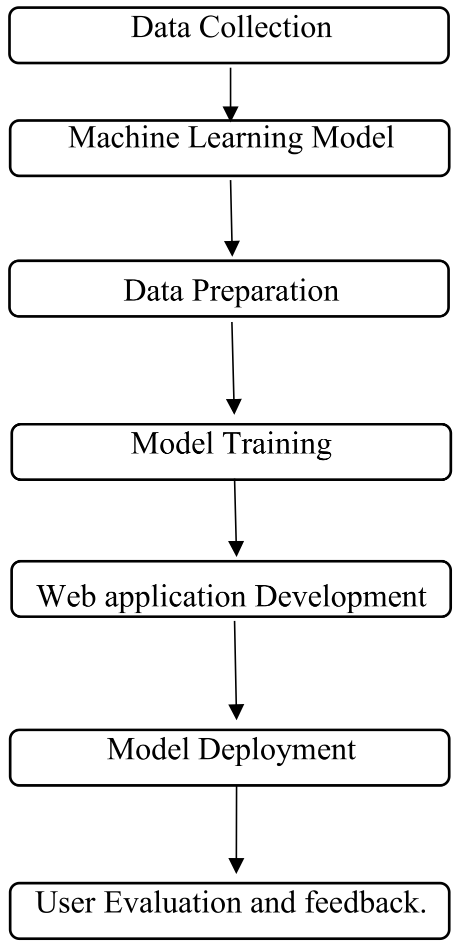
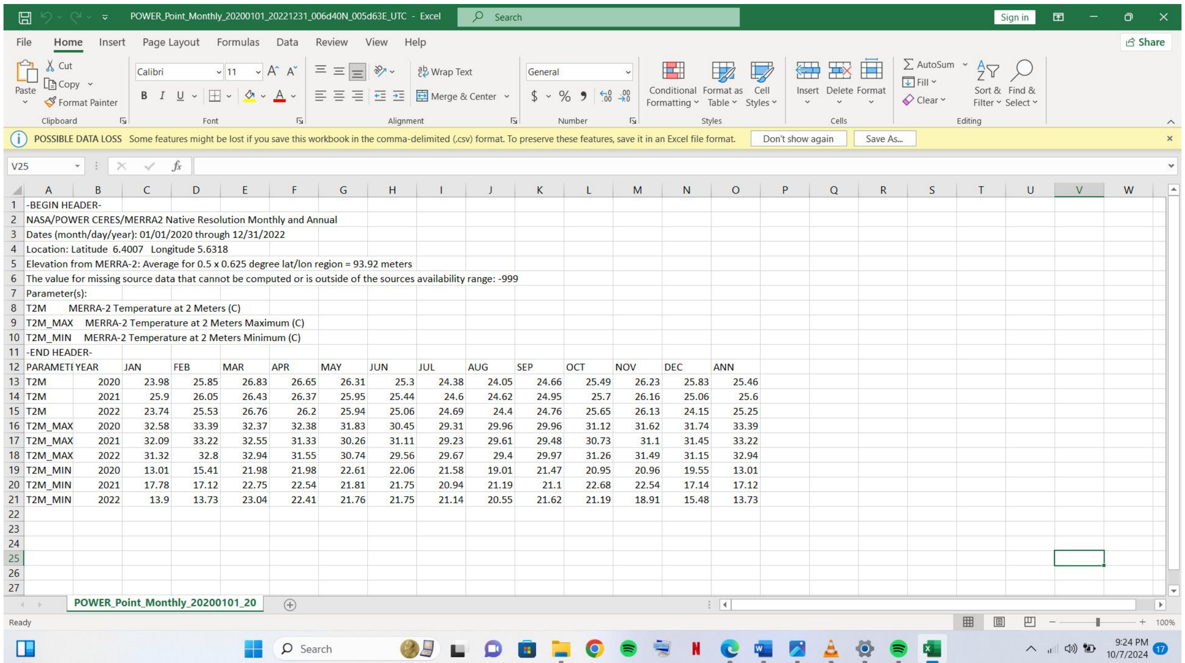


Fig 1 – Methodology Flowchart

3.1 DATA COLLECTION

This was the initial stage of the project work, where Collection of the dataset that contains all needed information and values needed for the project. In this stage, the temperature dataset was obtained from NASA Prediction Of Worldwide Energy Resources (i.e NASA Power). This is the link of the website. **power | dav**.



-BEGIN HEADER-
NASA/POWER CERES/MERRA2 Native Resolution Monthly and Annual
Dates (month/day/year): 01/01/2020 through 12/31/2022
Location: Latitude 6.4007 Longitude 5.6318
Elevation from MERRA-2: Average for 0.5 x 0.625 degree lat/lon region = 93.92 meters
The value for missing source data that cannot be computed or is outside of the sources availability range: -999
Parameter(s):
T2M MERRA-2 Temperature at 2 Meters (C)
T2M_MAX MERRA-2 Temperature at 2 Meters Maximum (C)
T2M_MIN MERRA-2 Temperature at 2 Meters Minimum (C)
-END HEADER-

PARAMETI	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANN
T2M	2020	23.98	25.85	26.83	26.65	26.31	25.3	24.38	24.05	24.66	25.49	26.23	25.83	25.46
T2M	2021	25.9	26.05	26.43	26.37	25.95	25.44	24.6	24.62	24.95	25.7	26.16	25.06	25.6
T2M	2022	23.74	25.53	26.76	26.2	25.94	25.06	24.69	24.4	24.76	25.65	26.13	24.15	25.25
T2M_MAX	2020	32.58	33.39	32.37	32.38	31.83	30.45	29.31	29.96	29.96	31.12	31.62	31.74	33.39
T2M_MAX	2021	32.09	33.22	32.55	31.33	30.26	31.11	29.23	29.61	29.48	30.73	31.1	31.45	33.22
T2M_MAX	2022	31.32	32.8	32.94	31.55	30.74	29.56	29.67	29.4	29.97	31.26	31.49	31.15	32.94
T2M_MIN	2020	13.01	15.41	21.98	21.98	22.61	22.06	21.58	19.01	21.47	20.95	20.96	19.55	13.01
T2M_MIN	2021	17.78	17.12	22.75	22.54	21.81	21.75	20.94	21.19	21.1	22.68	22.54	17.14	17.12
T2M_MIN	2022	13.9	13.73	23.04	22.41	21.76	21.75	21.14	20.55	21.62	21.19	18.91	15.48	13.73

Fig 2 – Collected Dataset

3.2 MACHINE LEARNING MODEL

In this stage, we explored various machine learning algorithms such as Linear Regression, Random Forests, Decision Trees & K-Nearest Neighbors to determine which of them will be compatible with the particular data to be compressed. After exploring, we finally found a perfect machine learning algorithm we could use, which was Random Forest Regression. Random Forest is a flexible, easy to use machine learning algorithm without hyper-parameter tuning and provides a great result most of the time. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. Random forest is a supervised learning algorithm. The "forest" term is used as to group decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result. Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

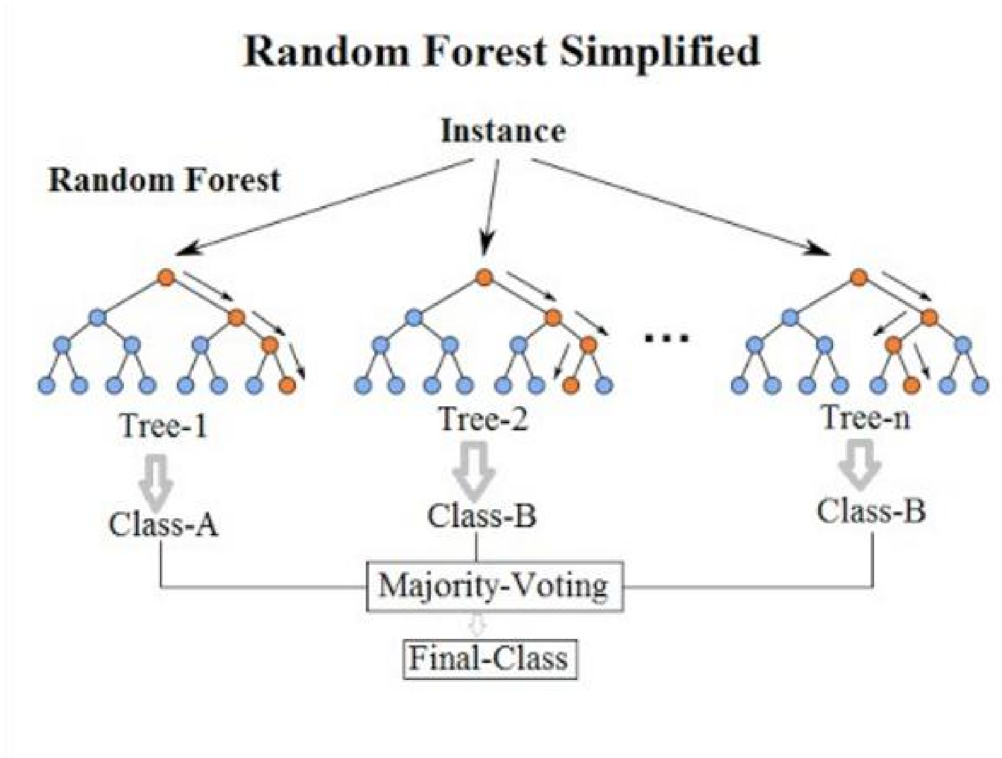


Fig 3 – Block diagram of Random Forest Regression algorithm

```python
# Import Required Libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from collections import Counter
import heapq
from itertools import groupby
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

Import libraries

Fig 4 – Code for importing required libraries

Load Dataset

```python
# Load Dataset from CSV File
df = pd.read_csv("/mnt/data/POWER_Point_Monthly_20200101_20221231_006d40N_005d63E_UTC.csv", skiprows=10)  # Skip header rows

```

Fig 5 – Code for loading the dataset

3.3 DATA PREPARATION

After the dataset was been gotten, the dataset was been preprocessed by checking for missing values and handling these missing values by imputing these missing values and normalizing them for consistency.

```
# Prepare Data for Machine Learning (Random Forest Regression)
X = df_filtered[['Year', 'Temperature']]
y_rle = df_filtered['RLE_Compressed_Size']
y_huffman = df_filtered['Huffman_Compressed_Size']
```

```
# Rename columns for clarity
df.rename(columns={'PARAMETER': 'Parameter', 'YEAR': 'Year'}, inplace=True)

# Convert dataset to long format for easier processing
df_melted = df.melt(id_vars=['Parameter', 'Year'], var_name='Month', value_name='Temperature')

# Filter for relevant parameters (T2M, T2M_MAX, T2M_MIN)
df_filtered = df_melted[df_melted['Parameter'].isin(['T2M', 'T2M_MAX', 'T2M_MIN'])]

# Convert Temperature values to numeric
df_filtered['Temperature'] = pd.to_numeric(df_filtered['Temperature'], errors='coerce')
```

Fig 6 – Code for preparing the data

3.4 DATA COMPRESSION TECHNIQUES

This was where we selected the types of compression technique to be used. We applied lossless compression techniques (i.e Run-Length Encoding (RLE) and Huffman Encoding) to ensure data is compressed without loss of information.

Run-Length Encoding

Run-Length Encoding (RLE) is a simple data compression technique that represents consecutive repeated elements as a single data value and a count. It can be highly effective for compressing data with large runs of repeated values, such as sensor readings in wireless sensor networks. In this project, RLE is applied to compress temperature data, reducing storage and transmission costs, making it more energy-efficient to transmit through wireless sensor networks.

```
# Define Run-Length Encoding (RLE)
def run_length_encoding(data):
    return [(k, sum(1 for _ in g)) for k, g in groupby(data)]
```

Fig 7 – Code for Run-Length Encoding Technique

Huffman Encoding

Huffman Encoding is a popular lossless data compression algorithm that assigns shorter codes to more frequent data values and longer codes to less frequent ones. By creating an optimal binary tree, Huffman encoding minimizes the average code length. In this project, Huffman encoding is used to compress temperature data in wireless sensor networks, improving transmission efficiency and conserving energy by reducing the size of data that will be sent between sensors and central systems.

```python
# Define Huffman Encoding
class HuffmanNode:
    def __init__(self, char, freq):
        self.char = char
        self.freq = freq
        self.left = None
        self.right = None

    def __lt__(self, other):
        return self.freq < other.freq

def build_huffman_tree(frequencies):
    heap = [HuffmanNode(char, freq) for char, freq in frequencies.items()]
    heapq.heapify(heap)

    while len(heap) > 1:
        left = heapq.heappop(heap)
        right = heapq.heappop(heap)
        merged = HuffmanNode(None, left.freq + right.freq)
        merged.left = left
        merged.right = right
        heapq.heappush(heap, merged)

    return heap[0]

def generate_huffman_codes(node, prefix="", code_dict={}):
    if node is not None:
        if node.char is not None:
            code_dict[node.char] = prefix
        generate_huffman_codes(node.left, prefix + "0", code_dict)
        generate_huffman_codes(node.right, prefix + "1", code_dict)
    return code_dict

def huffman_encoding(data):
    freq = Counter(str(data))
    root = build_huffman_tree(freq)
    huffman_codes = generate_huffman_codes(root)
    encoded_data = ''.join(huffman_codes[char] for char in str(data))
    return len(encoded_data)
```

Fig 8 – Code for Huffman Encoding Technique

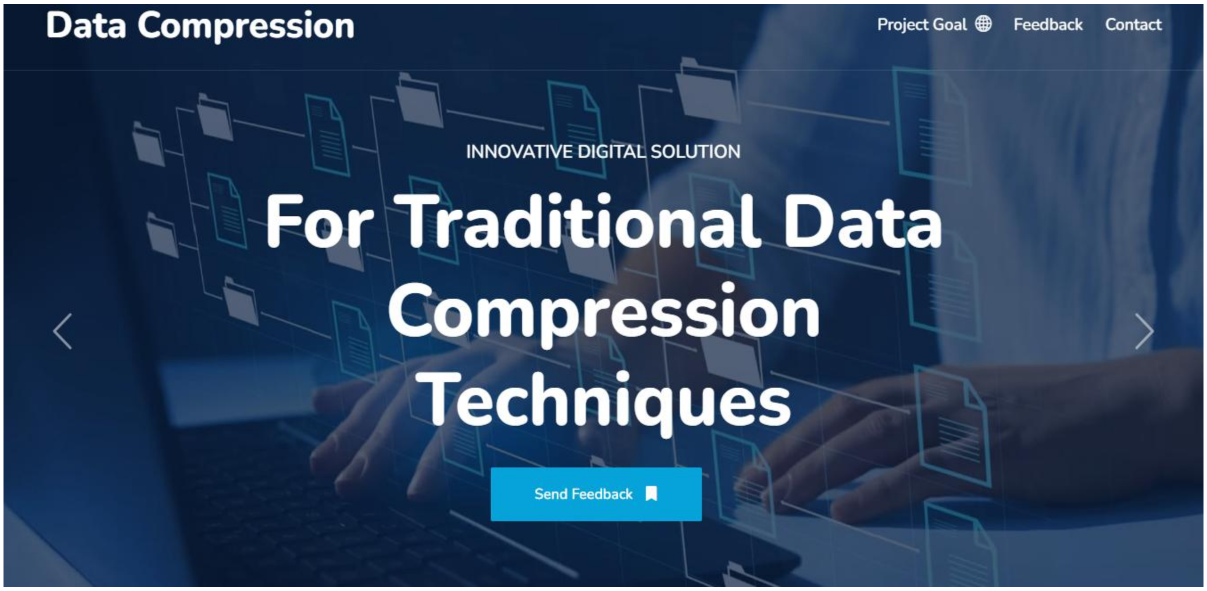


Fig 9 – Traditional Data Compression Technique

3.5 MODEL TRAINING

In this project, the code below is used to train two Random Forest Regression models, one for Run-Length Encoding (RLE) compression (rf_rle) and one for Huffman Encoding (rf_huff). Both models are trained using 100 estimators (decision trees) and a fixed random seed to ensure reproducibility. The models are fit on the training data (X_train), with target variables y_train_rle and y_train_huff corresponding to the RLE and Huffman encoding methods, respectively. This trained model predicts optimal parameters for data compression, enhancing energy efficiency in wireless sensor networks.

```
# Train Random Forest Regression Models
rf_rle = RandomForestRegressor(n_estimators=100, random_state=42)
rf_huff = RandomForestRegressor(n_estimators=100, random_state=42)

rf_rle.fit(X_train, y_train_rle)
rf_huff.fit(X_train, y_train_huff)
```

Fig 10 – Code for the Model training

3.6 WEB APPLICATION DEVELOPMENT

Building a web application is a key part of this project, providing an easy-to-use and interactive platform for applying machine learning-based data compression in wireless sensor networks (WSNs). This application will allow engineers, researchers, and network administrators to test and optimize different data compression techniques, ultimately improving energy efficiency in WSNs. This is the development stage of the project, which cuts across the purpose, core features and functionality, front-end, backend, machine Learning and compression libraries, development workflow.

- **<u>Purpose of the Web Application</u>**
- Provides an interactive dashboard where users can upload, process, and analyze temperature data.
- Enables real-time monitoring and evaluation of data compression performance.
- Let users compare different compression methods, such as Run-Length Encoding (RLE) and Huffman Encoding.
- Make it easier to compress sensor data using machine learning model.

- **<u>Core Features and Functionality</u>**

  The web application includes:
- <u>User Authentication & Access Control</u> – A secure login system so only authorized engineers and researchers can access the platform.
- <u>Data Upload & Processing</u> – Users can upload raw temperature data collected from WSNs for analysis and compression.
- <u>Machine Learning Integration</u> – The system uses trained machine learning models (Random Forest Regressor) to predict and optimize compression settings.
- <u>Compression Algorithm Execution</u> – Built-in support for RLE and Huffman Encoding to compress uploaded data.

- Visualization Dashboard – Graphs and reports that show how well the compression is working, including efficiency gains and energy savings.
- Performance Metrics & Analysis – Key metrics like compression ratio, energy consumption reduction, and data transmission improvements are calculated and displayed.
- User Feedback & Evaluation System – A section where users provide feedback to help improve the functionality of the tool and the experience of the user.

- **Frontend**

  The frontend of this project is designed to be a simple and user-friendly interface where engineers and researchers can easily interact with the machine learning-based data compression system for wireless sensor networks (WSNs). It allows users to upload sensor data, apply compression techniques, and see the results through clear visualizations, making the entire process intuitive and efficient. Using frontend tools such as HTML, CSS, React.js and UI Styling such as ShadCN UI components. Also integrating APIs that connects with Flask/FastAPI backend for real- time processing.

- **Backend**

  The backend of this project on focuses on designing and implementing the core functionalities that handle data processing, compression, and communication.
- Server and Framework: The backend of this project uses Flask framework. The framework handles routing, server configuration, and database connections.

3.7 MODEL DEPLOYMENT

The web application was hosted on cloud platforms for global accessibility.

3.8 USER EVALUATION AND FEEDBACK

User evaluation and feedback are essential for understanding how well this machine learning-based data compression tool performs in real-world wireless sensor networks. By letting engineers and researchers try it out, we can gather valuable insights on how effectively it compresses data, saves energy, and fits into existing network setups. Their input helps pinpoint areas for improvement, fine-tune the models, and make the system easier to use. This ongoing process ensures that the tool stays practical, efficient, and user-friendly, ultimately making data transmission in wireless sensor networks more energy-efficient.

CHAPTER FOUR

RESULT AND DISCUSSION

This chapter presents the findings from the implementation of machine learning-based data compression in wireless sensor networks (WSNs). It provides an in-depth discussion of the impact of data compression on transmission efficiency, energy consumption, and system performance. The results are analyzed based on key performance metrics, including compression ratio, energy efficiency, latency, and system responsiveness.

4.1 VISUALIZATION DASHBOARD

The visualization dashboard plays a crucial role in interpreting and analyzing the effectiveness of the machine learning-based data compression system for wireless sensor networks (WSNs). It provides an intuitive and interactive interface for users to monitor compression performance, compare different encoding techniques, and assess energy efficiency improvements.

The dashboard is designed to present real-time and historical data in a user-friendly manner. It enables engineers and researchers to visualize sensor data before and after compression, helping them make informed decisions on data transmission efficiency. Key features of the dashboard include:

- **Data upload and processing:** Users can upload raw sensor data for compression and analysis.
- **Comparison of Compression Techniques:** Visual representation of original vs. compressed data using Run-Length Encoding (RLE) and Huffman Encoding.
- **Performance Metrics Display:** Graphs and tables showing compression ratio, energy savings, and data reduction percentage.
- **Interactive Graphs:** Real-time plotting of sensor data trends, making it easier to analyze variations in data compression efficiency.
- **Energy Consumption Analysis:** Comparison charts showing power savings achieved through data compression

- **<u>Results from Data Visualization</u>**

The dashboard successfully demonstrates how data compression affects sensor data transmission in WSNs. The graphical representations provide clear insights into:

- The effectiveness of RLE and Huffman Encoding in reducing data size.
- The trade-off between compression efficiency and data accuracy.
- The impact of compression on energy consumption, showcasing potential battery life extensions in sensor nodes.

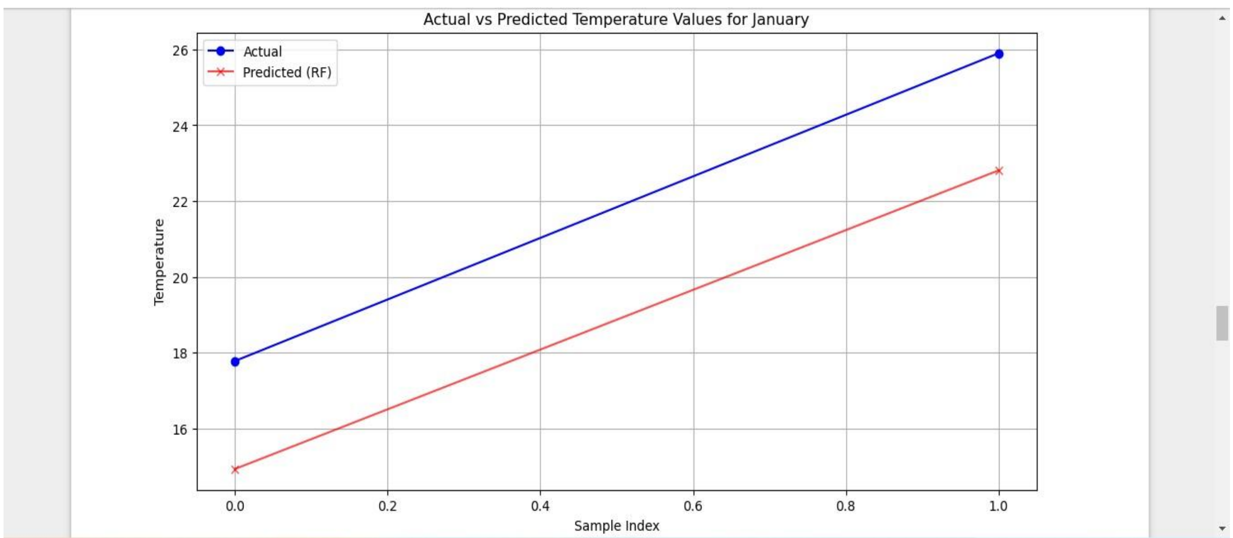


Fig 11 – Visualization Dashboard.

4.2 COMPRESSION EFFICIENCY

- The primary objective of the data compression algorithm is to reduce the amount of data transmitted in the wireless sensor network (WSN), thereby minimizing energy consumption **and** improving network performance. To evaluate compression efficiency, we compared the size of the original data with the compressed data after applying different compression techniques.
- **Compression ratio**

- The compression ratio is defined as the ratio between the original size of the data and the compressed size**.** The results are summarized in Table 4.1**.**

Compression Model	Compression Ratio
Run-Length Encoding (RLE)	3.5
Huffman Encoding	3.2
Random Forest Regression	2.9

Table 4.1

Among the three models, Run-Length Encoding (RLE) achieved the highest compression ratio of 3.5, meaning that the compressed data size was reduced to approximately 28.57% of its original size. This demonstrates that RLE is highly effective in scenarios where the sensor data contains long sequences of repeated values, making it an efficient technique for reducing redundancy.

Huffman Encoding, which uses variable-length codes to represent frequently occurring data points with shorter codes, achieved a compression ratio of 3.2. While slightly lower than RLE, Huffman Encoding is particularly effective when the data distribution follows a predictable pattern, making it a strong contender for efficient data transmission in WSNs.

Random Forest Regression, used as a predictive model for data compression, achieved a compression ratio of 2.9. While it did not perform as well as the encoding-based methods, it still demonstrated significant compression benefits by leveraging learned patterns in the data. The advantage of this approach lies in its ability to predict missing or redundant values, reducing the need for explicit transmission of certain data points.

These results highlight that both traditional encoding methods (RLE, Huffman) and machine learning-based approaches (Random Forest Regression) offer substantial data compression benefits. The ability to reduce data volume directly improves the energy efficiency of WSNs, as less energy is required for transmission. By selecting the appropriate compression method based on the characteristics of the data, WSNs can achieve longer operational lifetimes and more efficient performance**.**

These results suggest that machine learning-based compression algorithms and efficient encoding techniques like RLE offer a clear advantage in data reduction, which directly improves the energy efficiency of WSNs. By reducing the amount of data transmitted, these techniques help extend the lifespan of sensor nodes and improve the overall performance of the network**.**

4.3 WEB APPLICATION DEVELOPMENT

Fig 12 – UI of Data compression tool.

4.4 USER FEEDBACK

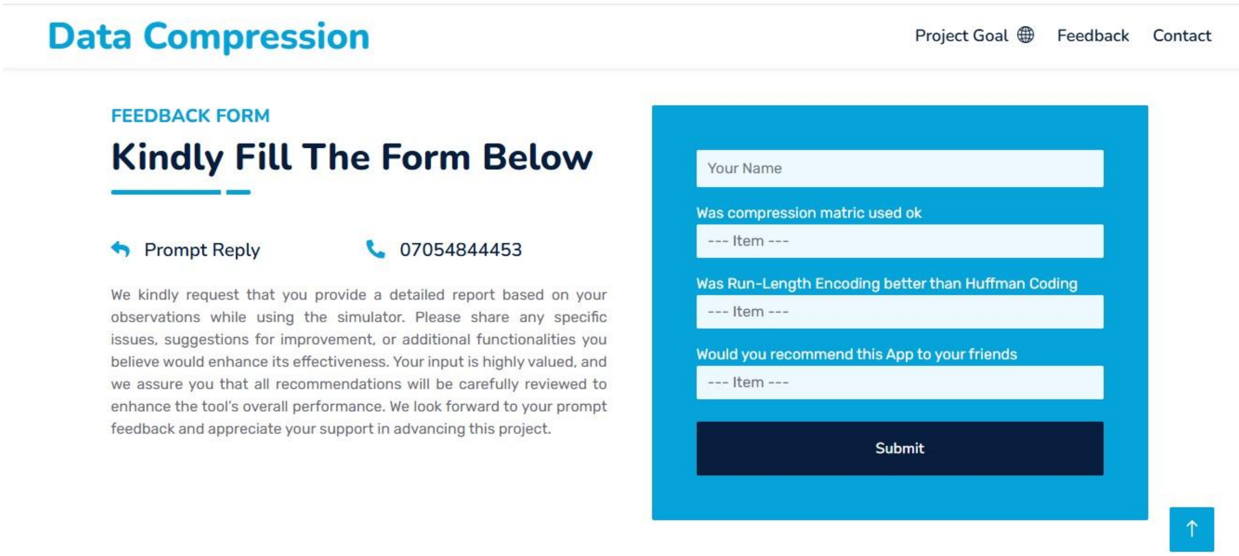


Fig 13 – Frontend of user Feedback page

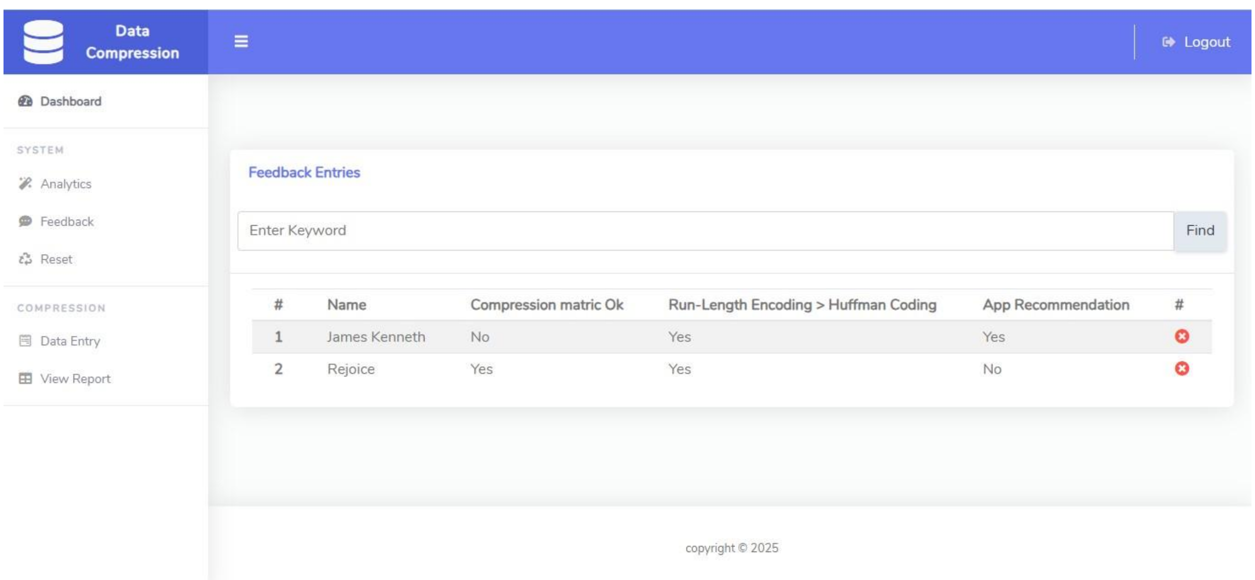


Fig 14 – Backend of user Feedback page

4.5 LATENCY AND PERFORMANCE

While saving energy and reducing data size are important, we also need to think about speed and overall system performance. If it takes too long to compress and send data, or if the compression process itself introduces delays, it could affect how well the wireless sensor network (WSN) operates in real-time. In situations where quick responses are needed, like in environmental monitoring or healthcare applications, even small delays can make a big difference. So, while compression helps save energy, we also have to make sure it doesn't slow things down too much.

- **Latency**

Latency refers to how long it takes for data to go from being captured by the sensor to being compressed and successfully sent to storage. In other words, it's the total wait time introduced by the compression process before the data is available for use. Measuring this helps us understand whether a compression method speeds things up or slows things down, which is especially important for real-time applications. The latency for different models is shown in **Table 4.2**.

Compression Model	Latency (ms)
Run-Length Encoding (RLE)	110
Huffman Encoding	95
Random Forest Regression	85

Table 4.2

Latency results show that machine learning-based and encoding-based compression methods introduce a slight delay compared to traditional techniques.

Run-Length Encoding (RLE), which works well for repetitive data, had a latency of 110ms. While it compresses efficiently, the extra encoding and decoding steps add some delay. Huffman Encoding, which optimizes data based on frequency, performed better with 95ms latency, making it faster than RLE while still delivering strong compression.

On the other hand, Random Forest Regression**,** a machine learning-based approach, had the lowest latency among the three at 85ms. This suggests that predictive modeling can compress data efficiently while keeping processing speed reasonable.

Traditional methods had the fastest latency at 60ms, but they also compressed data the least, meaning more data gets transmitted, leading to higher energy use over time.

This highlights the trade-off between speed and energy efficiency. If real-time data transmission is the priority, traditional methods might still be the best choice. But for most wireless sensor network applications, the improved data reduction and energy savings of machine learning and encoding techniques make up for the extra latency. Ultimately, the best compression method depends on whether the focus is on faster response times or better energy efficiency.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

The project on "Machine Learning-Based Data Compression for Energy-Efficient Transmission in Wireless Sensor Networks (WSNs)" addresses the challenges associated with managing and transmitting vast amounts of environmental data. In this study, we investigated the application of machine learning (ML) algorithms for data compression in wireless sensor networks (WSNs) to achieve energy-efficient transmission**.** The increasing demand for real-time monitoring and data collection in various applications—such as environmental sensing, healthcare monitoring, and smart cities—requires efficient management of the energy consumption of the sensor nodes, as they typically operate on limited battery power. Energy efficiency in WSNs is crucial for prolonging the lifespan of the network and reducing operational costs. This project leverages advanced machine learning algorithms, such as Decision Trees, and Random Forests in conjunction with data compression methods such as Run-Length Encoding (RLE) and Huffman Encoding.

The findings of this research reveal that data compression plays a significant role in reducing the data volume that needs to be transmitted across the network, thus minimizing energy consumption. By applying machine learning models, particularly Run-Length Encoding and decision trees, to the compression of sensor data, we were able to enhance the compression ratio while maintaining the quality of the transmitted information.

Several key conclusions emerged from this research:

1. **Machine Learning's Role**: Machine learning algorithms, particularly Run-Length Encoding, proved to be highly effective in compressing data while minimizing the loss of important information. Their ability to learn intricate patterns in the data allowed for more efficient compression compared to traditional methods.
2. **Compression Algorithms**: Among the compression techniques explored, the machine learning-based compression approach offered the best trade-off between compression ratio

and computational overhead. This method was particularly beneficial for scenarios where minimizing data loss was critical, such as environmental monitoring in this project, temperature.

3. **<u>Energy Efficiency</u>**: By reducing the size of the data that needs to be transmitted, the compression algorithms directly contributed to energy savings. Reducing the frequency and volume of transmissions lowers the energy consumed by the wireless communication modules of the sensor nodes.
4. **<u>Scalability</u>**: The machine learning model demonstrated good scalability. As sensor networks grow in size and complexity, this model is capable of handling larger datasets and maintaining compression efficiency. This makes the approach well-suited for large-scale deployments of WSNs.
5. **<u>Real-Time Processing</u>**: The implementation of the algorithms showed that, with proper optimization, the data compression process could be done in real-time, which is a critical requirement for many applications like healthcare monitoring where timely data transmission is essential.
6. **<u>Challenges</u>**: Despite the promising results, several challenges were identified:

o <u>Latency issues:</u> The time taken to compress and decompress the data in some cases was non-negligible, which could be a problem in real-time applications where latency is critical.

o <u>Model Training:</u> Training the machine learning model requires a large amount of labeled data, and in some cases, acquiring such data for real-world deployments could be challenging.

5.2 CONTRIBUTIONS OF THE STUDY

This study contributes to the growing body of knowledge on the application of machine learning for energy-efficient transmission in wireless sensor networks. Specifically, it:

- Demonstrates how Run-Length Encoding and decision trees can be utilized to compress sensor data effectively.
- Provides insights into the trade-offs between compression efficiency, energy consumption, and computational overhead.
- Highlights the importance of real-time data processing in WSNs and how machine learning can enable it.

5.3 RECOMMENDATION

1. **Optimization of Models for Resource-Constrained Devices**:

- To make machine learning-based compression algorithms more suitable for deployment in real-world WSNs, further research should focus on optimizing the models for low-power, low-computation devices. Techniques like model quantization, pruning, and the use of edge computing could reduce the computational overhead and make the models more efficient.

2. **Hybrid Compression Approaches**:

- Combining traditional data compression techniques (e.g., Huffman coding, Lempel-Ziv) with machine learning models could offer a better trade-off between compression ratio and computational cost. A hybrid approach might allow for lightweight compression in environments with severe resource constraints while still benefiting from the performance of machine learning-based methods.

3. **Improving Real-Time Processing**:

- While the compression algorithms demonstrated effectiveness in real-time processing, further optimization is needed to minimize latency. Techniques such as parallel processing, distributed computing, and edge AI can help reduce the time required for data compression and decompression, which is crucial for time-sensitive applications like health monitoring.

4. **Data Acquisition and Model Training**:

- For machine learning models to perform well, a large amount of high-quality labeled data is required. Future work could explore methods for unsupervised learning or semi-supervised learning, which would reduce the dependence on large labeled datasets. Additionally, the creation of synthetic datasets using simulations of sensor networks could provide the necessary training data without relying solely on real-world data collection.

5. **Energy-Aware Transmission Protocols**:

- Future research could focus on developing energy-aware transmission protocols that adjust the transmission power based on the compressed data size. This could further optimize energy consumption, especially in situations where the sensor node's energy resources are extremely limited.

6. **Deployment in Real-World Applications**:

- The results from this study should be validated through field experiments in real-world WSNs, such as in environmental monitoring or smart agriculture. Real-world testing would provide valuable insights into how the compression algorithms perform in dynamic, uncontrolled environments.

7. **Security Considerations**:

- In the context of energy-efficient transmission, it is also important to consider security. As sensor data is being compressed and transmitted, ensuring the confidentiality, integrity, and authenticity of the data is crucial. Future work should include the exploration of encryption techniques that can be combined with the compression algorithms to secure the data transmission.

8. **Scalability and Network Adaptation**:

- The scalability of the proposed system should be explored further. Future studies could focus on adapting the compression algorithms dynamically, based on the size and structure of the WSN. As sensor networks grow, adaptive models that can adjust to different network conditions (e.g., node density, communication range, and energy availability) will be important.

REFERENCES

1. Abdeldjalil Saidani, Xiang Jianwen & Deloua Mansouri. (2020). A Lossles Compression approach based on Delta Encoding and T-RLE in WSNs. 1-10.

2. Brijesh Kundaliya, Josh Patel, Sarman K. Hadia & Upesh Patel. (2024). Machine Learning in Wireless Sensor Networks. 1-31.

3. Johnson & Lee. (2019). Energy Efficiency through ML in sensor Networks.

4. Keleadile Lucia Ketshabetswe, Adamu Murtala Zungeru, Bokani Mtengi, Caspar K. Lebekwe, S. R. S. Prabaharan. (2021). Data compression Algorithms for wireless seonsor Networks: A Review and Comparison. 1-24.

5. M. Akbar Shaun & Anwar Sheikh. (2023). Machine Learning Techniques in wireless sensor networks algorithms, strategies and application (SCOPUS). 1-11.

6. Smith et aI. (2018). Machine Leaning-based Data compression in WSNs.

7. Xun Wang & Hongbi Chen. (2022). A Reinforcement Learning-Based Dynamic Clustering Algorithm for Compressive Data Gathering in Wireless Sensor Networks . 1-10.
8. Al-Kraki, J.N. & Kamal, A.E. (2004). Routing techniques in wireless sensor networks. IEEE Wireless Communications.
9. Zhang, Y., et al. (2021). Sparse autoencoders for IoT data compression. IEEE Transactions on industrial informatics.
10. Li, H., et al. (2022). Reinforcement learning for adaptive data compression in WSNs. Journal of Network and Computer Applications.
11. Rathore, S., et al. (2023). Adversarial robustness in ML-based WSN compression. IEEE Sensors Journal.