DEEP LEARNING APPROACH FOR EFFICIENT ENERGY CONSUMPTION AND HIGH THROUGHPUT IN MOBILE WIRELESS SENSOR NETWORKS

Nada Mohammed Alsalmi, BSc (Hons), MRes
School of Computing and Communications
Lancaster University

A thesis submitted for the degree of
Doctor of Philosophy

July, 2024

I dedicate this thesis to my late parents. Oh God, forgive them and have mercy on them. Your loving upbringing and wise guidance have deeply shaped who I am today. Even though you are no longer here, I feel your spirit and your teachings guiding me every step I take. I am forever grateful for the solid foundation you provided and the lasting values you instilled in me.

Declaration

I declare that the work presented in this thesis is, to the best of my knowledge and belief, original and my own work. The material has not been submitted, either in whole or in part, for a degree at this, or any other university. This thesis does not exceed the maximum permitted word length of 80,000 words including appendices and footnotes, but excluding the bibliography. A rough estimate of the word count is: 33979 .

Nada Alsalmi

Abstract

Features such as scalability, smaller size, simplicity, low-cost operation, self-organization abilities, and easy and fast deployment are the main parameters of a Wireless Sensor Network (WSN). The research demand is growing on WSNs, and therefore, areas under agriculture, industry, healthcare, manufacturing, security, surveillance, transport, air quality, water quality, etc., have started to possess the attributes of WSNs.

The primary goal of SNs is to collect the data from the area of interest and communicate it to the sink or base station (BS) for further processing via single or multi-hop transmission. Sometimes, the BS acts as a gateway to the Internet of Things (IoT), where the IoT can communicate the data to the Cloud using the Internet. The battery-equipped SNs consume more energy for heavy data transmission. Transmission of high-quality data in SN makes the battery-equipped micro-sensors consume much energy.

Mobile Wireless Sensor Network (MWSN) represents a fast-evolving technology, and its use in many things is not limited. While fixed-infrastructure networks constrain sensor nodes to one specific location, MWSNs allow the partial nodes or all nodes to move wherever they want and communicate between themselves, making the whole system more flexible. Furthermore, MWSNs can be compared with respect to GPS, Bluetooth Low Energy (BLE), and existing wireless sensor networks in aspects of extended network lifespan, energy saving, multiband functionality, and high targeting. Nevertheless, pathfinding in MWSNs is very challenging since the sensor nodes are mobile, low-cost devices that are time-constrained, allowing limited resources to be used. On the mobile network, this unique frequency scheme creates extra difficulty in routing. In most monitoring applications, only partial nodes need to be moved in the network. Such nodes are called mobile agent sink nodes or sensor nodes. In the present work, the movement of only a few nodes is considered in MWSN.

Energy consumption and network connectivity are two major issues in MWSNs. Several studies have been conducted to develop and propose suitable solutions for these problems. Many researchers are working to develop the best solutions due to the severe problems with energy consumption and network connectivity in mobile wireless sensor networks. To investigate network connectivity, this study introduces a new efficient technique that considers parameters like network stability, detection area, low energy consumption, etc. This approach guarantees network connectivity, communication sustainability, and the highest level of energy consumption optimization.

This research investigates network connectivity issue and proposes two routing algorithms, namely Self-Organizing Maps based-Optimized Link State Routing (SOM-OSLR) and Deep Reinforcement Learning based-Optimized Link State Routing (DRL-OLSR) for MWSNs. Both algorithms undertake the relationship between sensor node deployment, communication radius, and detection area and

suggest a new way to maintain communication while optimizing energy usage. I have evaluated both algorithms through simulations by considering various performance metrics such as connection probability, end-to-end delay, overhead, network throughput, and energy consumption.

The network is analyzed for proposed routing and aggregation methods to analyze the performance. The simulation analysis is discussed under three scenarios. The first scenario undertakes 'no optimization,' the second considers SOM-OLSR, and the third undertakes DRL-OLSR. The simulation results indicate that the SOM-OLSR performs better compared to the case with 'no routing' optimization. Comparing DRL-OLSR and SOM-OLSR indicates that the former outperforms the latter in terms of low latency and high network lifetime. Specifically, the DRL-OLSR achieves a 47% higher throughput and 67% lower energy consumption compared to the SOM-OLSR. In addition, when compared to the 'No optimization' condition, the DRL-OLSR achieves a notable 69.7% higher throughput and almost 89% lower energy consumption. These findings highlight the effectiveness of the DRL-OLSR approach in optimizing network performance and energy efficiency in wireless sensor networks. Similarly, data aggregation consistently reduces energy consumption across all scenarios, with up to 50% lower as compared to without data aggregation.

Publications

The following Publication have been published in major conference while developing this thesis.

- Nada Alsalmi, Keivan Navaie and Hossein Rahmani , Deep Reinforcement Learning based Energy-Efficient Aggregation Model for Wireless Sensor Network, 3th International Conference on Computing and Communication Networks,(ICCCNet-2023).
  urlhttps://www.research.lancs.ac.uk/portal/en/publications/- (50739e41-a739-4fe9-a867-d364443f7173).html

- Nada Alsalmi, Keivan Navaie and Hossein Rahmani , Energy and throughput efficient mobile wireless sensor networks: A deep reinforcement learning approach, IET Networks,https://doi.org/10.1049/ntw2.12126

Acknowledgements

I am deeply grateful to (Almighty God), Allah, for granting me the strength, knowledge, and determination to overcome the challenges I faced during my studies, and for bestowing upon me the ability to successfully complete my job.

In loving memory of my late parents, Mr. Mohamed and Mrs. Hamda, I dedicate this work to honor their enduring influence in my life. Their support and unwavering guidance have shaped my journey, and I am deeply grateful for the lasting impact they have had on my personal and academic growth.

To begin, I want to offer my sincere gratitude to Professor Keivan Navaie for everything he has done for me during my doctoral studies, including his all-encompassing support and very genuine care, as well as his advice and inspiration all through my Ph.D.

Second, I'd like to express my gratitude to Dr. Hossein Rahmani for his assistance as a co-supervisor. It's an honor for me to be a student at Lancaster University.

I'd like to extend my heartfelt appreciation to my husband, Mohammed, whose unwavering support has been a constant throughout every phase of my life. I am acutely aware that this work owes its existence to his presence and the sacrifices he has made.

I wish to convey my profound appreciation to my sisters and brothers for their unwavering faith in me, along with their continuous encouragement and support. They have been a constant source of inspiration, and I am deeply grateful to each and every one of them.

A special thanks to my biggest sister, Mrs. Rahmah, who became like a second mother to me. Her love, support, and guidance throughout my educational journey have been invaluable. Her encouragement and belief in me have made all the difference, and I am profoundly grateful for everything she has done. gratitude to each one of them.

I wish to extend my deepest appreciation to my cherished daughters, Mayar and Tulin, along with my son, Othman. Their smiles, and even their tears, have served as a powerful source of motivation throughout the completion of this work. They have accompanied me through both the triumphs and challenges of this journey, demonstrating a level of understanding and sacrifice far beyond their years.

I extend my thanks to Jeddah University, Saudi Arabia, for awarding me a scholarship that made it possible for me to pursue my studies in the UK.

Finally, I would like to express my gratitude to my colleagues, friends, and everyone at the School of Computing and Communications at Lancaster University for their cooperation and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Wireless Sensor Networks (WSNs) are essential enablers for emerging applications such as smart homes and cities, healthcare monitoring, surveillance, and disaster management systems. A WSN consists of many sensor nodes distributed in the area of interest for collecting various types of data as shown in Figure 1.1 (Sekhar et al., 2021), (Sah and Amgoth, 2018).

A typical sensor node (SN) includes sensing, processing, and communication modules. SNs transmit the collected data to a Base Station (BS) or a sink node directly or through multi-hop communication (J. Liang et al., 2021). In most practical applications, sensor nodes consist of limited energy supply, such as non-rechargeable batteries (Praveen Kumar, Tarachand, and Rao, 2019), (Banoth, Donta, and Amgoth, 2021). Thus, these nodes often operate within energy-constrained situations. Hence, energy consumption should be sensibly accomplished to confirm the effective utilization and efficient performance of basic operations, including sensing, processing, and communications (Praveen Kumar, Tarachand, and Rao, 2019), (Banoth, Donta, and Amgoth, 2021 Chang, H. Tang, et al., 2017).

For mobile wireless sensor networks (MWSN), achieving high coverage and connectivity is very challenging (Banoth, Donta, and Amgoth, 2023). In MWSN, the sensor nodes can move within the network. In practical scenarios, generally the partial sensor nodes only moves in the network.

In the context of agriculture monitoring or any other monitoring application using MWSN, a limited deployment of a particular percentage of the Mobile Agent sensor nodes is a strategic decision trying to strike a balance between the resource constraints and the effectiveness of the network. With the unification of stationary sensor nodes and mobile agents, a network is capable of dynamically adjusting to the agricultural field's changing conditions. Despite using only a partially deployed mobile node network, networks obtain good benefits including dynamic data collection and localized event detection. Mobile agents that can move around

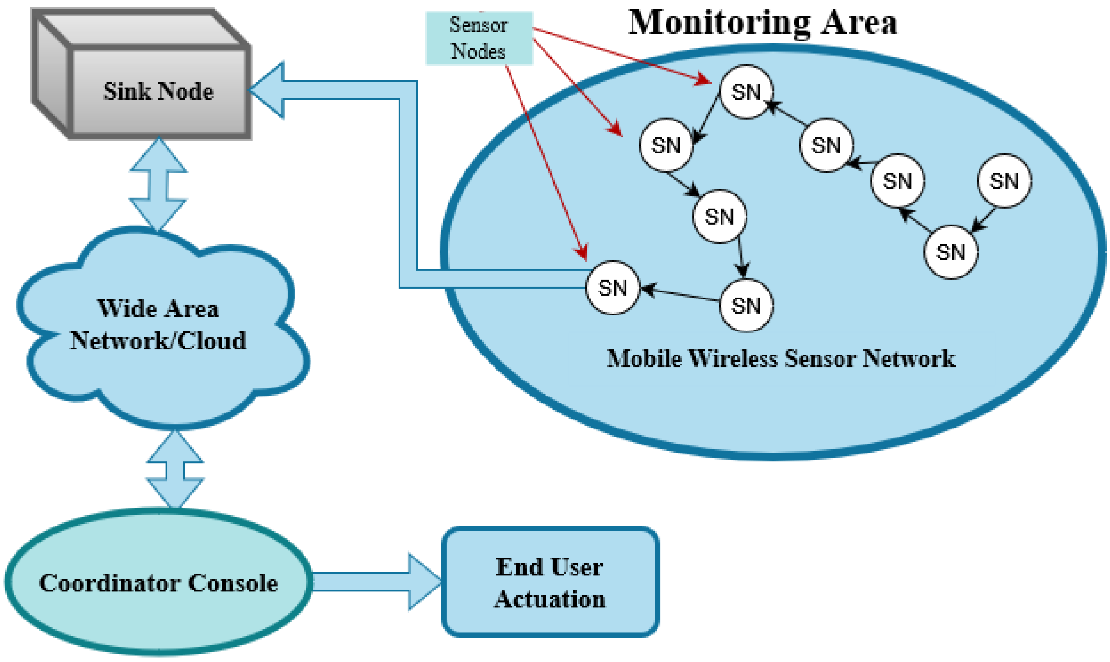


Figure 1.1: Architecture of MWSN (Mobile Wireless Sensor Network) (Sutton and Barto, 2018)

the agricultural area collect data at different points and then transmit the data back to the base station, improving the monitoring process as compared to the use of only stationary nodes. It is the dynamic data collection ability that makes the network react quickly to the occurrence of localized events, e.g., soil moisture changes or pest outbreaks, improving the management techniques of agriculture.

The selective allocation of mobile agent routers in the network increases energy efficiency. By optimizing their routes and motions, mobile nodes save energy while ensuring maximum coverage and data gathering. Through a partial deployment strategy, the network strives for a balance between data collection demands and power consumption maintaining the network longer in operation. Moreover, as nodes are mobile, they have the ability to change their positions in response to the dynamic conditions in their environment. They occupy areas of interest or reposition to where data collection is most of the essence. It, thus, makes the network highly adaptive in such a manner that it tremendously increases the efficiency and effectiveness of the networks without relying on many resources. This allows for continuous monitoring of the agricultural environment with minimal expenditure of resources.

Finally, combined mobile agents' nodes with fixed sensors improve the coverage and data granularity over the agricultural field. Although a partial deployment of mobile sensor nodes allows the network to fill the coverage gaps and offer more extensive monitoring than depending only on fixed sensors, it looks more cost-effective to implement that together with stably operated stationary sensors in reality. An increased area covered makes a deeper perception of the agricultural environment possible and enables farmers and administrators to take relief measures considering irrigation, fertilization, pest control, and production of commodities. With the partial deployment of mobile agent nodes, a MWSN results in a significant reduction of deployment cost and allows one to enjoy the benefits of mobility and coverage. Consequently, it is an economical and practical solution to contemporary agricultural monitoring and management.

Such networks differ from traditional WSNs due to their mobility feature that improves network coverage, connectivity, scalability, and energy efficiency while prolonging the network's lifetime (Cao, Y. Cai, and Yue, 2019). MWSNs can be used for various applications with enhanced connectivity and coverage and with limited computational complexity (Fei et al., 2017; Donta, Amgoth, and Annavarapu, 2022; Chang, W. Chen, et al., 2023). The commonly used applications are environmental monitoring, military surveillance, and healthcare. However, MWSNs face several challenges, including limited energy, memory, processing capabilities, and communication and coordination issues. To overcome these challenges, Machine Learning (ML), Deep Learning (DL), and Deep Reinforcement Learning (DRL) methods can be applied to MWSNs to improve their performance and capabilities.

ML and DRL are subsets of Artificial Intelligence (AI) that have gained much attention recently. ML refers to algorithms and techniques that allow computers to learn from data and make predictions or decisions based on that data. There are different types of ML algorithms, such as supervised learning, unsupervised learning, and DRL. DRL is a subset of ML that models and resolves complicated issues using Artificial Neural Networks (ANNs). ANN mimics the human brain and consists of layers of interconnected nodes that process and transform data.

DRL algorithms can automatically learn to extract features and patterns from large amounts of data, making them well-suited for tasks such as image and speech recognition, natural language processing, and recommendation systems. Generally, ML and DRL are powerful tools that can help to solve various problems across various domains, from healthcare and finance to transportation and entertainment.

For MWSN, the DRL algorithms can train the network from the existing datasets or historical records. The training can allocate a medium/channel or route the data. There are several ways in which ML and DRL can be applied to improve the performance and capabilities of MWSNs. Moreover, the training and execution of these models often demand significant quantities of data and computing power, rendering them impractical or unfeasible for certain applications.

WSNs are crucial in connecting dispersed, possibly autonomous sensors to monitor and manage systems or the environment. The integration of modern technologies, such as information and communication in WSNs, has enabled sensing and computation capabilities, making them essential to developing the next-generation Internet. These technological advancements pave the way for solving a wide range of problems and enhancing our daily lives. ML and DRL have shown great promise in addressing various challenges MWSNs face. There are several ways in which ML and DRL can be applied to improve the performance and capabilities of MWSNs as discussed below:

- The ML and DRL can be used for energy-efficient routing. By utilizing ML techniques, it is possible to predict the energy consumption of different routes in the network and determine the most energy-efficient route for data transmission. This approach can significantly reduce energy consumption and prolong the network's lifespan.
- With the help of DRL algorithms, detecting and tracking objects in the network, such as vehicles or criminals, is possible. This can be particularly useful for surveillance and security purposes, where it is necessary to monitor the movement of objects within the network.
- ML algorithms can detect anomalies in sensor data, such as sudden changes in temperature or humidity. This could indicate a potential problem, allowing for timely intervention before any significant issue occurs.
- ML algorithms can perform early prediction of sensor node failure or maintenance requirement, allowing for proactive maintenance to be carried out. This reduces downtime and increases the overall efficiency of the network.
- ML algorithms can optimize resource allocation in the network. For example, tasks can be assigned to nodes based on their available processing power and memory. This ensures that the available resources are used effectively for maximizing the network's overall performance.

In summary, ML and DRL techniques can improve the performance and efficiency of MWSNs by enabling energy-efficient routing, object detection and tracking, anomaly detection, predictive maintenance, and resource allocation. These techniques can help MWSNs meet the needs of various applications, making them more valuable and reliable. Future research can focus on further improving the accuracy and effectiveness of these techniques for MWSNs. However, in the present work, the main focus is to increase the lifetime of the network by introducing ML-based energy-efficient methods with maintaining coverage and connectivity without compromising with computational complexity (Chang, W. Chen, et al., 2023).

MWSN design and construction, topology selection, and node power allocation all depend extensively on coverage and network connectivity. A sensor node's coverage area is directly impacted by its transmission power. Reduced network connectivity and node coverage might result from reduced transmit power (Bhasgi and Terdal, 2021). However, to increase the battery life of nodes, it is often necessary to reduce transmit power. In the case of MWSN, probabilistic modeling is necessary to predict and optimize energy consumption as well as coverage because the location and separation between nodes continually change. In such cases, it is quite difficult to maintain connectivity and save energy together. Effective data transfer is the only effective way to maintain coverage and connectivity with optimal energy saving.

For effective transfer, the routing method plays a critical role in MWSNs that requires careful management to ensure reliable data transmission between sensor nodes and the base station. Mainly, mobile nodes are used to increase the coverage range and connectivity. The mobile agent is responsible for collecting data from various nodes and transmitting it to the base station. However, the routing process in (MWSNs) faces several challenges.Firstly, implementing a global addressing process is impractical due to the deployment of numerous sensor nodes. In MWSNs, conventional IP-based protocols, designed for large-scale network infrastructure, are often unsuitable.Secondly, while most Wireless Sensor Networks (WSNs) require a continuous stream of sensed data from multiple sources to a specific sink node or base station, this requirement conflicts with typical communication networks.Thirdly, within the vicinity of a phenomenon, multiple sensors may generate similar data, resulting in heavy redundancy traffic across the network (Sara and Sridharan, 2014). This redundancy consumes more energy and bandwidth, leading to various issues such as delays, packet loss, and bandwidth degradation.

1.2 Motivation

Routing techniques play a crucial role in Mobile Wireless Sensor Networks (MWSNs), especially when dealing with large-scale deployments encompassing a multitude of sensor nodes. However, assigning global IDs to these nodes becomes a challenging task. As a result, traditional protocols may not be well-suited for MWSNs. These networks possess unique inherent characteristics that add complexity to developing routing protocols. Such characteristics include a highly dynamic network environment, specific requirements driven by the application, and constraints on energy, storage, and processing capabilities.

Because of these specific characteristics of MWSNs, building an effective routing protocol is a complex issue. The routing protocol's architecture can be influenced by various MWSN characteristics. Some of the concerns and obstacles associated with routing in MWSNs are discussed below:

- **High Network Dynamics:** MWSNs experience frequent changes in their network topology due to factors such as node mobility, link failures, and node failures. Adapting to this dynamic nature necessitates the development of routing algorithms that can efficiently handle topology changes.

- **Energy consumption**: The mobile wireless sensor node can only be equipped with a limited power source. Sensor node lifetime is therefore highly dependent on battery lifetime. In a multi-hop ad-hoc sensor network, each node plays the dual role of data originator and data router. Basically, each node has three main tasks; sensing, communicating/relaying, and processing. The failure of a few nodes can cause significant topological changes and might require re-routing of packets and re-organization of the network. Hence, power conservation and power management take on additional importance.The total energy consumption of each node directly influences the overall lifetime of the network. When a sensor node's energy level drops below a certain threshold, it becomes nonfunctional, adversely impacting the network's performance.Consequently, optimizing energy management becomes a crucial task for routing protocol designers, aiming to maximize the network's lifespan.

- **Application-Specific Requirements**: Depending on considerations such as data delivery delay, dependability, and bandwidth limits, different applications have different routing needs. Routing protocols should be created to fulfill these special needs while taking into account the limits of MWSNs.

- **Fault Tolerance**: Sensor nodes in MWSNs are prone to failure, and network connectivity can fluctuate unexpectedly. As a result, routing protocols must be resilient and adaptable to deal with node failures and changes in network connectivity.

- **Scalability**: The routing technique must be scalable as the number of sensor nodes increases with the node density. The deployment of a substantial number of sensors is common in the target area, and as the network operates, its size may expand. It is critical to design the protocol so that node scalability does not have a detrimental influence on performance.

To address these challenges, intelligent routing algorithms that can dynamically adjust the data routing paths based on network congestion and bandwidth availability are necessary. Such algorithms can help optimize network performance and reduce energy consumption by minimizing data transmission over congested routes.

1.3 Research Questions

The problem statement of the thesis is based on the following research questions.

- How can high throughput be achieved in Mobile Wireless Sensor Networks (MWSNs) across different topologies and levels of network connectivity while ensuring energy efficiency?

- What techniques can be employed to estimate and optimize the energy consumption of sensor nodes in MWSNs?

- Which routing protocols can adapt to node mobility in MWSNs while optimizing energy consumption, and what characteristics make them effective?
- What performance metrics are crucial for assessing the impact of mobility on topology and network connectivity, considering energy efficiency in MWSNs?
- How does the performance of MWSNs in terms of energy consumption and throughput differ between self-organizing maps and other optimization methods?
- What are the key considerations in developing a simulation platform for comparative analysis of different optimization methods regarding energy consumption, both with and without optimization techniques?
- How can data aggregation methods contribute to energy-efficient data transmission in various optimization scenarios within MWSNs?

1.4 Contributions

The main contributions of this thesis are as follows.

- First,a system and energy model are presented to explain the topological configuration of MWSNs and to analyze the energy required for transmitting and receiving data across the network.
- To optimize the performance of the sensor network, routing-centric parameters are derived, focusing on expected energy consumption, expected node degree, and expected forward progress toward the sink.
- Two routing algorithms, namely Self-Organizing Maps-based Optimized Link State Routing (SOM-OLSR) and Deep Reinforcement Learning-based Optimized Link State Routing (DRL-OLSR), have been proposed for energy-efficient data transmission. Both algorithms leverage deep learning techniques to improve routing performance in Mobile Wireless Sensor Networks (MWSNs). These two methods are compared considering their practical significance in terms of computational complexity and deployment scenarios.
- The proposed methods dynamically adjust the data routing paths based on the network congestion status and bandwidth availability. The most suitable route is selected for data transmission while considering factors such as energy consumption, network congestion, and link quality.
- The main objective of both algorithms is to optimize the ideal balance between various parameters such as connection probability (CP), end-to-end (E2E) delay, overhead, throughput, and energy consumption.

- Both algorithms select the most suitable route for data transmission while considering factors such as energy consumption, network congestion, and link quality.
- Both algorithms aim to optimize the delicate balance between parameters such as connection probability, end-to-end delay, overhead, throughput, and energy consumption.
- The SOM-OLSR is an unsupervised artificial neural network-based energy-efficient routing protocol designed to discover the optimal path from the sensor node to the sink node. SOM-OLSR ensures reliable communication by effectively handling noisy and incomplete data, making it suitable for real-time applications.
- The DRL-OSLR algorithm is a fault-tolerant routing technique designed to maintain robust connectivity in the dynamic network topologies. By utilizing multiple paths between nodes, the algorithm ensures that data can still be successfully delivered even if one of the paths is disrupted.
- An aggregation method has been developed to achieve energy-efficient data transmission in both DRL and SOM scenarios without compromising throughput.
- Both algorithms are evaluated through simulations in Matlab to analyze their performance in terms of various performance metrics as mentioned above.
- The performance of the proposed methods is also compared with the traditional routing method indicating their significant performance improvement.

1.5 Structure of Thesis

The remaining chapters of this thesis are organized as follows.

- **Chapter 2** This chapter provides an overview of several topics related to WSNs and Mobile WSNs (MWSNs) and the use of ML in this context. The chapter discuss the characteristics of WSNs and is followed by a discussion of MWSNs, which covers topology and routing issues that arise due to mobility. The chapter then briefly introduces ML, including its classification and popular algorithms. Afterward, various applications are reviewed for ML in WSNs, including localization, connectivity and coverage, routing, data aggregation, and mobile sinks. Finally, the chapter discusses the benefits and drawbacks of implementing ML in WSNs.

- **Chapter 3** This Chapter talks about Deep Reinforcement Learning (DRL) as this lays the necessary foundation for our work. We begin with the introduction, followed by the discussion on routing protocols for MWSNs, which presents an overview of different types of routing protocols. In particular, Optimized link state routing (OSLR) protocol is discussed. Subsequently, routing protocols based on DRL with reference to WSNs are elaborated. I have reviewed and compare several DRL-based routing protocols for MWSNs that exist in the literature. After that, some well-known techniques that are the basis of DRL are discussed, such as ANNs, CNNs, and RNNs, along with their architecture. The pros and cons of using DRL in MWSN is also discussed. Finally, details of the performance metrics is discussed, which are used in evaluating various routing protocols in the context of MWSNs with selected routing protocol.

- **Chapter 4** In Chapter 4, the SOM model is developed for efficient routing of MWSN. The background of SOM is discussed with a mathematical model. The SOM-OLSR protocol is proposed, and the network is analyzed for different connection probabilities. The energy consumption, delay, and throughput are analyzed for different scenarios.

- **Chapter 5** This chapter delves into Deep Reinforcement Learning (DRL), showcasing its significance across diverse domains by addressing complex challenges through sequential decision-making. It emphasizes DRL's ability to learn from environment interactions and optimize decisions based on rewards, leading to robotics, gaming, healthcare, and more breakthroughs. The core components of DRL, including agents, environments, action and state spaces, reward systems, and learning algorithms, are elucidated, along with the integration of deep neural networks for handling high-dimensional data. The chapter further presents the outcomes of implementing DRL in mobile sensor networks. It compares its performance with traditional methods and highlights strengths, limitations, and future research prospects, providing a holistic understanding of DRL's potential in transforming decision-making processes.

- **Chapter 6** This chapter focuses on using aggregation methods in Deep Reinforcement Learning (DRL)-based wireless sensor networks (WSNs) to reduce overall energy consumption. The significance of aggregation algorithms in minimizing energy usage is underscored as these methods consolidate data samples, eliminate redundancy, and reduce the number of transmitted packets, ultimately leading to decreased energy consumption. The chapter proceeds to explore routing protocols integrated with aggregation for Multi-hop Wireless Sensor Networks (MWSNs), specifically investigating Optimized Link State Routing (OSLR), Self-Organizing Maps (SOM), and DRL-based protocols. By comparing these protocols with and without aggregation, the study sheds light on their efficiency in achieving energy-efficient data transmission. The related works section further analyzes existing research contributions in WSNs, particularly in data transmission and aggregation challenges, using machine learning and DRL techniques. This analysis reveals the potential of intelligent methods to enhance data aggregation, optimize energy usage, and address real-world complexities in wireless sensor networks.

- **Chapter 7** This chapter comprehensively evaluates and compares energy consumption optimization techniques in mobile sensor networks, specifically focusing on Self-Organizing Maps (SOM) and Deep Reinforcement Learning (DRL). The evaluation reveals the superiority of DRL over SOM and a non-optimized approach, demonstrating DRL's effectiveness in dynamically adapting decision-making policies to optimize energy usage. The findings emphasize the importance of incorporating optimization techniques for energy efficiency while highlighting SOM's limitations in this context. The chapter concludes by suggesting future research directions, such as advanced DRL architectures and hybrid approaches, to enhance energy consumption optimization further and advance the practical implementation of mobile sensor networks.

Chapter 2

Background Study

This chapter provides an overview of several topics related to WSNs and Mobile WSNs (MWSNs), as well as the use of ML in this context. The chapter begins by introducing the characteristics of WSNs and is followed by a discussion of MWSNs, which covers topology and routing issues that arise due to mobility. The chapter then briefly introduces ML, including its classification and popular algorithms. Afterward, the chapter reviews various applications of ML in WSNs, including localization, connectivity and coverage, routing, data aggregation, and mobile sinks. Finally, the chapter discusses the benefits and drawbacks of implementing ML in WSNs.

2.1 Wireless Sensor Networks

Wireless sensor networks (WSNs) are used for various real-time applications and they are one of the most promising and emerging technologies that have applications in diverse domains such as military, automation, vehicle tracking, environmental, wildlife tracking, agriculture, etc. (Yick, Mukherjee, and Ghosal, 2008).

WSNs offer many advantages such as cost-effectiveness, tiny size, ease of deployment, self-organization, and low maintenance cost (Rawat et al., 2014). They comprises tiny sensor nodes, randomly deployed in a targeted area to sense and collect data. Each sensor node is equipped with a sensing unit, a transceiver, a processing unit, a battery, and a memory unit.

The sensor unit is responsible for observing a physical phenomenon such as temperature, pressure, humidity, etc. A transceiver is used to connect to the network and exchange data. A small processing unit is employed to process the data, a battery unit powers up the nodes, and a buffer stores the data temporally (Akyildiz et al., 2002). In a typical WSN, sensor nodes not only communicate data among themselves but also communicate with a gateway, also known as sink or base station (BS), for further data processing via single or multi-hop network transmission (F. Wang and J. Liu, 2011; S. Yang et al., 2016; Yetgin et al., 2017; N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, 2013). The BS usually connects WSNs to external networks such as the Internet of Things (IoT) or cloud networks or the global Internet.

IoT, a successor of WSNs, refers to internet-connected devices, while IoT-enabled WSNs is a subset of an IoT-based system. In an IoT-enabled WSN, sensor nodes collect real-time data to monitor the environment, and a base station serves as an access point to connect to an end-user via the internet.

Although WSNs offer promising benefits, they face several challenges, such as unstable and dynamic connectivity links, limited processing power, limited coverage, low data rates, and limited battery power. One of the significant concerns in WSNs is energy because tiny sensor nodes are battery-powered, and the transceiver is the main source that drains the energy of the nodes extensively. Quick energy depletion of the nodes causes the nodes to die and thereby reducing the network lifetime. When old nodes die, new nodes must be added for the network to function correctly.

The addition of the new nodes also causes more energy consumption of the existing nodes because they exchange control packets to sustain network topology changes and compute new routes for data forwarding. This re-computation of routing paths incurs further energy consumption. Therefore, saving the energy of the nodes is of utmost importance because it impacts the overall network lifetime. Saving on extensive unnecessary communication attempts and recomputing dynamic routing paths can save the energy of the nodes and extend the network lifetime.

WSNs can include Mobile Wireless Sensor Networks (MWSNs), which have broad applications in various fields. In a mobile wireless sensor network, the sensor nodes can move within the network. The rapid growth of mobile technology and the Internet has made MWSNs a popular research area in WSNs. MWSNs differ from traditional WSNs due to their mobility feature that enhances network coverage, connectivity, scalability, and energy efficiency while prolonging the network's lifetime (Cao, Y. Cai, and Yue, 2019).

According to node movement, MWSNs can be categorized into three types: 1) the sink node moves while the ordinary nodes are stationary; 2) the sink node is stationary while few of the ordinary nodes move; and 3) both the sink node and ordinary nodes move (Cao, Y. Cai, and Yue, 2019). The focus of our work is based on two where the sink node is stationary while few of the ordinary nodes move. Usually routing process in a mobile network is typically intricate and even more challenging in MWSNs since sensor nodes are low-power, cost-effective, and resource-constrained mobile devices. Although many effective routing protocols for MWSNs have been proposed through recent research, there are still unresolved issues such as sustaining network connectivity, minimizing energy consumption, and maintaining sufficient sensing coverage (Sara and Sridharan, 2014). More on MWSNs in the next Section as follows.

2.2 Benefts and Possible Applications of WSNs

The key benefit of WSNs is that they can be implemented almost anywhere without the need for any specific communication infrastructure. The sensor nodes are networked in a self-organizing manner in many applications that require unattended operations. It allows a WSN to be deployed as an alternative to non-existent infrastructure (for cost effectiveness) or if the existing infrastructure is not appropriate to use. The following are seen as possible applications for WSNs.

- **Military applications:** WSNs can be used to monitor battlefield conditions, including intruder movement, drastic changes in temperature, humidity, radiation levels, and other environmental factors to ensure the safety and effectiveness of military operations. WSNs can also track and monitor the movement of military assets, such as vehicles, aircraft, and soldiers, providing situational awareness to commanders.

- **Environmental applications:** WSNs can measure pollutants, soil nitrogen level, potassium level, phosphorous level, temperature, pH levels, and other indicators of environmental health to ensure better air, soil, and water quality. WSNs can provide early alerts for earthquakes, tsunamis, hurricanes, and floods by detecting changes in environmental parameters and sending alerts to relevant authorities.

- **Health applications:** WSNs can collect real-time patient health data and transmit it to healthcare providers, enabling remote monitoring of vital signs and medical conditions. In addition, WSNs can be integrated into homes and healthcare services to monitor the elderly and individuals with medical conditions, aiding and alerts in emergencies. WSNs can track the availability of medical supplies, monitor equipment status, and improve hospital processes.

- **Home applications:**WSNs enable the automation and control of home devices, such as lighting, thermostats, security cameras, and appliances, for improved energy efficiency and accessibility. In real-time, WSNs can detect unauthorized access and alert homeowners about potential security breaches. WSNs can monitor indoor air quality, humidity levels, and other environmental issues to create a healthier and more comfortable living environment.

- **Commercial and industrial applications:** WSNs can track goods and assets throughout the supply chain, providing real-time visibility and enhancing logistics operations. WSNs can control and optimize building systems, such as lighting, HVAC, and occupancy, to improve energy efficiency and reduce operative costs. WSNs can monitor equipment conditions, production processes, and safety parameters in industrial settings, ensuring smooth operations and preventing failures.

2.3 Types of Wireless Sensor Networks

The WSN can be divided into various categories, mainly based on applications.

- **Underwater and Underground Wireless Sensor Network:** This network is designed to operate underwater, these networks are used for oceanographic research, environmental monitoring, and underwater exploration. In addition, the land WSN can be deployed in soil or underground for applications like agriculture, environmental monitoring, and geological studies.

- **Application based Mobile Wireless Sensor Network (MWSN):** In MWSN, sensor nodes are attached to mobile objects to measure and report environmental events. The network is deployed similarly to WSN in an ad-hoc fashion and without any centralized control. Problematic issues and research questions in MWSN are similar to WSN because of their similarity in terms of network architecture, deployment, and wireless communication. Although resource constraint is still a problematic issue in MWSN (similar to WSN), it is not critical as much as WSN; because, firstly, more powerful resources can be attached to the mobile objects (e.g. vehicles power resources), and secondly, MWSN nodes have the ability to move to re-charge area when their energy level is low. However, research in MWSN needs to give extra attention to sensor nodes' mobility patterns and the influences on the data collection and wireless communications. The vehicular sensor network is an example of MWSN in which the GPS-sensor nodes collect and report the location as well as traffic data to the sink node using GSM communications.

- **Aerial WSNs (or UAV-based WSNs):** Aerial WSNs utilize unmanned aerial vehicles (drones) to organize and manage sensor nodes for applications like aerial surveillance, disaster management, and precision agriculture. The integration of UAVs with terrestrial networks is also used for various applications. Most such WSNs are deployed on land or in terrestrial environments.

2.4 Mobile Wireless Sensor Networks

WSNs are deployed in different environments, including land, underground, and underwater, which pose distinct challenges and constraints. WSNs are classified based on deployment as terrestrial, underground, multimedia, mobile, and multi-media. Based on sensor node resources, an MWSN can be classified as homogeneous or heterogeneous. A homogeneous MWSN consists of identical mobile sensor nodes. At the same time, a heterogeneous MWSN comprises mobile sensor nodes with varying abilities in properties such as battery power, memory size, computing power, sensing range, transmission range, and mobility. Deploying nodes in a heterogeneous MWSN are more complicated than in a homogeneous MWSN (Ramasamy, 2017).

MWSNs are influenced by the shared medium and varying topology, where channel access needs to be regulated. As a result, the network topology is crucial in routing protocol design and determining the transmission path of data packets to their intended destination (Silva et al., 2011; Mamun, 2012). However, for large-scale MWSNs with mobile sensor nodes, traditional topologies such as flat/unstructured, chain, tree, and cluster may not perform well. A hybrid topology is the preferred option to address these issues, as it enhances data collection and improves network performance. In addition, the routing protocol is critical for selecting an efficient and reliable data transmission path (Velmani, n.d.; Rezazadeh, Moradi, and Ismail, 2012). In addition, the distinctive feature of MWSNs presents additional obstacles in creating an effective routing protocol that considers the network's dynamic topology, node mobility, and various constraints such as energy, computational complexity, resource availability, storage, and bandwidth.

2.4.1 Mobility and Topology in MWSN

In WSNs, mobility refers to the capacity of nodes to change their location from the one they were initially deployed in. Mobile wireless sensor networks offer various benefits over traditional static wireless sensor networks (Wichmann and Korkmaz, 2015): First, the mobility of MWSNs expands the coverage of wireless sensor networks, which reduces the number and difficulty of sensor node deployment required for area coverage. Unlike traditional static wireless sensor networks that depend on the layout of many sensor nodes, the mobility characteristics of MWSNs make coverage more efficient.

Second, MWSNs can improve data transmission speed, throughput, and network latency by utilizing mobility to adopt a delay-tolerant routing strategy, which reduces the delay caused by multi-hop transmission. The introduction of mobile nodes results in a linear increase in throughput compared to static wireless sensor networks. Additionally, reducing multi-hop transmission can enhance communication quality by reducing errors and packet loss during data transmission. Static sensors use the multi-hop method, which is only appropriate for small-scale wireless networks. Once the network size increases, the multi-hop transmission will cause significant delay and increase data transmission unpredictability (H. Zhao et al., 2015).

Third, integrating mobile nodes in MWSNs can reduce energy consumption and increase network lifetime. In static wireless sensor networks, energy consumption is unbalanced, and nodes that forward data over multi-hops consume significant energy, which impacts network stability and longevity. Due to node mobility, MWSNs can supply charging power or larger batteries, eliminating the need to optimize energy consumption during network design. With mobile nodes, static nodes do not have to wait for data forwarding during wake-up, and they can transmit data to mobile nodes nearby, resulting in reduced energy consumption and increased network lifetime (Ahmad et al., 2015). Furthermore, after mobile nodes are introduced, sensor nodes can communicate with mobile nodes directly without transferring data through multiple hops to the fixed data center. This reduces the burden of the one-hop node around the data center and balances energy consumption across the network (Shi et al., 2015).

However, designing a mobile wireless sensor network is difficult due to frequent path breakages caused by channel fading, shadowing, interference, node mobility, and node failure. Pre-constructed message delivery networks cannot cope with the changing topology. Frequent location updates from mobile nodes can drain sensor node batteries and increase collisions (Sara and Sridharan, 2014). Therefore, factors like node mobility, limited resources, and bandwidth restrictions must be considered when designing MWSN.

For mobile wireless sensor networks on a large scale to work well, factors such as communication reliability, network connectivity, data collection, sensor mobility, and management of network topology must be taken into account. As a result, designing efficient routing protocols for WSNs requires accurate modeling of both sensor mobility and topology management. Topology helps establish a dependable network and ensures a better quality of service for traffic and end-to-end connectivity, while mobility describes how sensor nodes behave in terms of their movement pattern (Sara and Sridharan, 2014).

2.4.2 Routing in MWSN

In WSNs, the data routing process holds significant importance as it is responsible for establishing the appropriate routing paths between sensor nodes and forwarding data packets to the sink. Hence, the primary aim of routing protocols is to establish efficient paths between sensor nodes within the network. When dealing with MWSNs, routing protocols must consider factors such as mobility, data redundancy, energy efficiency, and the dynamic nature of network topology. As a result, routing in MWSNs requires careful management and implementation due to the numerous challenges and limitations associated with sensor networks, as mentioned in references (Mehta and Pal, 2017). Routing protocols are typically categorized based on various characteristics, such as network structure and application, which suggests that no one routing mechanism can effectively and practically serve all types of WSN applications (Ketshabetswe et al., 2019; Biradar et al., 2009). As a result, routing protocols can be divided into several categories, and those have been discussed thoroughly in 3.1.1.

In the present work, OLSR is hybridized with SOM and DL methods. In scenarios where partial sensor nodes work as mobile agent nodes is deployed for monitoring applications, the OLSR protocol is the best as it offers many key benefits. Therewith, OLSR is well-adapted to dynamic and resource-limited contexts, making it suitable for monitoring application scenarios where the network topology could undergo changes due to the mobility of agent nodes and environmental conditions. OLSR uses proactive routing, where nodes periodically share link-state information in order to maintain current routing tables. Such a proactive approach guarantees that routing decisions can be made with less delay and with the optimal usage of network resources, even in networks with mobile nodes and varying connectivity.

The OLSR is created to ensure the most effective routing protocol through the reduction of redundancy and control of message transmission. In MWSNs, especially those that deploy mobile nodes as partial networks, the energy and bandwidth saving work as the pivotal factors. OLSR optimizes it in such a way that it is only exchanging routing information with a not so large subset of nodes simply known as Multi-Point Relays (MPRs) rather than broadcasting it to all nodes in the network. This minimizes the cost per packet transmitted and the energy consumption within the network.

On the other hand, the design of OLSR in such a way, that it can handle multi-routing paths and its resilience to link failures make OLSR a good fit for scenarios where the network connectivity can be intermittent or characterized by disruptions, for instance in agricultural areas with variable topography or signal interference. OLSR maintains multiple routes that allow it to adapt to the network topology changes, ensuring the transmission of data even in challenging conditions.

The proactive nature, routing efficiency, and robustness of the OLSR protocol recommend it as a suitable candidate for Mobile Wireless Sensor Networks deployed in agricultural monitoring with the mobile agent node distribution in a partial way. Its ability to cope with diversified surroundings, save energy, and ensure dependable communication in the face of different conditions meet the needs and the challenges of such environments fittingly, making this a protocol well suited for assuring smooth and competent data transfer in agricultural MWSNs.

2.5 Artificial Intelligence

In recent times, the increasing use of Artificial Intelligence (AI) has resulted in a shift in focus toward the need for devices and sensor nodes to learn from their experiences and events rather than being explicitly programmed to act in certain ways (Mitchell, 1997; Ayodele, 2010; Langley and Simon, 1995). This is what is termed ML, and DRL is a subclass of machine learning that involves training an agent to learn how to make decisions by interacting with an environment and receiving feedback in the form of rewards or penalties. The goal of DRL is to enable the agent to learn the optimal actions to take in a given situation to maximize the cumulative reward over time.

DRL has been proposed as a potential solution to routing problems in mobile wireless sensor networks (MWSNs). In an MWSN, sensors are deployed in a mobile environment and are capable of sensing and communicating with each other. However, due to the mobility of the sensors, traditional routing algorithms may not be effective, and a more adaptive approach is required. DRL can be used to train sensors to learn how to make routing decisions in an MWSN. The sensors can be viewed as agents that interact with their environment by selecting actions, such as selecting a path for data transmission. The environment includes other sensors in the network and the state of the network, such as the topology and traffic load. The sensors receive feedback in the form of rewards, which indicate the quality of their actions in terms of performance metrics such as end-to-end delay, energy consumption, or throughput. The reward function can be designed to reflect the specific requirements of the network, such as minimizing delay or maximizing energy efficiency.

The thesis focuses primarily on routing in MWSNs, but it is essential to consider other factors, including energy efficiency, network security, and random deployment, in addition to relevance clustering and combining AI and optimization techniques. To address the challenges faced by MWSNs, a location-aware and energy-efficient routing mechanism based on an optimized AI technique was selected. The proposed solution aims to enhance QoS by improving parameters such as throughput, end-to-end delay, packet delivery ratio, packet loss rate, collision avoidance intensity, and energy consumption.

2.6 Machine Learning

Machine Learning (ML) is a subfield of AI, that allows computers to autonomously learn patterns from data and make predictions or decisions without being explicitly programmed (Mitchell, 1997; Ayodele, 2010; Langley and Simon, 1995). Instead of following a rigid set of instructions, algorithms based on ML involve processes of feeding computers with large amounts of data, which allows them to analyze data, identify patterns, and make predictions on that data. ML makes computing processes more efficient, reliable, and cost-effective, because it continually improves the performance of tasks over time without the need for manual intervention.

ML is broadly classified into supervised learning, unsupervised learning, and reinforcement learning. Certain hybrid approaches also exist that combine supervised, unsupervised, and reinforcement learning elements, such as semi-supervised learning. Machine learning techniques have been utilized extensively for tasks like classification, regression, and density estimation across different domains, including engineering, computing, bio-informatics, computer vision, graphics processing, and natural language processing. Recently, the latest advancements in ML have been utilized to tackle different challenges in WSNs (Mohammad Abu Alsheikh et al., 2014) and IoT (Jagannath et al., 2019; Mahdavinejad et al., 2018). Specifically, ML has been applied to improve energy efficiency, data analysis, fault tolerance, security, and network self-organization in WSNs without re-programing. These advances have enabled the development of more efficient, robust, and intelligent WSNs that can better meet the demands of various applications.

As WSNs are battery-powered, they perform duty cycling to save energy, which means most of the time, they are in a sleep state, and wake only if they have data to transmit. In this way, they extend network life time (Yuqin Wang et al., 2020; F. Liu et al., 2017). In duty-cycled WSNs, data transmission between nodes can occur only when both are in an active state (Q. Chen, Gao, Cheng, et al., 2017; Akbar, Yu, and Cang, 2016). In this scenario, determining the appropriate forwarder node is a complex task. However, maintaining accessibility between the sender and receiver is crucial during the data routing process (Carrano et al., 2014). An incorrect choice of forwarder can result in a misdirection, causing data to take a longer route to reach the base station, resulting in increased latency. Moreover, different applications of WSNs require different performance metrics to be achieved such as low end-to-end delay, high energy efficiency, reliable link quality, and protection against nearby interference. There is a trade-off involve among these metrics, for example, saving energy causes nodes to sleep more often which in turn increased delay or latency and this can be problematic for delay-sensitive applications (Q. Chen, Gao, Z. Cai, et al., 2018; Sodhro et al., 2018). Thus, a balanced trade-off must exist among these metrics so as to satisfy certain application requirements, however, achieving such balance is a challenge in duty-cycled WSNs.

2.6.1 Machine Learning Techniques

There is an abundant body of literature addressing the various challenges of WSNs through the use of machine learning (Mohammad Abu Alsheikh et al., 2014). In this section, ML and its classifications are discussed, later in subsequent sections, we extended survey on various ML-based algorithms for WSNs, evaluating their benefits, drawbacks, and impact on network parameters such as lifetime, energy consumption, packet loss, etc. Classification of ML techniques has been done based on the learning styles into four categories: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Figure 2.1 depicts the taxonomy of ML techniques.

2.6.1.1 Supervised Learning Supervised learning is one of the important data processing and training approaches in ML. In supervised learning, algorithms are trained with the help of labeled datasets to build models that can classify data or predict outcomes. The model learns the relationship between the input features and the output labels based on the labeled training data, this relationship is then used to make predictions on new, unseen data. At the end of the training process, I can find a function from an input x with the best estimation of output y ($f : x \rightarrow y$). Supervised learning algorithms are crucial in creating a model that captures the relationships and dependencies between the input features and the target outputs. Using this model, predictions can be made for new data by utilizing the relationships learned from prior datasets. The ultimate goal of supervised learning is to generate accurate predictions for unseen data.

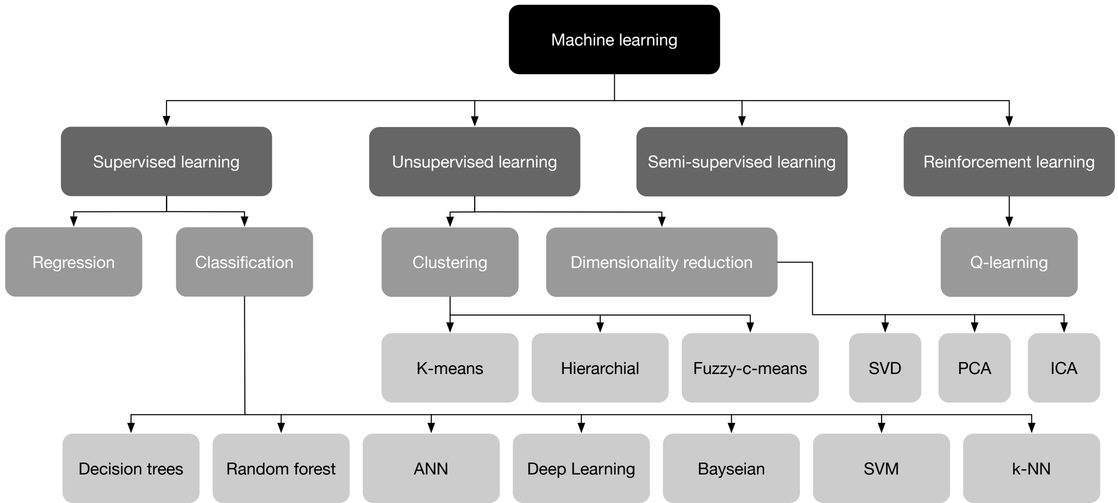


Figure 2.1: Taxonomy of ML techniques

Supervised learning can tackle a range of challenges faced in WSNs, such as improving localization (Banihashemian, Adibnia, and Sarram, 2018), addressing coverage issues (W. Sun et al., 2018), optimizing routing (Mehmood et al., 2017), facilitating data aggregation (Song et al., 2013), controlling congestion (M. Abu Alsheikh et al., 2016), and maximizing energy harvesting (A. Sharma and Kakkar, 2018; Tan et al., 2017). Supervised learning can be divided into two categories: regression and classification. Classification, in turn, encompasses several types of algorithms, including logic-based algorithms (such as decision trees and random forests), perception-based algorithms (such as Artificial Neural Networks and deep learning), statistical learning algorithms (such as Support Vector Machines, and Bayesian algorithms), and instance-based (k-NN) algorithms.

2.6.1.2 Unsupervised Learning Unsupervised learning involves working with input data with no associated output labels. The model tries to identify relationships within the data, even without any prior knowledge of what the outputs should be. This approach is often used for tasks such as grouping similar patterns into clusters, reducing the number of dimensions in the data, and detecting anomalies. In WSNs, unsupervised learning plays a key role in addressing challenges such as connectivity problems (Qin et al., 2017), routing (El, Youssif, Ghalwash, et al., 2016) and data aggregation. The most commonly used unsupervised learning method is K-means for clustering the SNs used in the LEACH protocol. The centroid update equation of K-Means is given below:

$$\mathbf{c}_i = \frac{1}{N_i} \sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j \tag{2.1}$$

One more dimension reduction technique is principal component analysis (PCA). PCA, a vital tool in unsupervised learning, is chiefly employed for dimensionality reduction and data visualization. It transforms high-dimensional data into a lower-dimensional space while retaining maximal variance. This aids in simplifying complex datasets, eliminating noise, and enhancing algorithm efficiency. PCA can decrease dimensionality by projecting data onto principal components, which are orthogonal linear combinations of original features. The covariance matrix for PCA is given below:

$$\mathbf{S} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{2.2}$$

The corresponding eigen values for PCA are given by the below equation.

$$\mathbf{S}\mathbf{v} = \lambda\mathbf{v} \tag{2.3}$$

2.6.1.3 Semi-supervised Learning

Semi-supervised learning is a technique in machine learning where both labeled and unlabeled data are used for training. Unlike supervised learning, where only labeled data is used, or unsupervised learning, where only unlabeled data is used, semi-supervised learning combines the strengths of both to improve accuracy. This technique is used in real-world applications such as natural language processing, web content classification, speech recognition, spam filtering, video surveillance, and protein sequence classification (X. Zhu and Goldberg, 2009). Two main goals in semi-supervised learning are to predict the labels of unlabeled data in the training set and to predict labels on future test data (X. Zhu and Goldberg, 2009). This technique can be divided into two categories: Transductive learning, which predicts the exact labels of a given unlabeled dataset, and Inductive semi-supervised learning, which learns a mapping function $f : X \mapsto Y$ so that f expected to be a good predictor on future data. Recently, WSNs have been utilizing this learning technique to address localization issues (Yoo, W. Kim, and H. J. Kim, 2015).

2.6.1.4 Reinforcement Learning

The reinforcement learning (RL) algorithm learns through ongoing interactions with the environment and collects data to make decisions. It aims to achieve the best possible outcome by identifying the optimal result from the environment. The process of RL is depicted in Figure 2.2. Q-learning is a model-free technique used in reinforcement learning. In this approach, agents interact with the environment and generate a sequence of state-action-reward observations as *state-action-rewards* (for example $< s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, r_3, ... >$) (Poole and Mackworth, 2010). A matrix of rewards, denoted as $R(\mathcal{S}, \mathbb{A})$, is maintained where $\mathbb{A}$ and $\mathcal{S}$ represent the sets of actions and states, respectively. In Q-learning, the actions of the agent are represented in the form of a matrix $Q(\mathcal{S}, \mathbb{A})$, which is of the same size as R and initialized with zero values. The rows and columns of the Q matrix correspond to the current state of the agent and the possible next state, respectively. The transaction rule for updating each entry in the Q matrix involves adding the

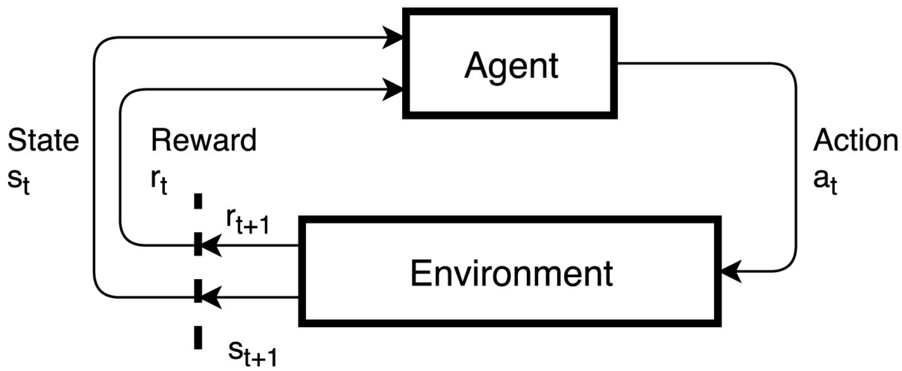


Figure 2.2: Reinforcement learning

corresponding value in the R matrix to the product of the discount factor, denoted as Γ (which has a value between 0 and 1], and the maximum Q value for all possible actions in the next state. The learning rate, denoted as $\wp$ (where $0 \leq \wp < 1$), is also considered. Eq. (2.4) shows how this is done.

$$\begin{aligned} Q_{t+1}(\mathcal{S}_i, \sigma_i) &= (1-\wp)Q_t(\mathcal{S}_i, \sigma_i) + \wp \times R_t(\mathcal{S}_i, \sigma_i) \\ &\quad + \wp \times \Gamma \left(\max_a Q^{'}(\mathcal{S}_i^{'}, \sigma_i^{'}) - Q_t(\mathcal{S}_i, \sigma_i) \right) \end{aligned} \tag{2.4}$$

2.7 Basic System Model

In the proposed Mobile Wireless Sensor Network (MWSN), the establishment of communication and data transmission relies on the Optimized Link State Routing (OLSR) protocol. The goal is to identify neighboring nodes within range and utilize OLSR to determine the most efficient route for data transmission. In the proposed model, it is assumed that all the sensor nodes (end devices) are static and only a few agent sensor nodes are moving for the collection of the data.

To achieve the identification of neighboring nodes, the sensor nodes in the network broadcast periodic 'Hello' messages to discover neighboring nodes within their communication range. By exchanging Link State Packets (LSPs), nodes gather information about their neighbors and the quality of the links between them, including metrics like hop count, signal strength, and available bandwidth.

Based on the collected topology information, nodes construct a network topology map that represents the connectivity between nodes and includes the quality metrics of the links. This map serves as the basis for calculating the shortest path to reach any destination node within the network. Common algorithms, such as Dijkstra's algorithm, can be used to compute these paths efficiently (Fuhao and Jiping, 2009).

Using the shortest path calculations, nodes construct their routing tables, which contain entries specifying the next hop for each destination node. These tables guide the routing of data packets through the network, ensuring efficient and reliable transmission.

The performance metrics help to characterize the network that is substantially affected by the routing algorithm to achieve the required Quality of Service (QoS). The most important QoS parameter is End-to-End Delay(EED). EED is the time taken for an entire message to completely arrive at the destination from the source. Evaluation of end-to-end delay mostly depends on propagation time (PT), transmission time (TT), queuing time (QT) and processing delay (PD). The EED also depends upon Control Overhead. The control overhead is the ratio of the control information sent to the actual data received at each node.

The second important performance parameter of OLSR protocol is routing efficiency (β). The routing efficiency measures the ability of the protocol to establish and maintain communication paths between nodes. It is calculated as the ratio of the number of successfully delivered packets (D_{packet}) to the total number of packets transmitted ($T_{packets}$) in the network. The formula for routing efficiency is expressed as given in (2.5).

$$\beta = \frac{D_{packets}}{T_{packets}} \times 100 \tag{2.5}$$

The routing efficiency depends upon Connection Probability C_p is:

$$C_p(i) = \frac{T(i) \times E(i)}{N \times T_{bps}} \tag{2.6}$$

where $C_p(i)$ is the Connection Probability of i^{th} node, $E(i)$ is E2E delay taken by each node to transmit the packets to the receiver side. $T(i)$ is throughput of i^{th} node. N is the total number of active nodes, and T_{bps} is the total bit per second the node takes to transmit the packets. Throughput is the amount of data successfully transmitted from one point to another in a specified time, measured in units like bits per second (bps).

The energy consumption of the nodes is calculated in terms of the throughput of the network and is given below in (2.7),

$$E_{cons}(i) = Ov_{cons}(i) \times T_p/N \tag{2.7}$$

where $E_{cons}(i)$ is the total energy consumption of i^{th} node, $Ov_{cons}(i)$ is overhead consumption, T_p/N is the total number of packets sent by the total active nodes within the network. Overhead consumption refers to the additional data and resources required for control information, such as headers and error-checking, reducing the effective bandwidth for actual data.

The detailed model of the SOM and DRL are discussed in Chap 4 and Chap 5, respectively.

2.8 Applications of Machine Learning in WSNs

This section presents the use of ML techniques and algorithms in WSNs. We explore how these ML techniques can tackle various challenges in WSN. Specifically, the suitability of various existing ML techniques for different WSNs applications is explored.

2.8.1 ML for WSN Localization

The recognition of the physical or geographical location of a sensor node is referred to as localization in WSNs. In certain applications, sensor nodes are deployed in a field without prior knowledge of their positions, and there may not be enough infrastructure available to locate them after deployment. Nonetheless, it is critical to identify the location of these nodes. The location of a sensor node can be determined through manual assignment, geographical position system (GPS), or using special nodes such as Anchor or Beacon nodes. An example is illustrated in Figure 2.3. Localization is broadly categorized into four types: proximity-based, range-based, angle and distance-based, and known location-based localization (Kuriakose et al., 2014). The position of sensor nodes in the environment can change dynamically due to external factors, and in such situations, network reprogramming or reconfiguration may be necessary. Applying ML techniques in such scenarios can improve the accuracy of node localization (Cottone et al., 2016).

2.8.2 ML for WSNs Coverage & Connectivity

In addition to energy efficiency, ensuring adequate coverage and connectivity are also challenging issues in WSNs. Coverage refers to how effectively each deployed sensor can monitor the area of interest. The deployment of sensor nodes in a network can be either deterministic or random, depending on the application (Mohamed, Hamza, and Saroit, 2017; Fang et al., 2018). Random deployment is often more feasible in most WSN applications compared to deterministic deployment. Coverage can be classified into two categories: full coverage and partial coverage (Mohamed, Hamza, and Saroit, 2017; Elhoseny et al., 2017). Partial coverage can be further classified into focused coverage, sweep coverage, target coverage, and barrier coverage. Connectivity refers to the absence of isolated sensor nodes in the network, meaning that every node in the WSNs is capable of sending its data to the sink node directly or through relay nodes. Figure 2.4 illustrates coverage and connectivity, where nodes A and node B are isolated (disconnected) nodes and the white area represents a coverage hole in the network.

Numerous algorithms have been proposed to address coverage and connectivity problems in WSNs. (Hongbing Li et al., 2017) focuses on addressing static sensors in WSNs, while (Fang et al., 2018; Abo-Zahhad et al., 2016) aim to

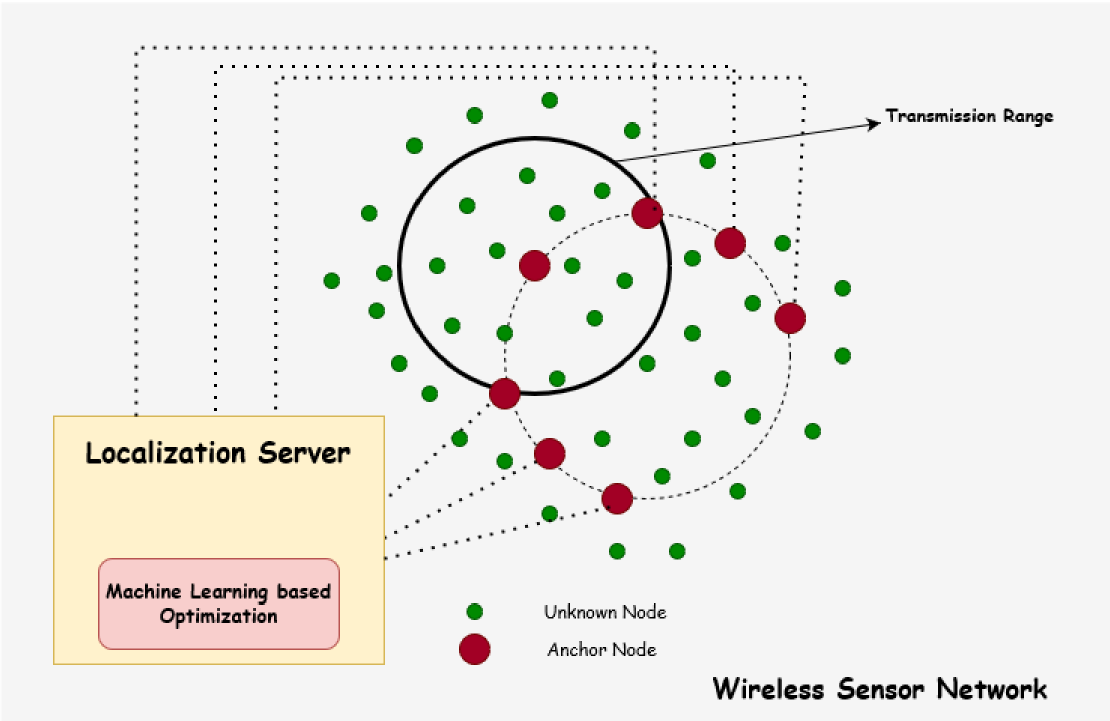


Figure 2.3: ML for WSN Localization

solve similar problems in mobile WSNs. Using ML techniques to tackle coverage and connectivity issues in WSNs offers several advantages, such as quickly and dynamically identifying the minimum number of SNs required to cover the target area, as demonstrated in (Elghazel et al., 2015). Additionally, these techniques can classify nodes as either connected or disconnected in the network and dynamically change routes without losing data.

2.8.3 ML for WSNs Routing

Routing is a significant challenge in WSNs due to limited power, low transmission bandwidth, and processing and memory capacity. In a WSN, sensor nodes are randomly deployed in the environment. Each node collects data from the environment and transmits it to the BS for further processing, as shown in Figure 2.5, where multi-hop transmission from SNs to BS is depicted. Typically, nodes close to the BS consume more energy because they act as relay nodes. The primary objective of the routing protocol design is to reduce the energy consumption of SNs and increase the network lifetime. The potential enhancement of routing in WSNs through the use of Machine Learning (ML) techniques has been increasingly studied in recent years. Researchers have developed several routing methods for WSNs using different approaches (Hammoudeh and Newman, 2015; X. Liu, 2017; Asif et al., 2017). By utilizing machine learning techniques, routing protocols in WSNs can adapt to changes in network conditions, such as node

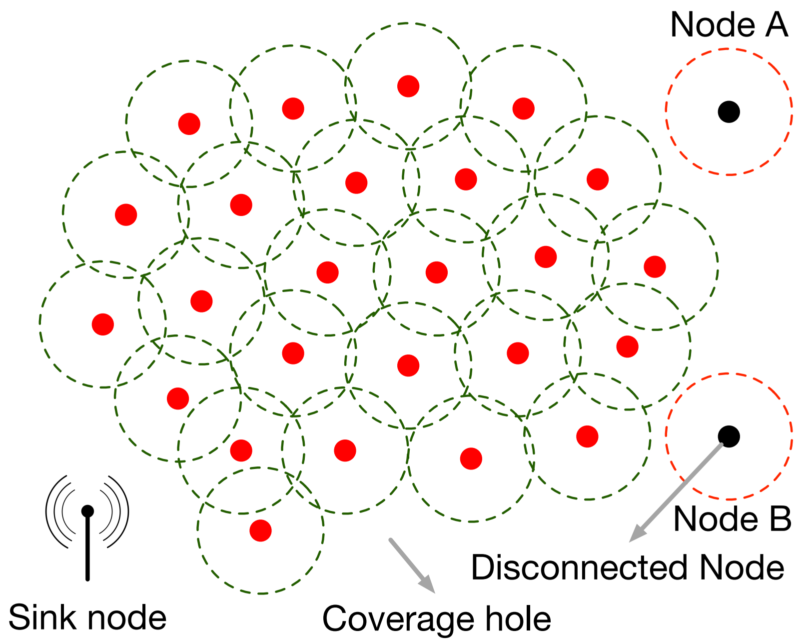


Figure 2.4: Example of coverage and connectivity

mobility or network congestion, and make decisions in real-time to ensure the most efficient routing path for data. Moreover, ML-based routing protocols can predict network changes in real-time, optimize the route for data transmission, and minimize energy consumption, which is crucial for the network's lifetime.

2.8.4 ML for Data Aggregation

Data aggregation is a crucial task in WSNs, as it involves collecting and combining data from multiple sensors to produce a more accurate and complete picture of the environment being monitored. The process of data aggregation in WSNs impacts multiple parameters, including power usage, memory, communication overhead, and computational units. A critical role of data aggregation is to decrease the number of transmissions and communication overhead to achieve an efficient WSN. A practical method for data aggregation balances the energy consumption of sensor nodes and extends the lifetime of the network. Various data aggregation techniques have been developed depending on the network's structure, such as cluster-based, tree-based, in-network, and centralized data aggregation (Ambigavathi and Sridharan, 2018). Previous studies (xie et al., 2017; Lin, Bai, and Yunfei Liu, 2017; Kanjo, Younis, and Sherkat, 2018) have proposed different approaches for data aggregation in WSNs.

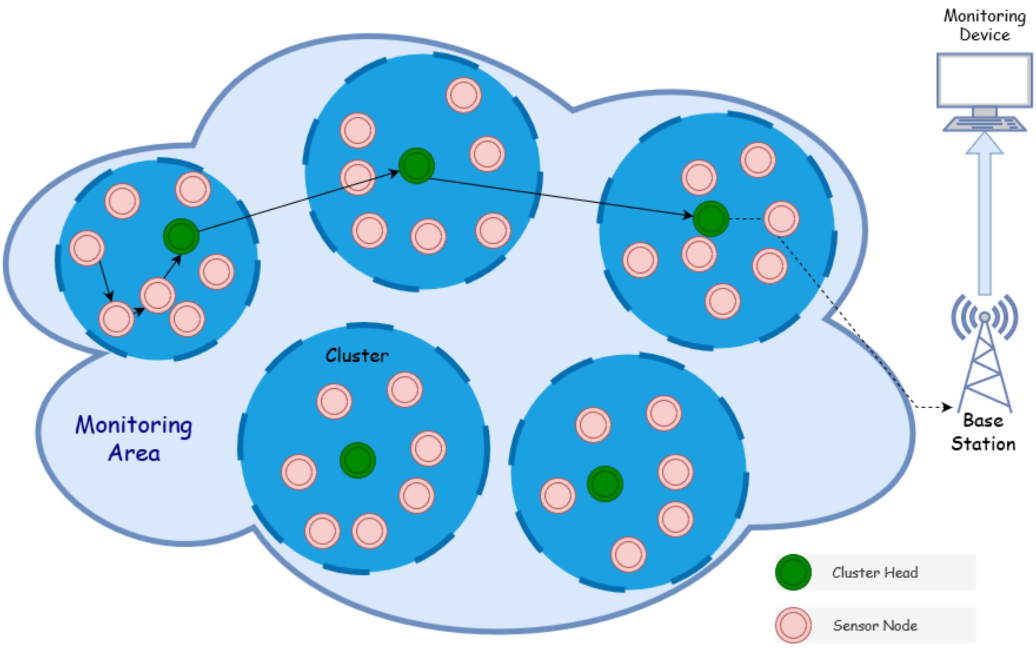


Figure 2.5: Routing in MultiCluster Model for WSN via Cluster Heads to Base Station

2.8.5 ML for Mobile Sink

In WSNs, SNs gather information from the environment and transmit the data to the BS directly or multi-hop manner. When the data transmits in a multi-hop manner, the node which is near the sink will die soon, referred to as an energy-hole problem. To avoid the energy-hole problem, an MS concept has been introduced, a MS visits each sensor node in the network and collects information directly. In large WSNs, visiting every node is difficult, so scheduling MS in an efficient delay-aware manner is a research issue. Therefore instead of visiting each sensor node in the network, MS visit only a few nodes or points in the network called rendezvous points (RPs) to collect data, and all remaining nodes send their data to nearest RPs (Wen et al., 2018; Salarian, Chin, and Naghdy, 2014). Multiple MSs can also be used to avoid the delay of MS to visit SNs, but it is cost-effective.

Authors in (T. Wang et al., 2017) have proposed multiple MS concepts to gather the information from the WSNs and store it in the cloud. In this, each MS is treated as a fog device, and it acts as the bridge between WSNs and the cloud. This algorithm is designed for parallel data-gathering process to minimize the latency and maximizing the scheduling efficiency. This approach balances energy consumption and improves the network lifetime. Recently, ML approaches have been adopted for WSNs to schedule MS and to choose the optimal set of rendezvous points.

2.9 Benefits and Drawbacks of ML in WSNs

Employing ML techniques in diverse WSN applications can enhance efficiency and precision. The significance of ML in WSNs stems from its advantages, including the ability to improve performance and accuracy. However, there are also several drawbacks and limitations that must be taken into account when utilizing machine learning techniques in WSNs, as outlined below (D. P. Kumar, Amgoth, and Annavarapu, 2019).

Benefits

- ML's capacity to update models in real-time provides an excellent solution for modeling dynamic environments. For instance, a landscape monitoring system that relies on WSNs may experience changes in network topology due to sensor failure or relocation. With the use of ML methods, the connection model can be rebuilt dynamically, and an optimal clustering and routing scheme can be selected.

- Developing precise mathematical models to approximate the relevant environmental factors can be challenging in many applications. ML methods offer practical solutions that enable the establishment of low-complexity models that provide good approximations.

- ML methods have a strong ability to exploit temporal and spatial correlations, which makes them an excellent choice for event detection, fault node tolerance, and prediction-based data fusion.

- ML methods can improve their intelligence by continually learning from larger datasets, enhancing decision-making's reliability and accuracy. For instance, intrusion detection systems based on WSNs can enhance their detection accuracy by learning from past experiences over time.

- WSNs are valuable in exploratory applications that collect new knowledge about inaccessible or hazardous locations, such as monitoring volcanic eruptions and wastewater. In such scenarios, unexpected behavior patterns may emerge, which could cause initial system solutions to malfunction. However, robust ML algorithms can adapt to the newly acquired knowledge and improve model adaptability.

Drawbacks

- ML techniques cannot provide immediate and accurate predictions as they require learning historical data. The performance of the system is dependent on the quantity of historical data provided as input, and a more significant amount of data can improve performance accuracy. However, due to the resource limitations in WSNs and the high computational complexity of ML, implementing it can require more energy. To address this, energy efficiency can be improved by implementing ML in a centralized manner.

- The dynamic actions performed by ML are based on historical data, but verifying the accuracy of predictions in real time can be challenging. Diagnosing and correcting errors during network operation can be difficult and involve complex processes associated with the source code.
- It can be challenging to determine which type of ML method is most suitable for a given action taken by WSNs. Simulating when and where the ML algorithm needs to respond can also be difficult.
- ML methods can consume significant computational time and hardware resources, which can be problematic in WSNs with limited resources. To address this issue, centralized algorithms may be preferable to distributed ones.

2.10 Introduction of DRL

DRL is an advanced branch of machine learning that combines deep learning techniques with reinforcement learning algorithms. It represents a powerful approach for training agents to make sequential decisions in complex and dynamic environments. DRL enables machines to learn from interactions with the environment, receive feedback through rewards, and iteratively improve their decision-making abilities.

At its core, DRL employs deep neural networks as function approximations to capture and model the state-action value function, commonly known as the Q-function. The Q-function estimates the expected cumulative rewards for taking specific actions in different states of the environment. By utilizing deep neural networks, DRL algorithms can effectively handle high-dimensional and raw input data, such as images or sensor readings, enabling agents to learn directly from raw sensory inputs.

The training process in DRL revolves around the concept of reinforcement learning, which involves an agent interacting with an environment to maximize cumulative rewards. The agent takes actions based on its current state, receives feedback through rewards or penalties, and learns to improve its decision-making through trial and error. DRL algorithms leverage the principles of reinforcement learning, such as the Markov Decision Process and the Bellman equation, to optimize the agent's policy over time.

One of the key advantages of DRL is its ability to learn complex and hierarchical representations of the environment. Deep neural networks can automatically extract abstract features from raw data, allowing DRL agents to uncover intricate patterns and structures within the environment. This enables them to handle environments with high-dimensional state spaces, making DRL suitable for a wide range of applications, including robotics, game playing, finance, and autonomous driving.

Furthermore, DRL algorithms can handle environments with sparse or delayed feedback, where the rewards are not immediately evident. By exploring and exploiting the environment, DRL agents learn to make long-term decisions, balancing immediate rewards with future potential gains. This temporal credit assignment is essential for solving tasks that require strategic planning and decision-making over extended periods.

In recent years, DRL has achieved remarkable successes, surpassing human-level performance in complex tasks such as playing video games, mastering board games like Go and chess, and controlling robotic systems. These achievements have demonstrated the potential of DRL to tackle real-world problems that were previously considered challenging or intractable.

2.10.1 Components of DRL

DRL consists of several key components that work together to enable agents to learn and make sequential decisions in complex environments. These components include the agent, the environment, the action space, the state space, the reward system, and the learning algorithm. Each component plays a crucial role in the DRL framework and contributes to the overall significance of DRL in solving complex tasks.

- **Agent:** The agent is the entity that interacts with the environment and learns to make decisions. It can be represented by a neural network or any other function approximately capable of mapping states to actions. The agent's objective is to maximize the cumulative rewards it receives from the environment by selecting optimal actions based on its current state.

- **Environment:** The environment represents the external system with which the agent interacts. It provides the agent with observations or states, receives the agent's actions, and returns rewards or penalties based on the agent's actions. The environment can range from simulated virtual environments to physical systems, depending on the application domain.

- **Action Space:** The action space defines the set of possible actions that the agent can take in a given state. It can be discrete, where the agent chooses from a predefined set of actions, or continuous, where the agent selects a value from a continuous range. The action space depends on the specific task or problem being addressed.

- **State Space:** The state space represents the set of all possible states that the agent can perceive from the environment. It can be discrete, where each state is distinct and separate, or continuous, where states exist in a continuous range. The state space encapsulates the relevant information necessary for the agent to make decisions.

- **Reward System:** The reward system provides feedback to the agent based on its actions in the environment. It assigns numerical values, known as rewards, to different states and actions. The agent's objective is to maximize the cumulative rewards over time. The reward system guides the agent's learning process by providing signals that indicate the desirability or undesirability of its actions.

- **Learning Algorithm:** The learning algorithm is responsible for updating the agent's decision-making policy based on the feedback received from the environment. It employs techniques from reinforcement learning, such as Q-learning, policy gradients, or actor-critic methods, to iteratively improve the agent's performance. The learning algorithm enables the agent to learn from past experiences, explore new actions, and exploit learned knowledge to make better decisions in the future.

  The significance of these components in DRL lies in their ability to address complex tasks and learn from raw sensory inputs. By combining deep learning techniques, which allow for the processing of high-dimensional data, with reinforcement learning algorithms, DRL can handle challenging problems that were previously considered intractable. DRL agents can learn directly from raw sensory inputs, such as images or sensor readings, enabling them to understand and interpret complex information in the environment.

Additionally, the agent-environment interaction in DRL allows for continuous learning and adaptation. As the agent explores the environment and receives feedback through the reward system, it can adjust its decision-making policy accordingly. This adaptability makes DRL well-suited for dynamic and changing environments, where optimal strategies may evolve over time.

Moreover, DRL's ability to handle sequential decision-making and long-term planning sets it apart from other machine-learning approaches. By considering the future consequences of its actions and optimizing for cumulative rewards, DRL agents can make decisions that balance short-term gains with long-term objectives. This temporal credit assignment is crucial in real-world applications where actions have delayed or cascading effects.

Eventually, the components of Deep Reinforcement Learning - the agent, environment, action space, state space, reward system, and learning algorithm - work together to enable agents to learn from raw sensory inputs and make sequential decisions.

2.10.2 Application of Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) has emerged as a powerful and versatile technique with a wide range of applications across various domains. Combining deep learning and reinforcement learning, DRL enables agents to learn optimal decision-making policies through trial-and-error interactions with their environment. The following are some of the key applications where DRL has shown significant promise.

- **Robotics:** DRL has revolutionized the field of robotics by enabling autonomous robots to learn complex tasks and control policies. It has been successfully applied to tasks such as robotic manipulation, grasping objects, locomotion, and even playing games like chess and Go. DRL allows robots to learn from their own experiences and adapt their actions in real-time, leading to more robust and intelligent robotic systems.

- **Game Playing:** DRL has demonstrated remarkable success in playing a variety of games, both in traditional board games and video games. Notably, DRL algorithms have achieved superhuman performance in games like Atari, Dota 2, and AlphaGo, surpassing human-level capabilities. By learning directly from raw sensory inputs, DRL agents can devise sophisticated strategies and exhibit adaptive gameplay.

- **Autonomous Vehicles:** DRL has promising applications in autonomous driving, where agents learn to navigate complex traffic scenarios and make decisions in real-time. DRL models can learn to perceive the environment from sensor inputs, such as camera data, and make appropriate decisions, such as lane changing, overtaking, or yielding, while adhering to traffic rules and safety constraints.

- **Healthcare:** DRL holds great potential for healthcare applications, including personalized treatment recommendation systems, disease diagnosis, and medical imaging analysis. DRL models can learn optimal treatment policies by considering patient-specific characteristics and optimizing treatment outcomes. Moreover, DRL has been used to improve medical imaging analysis, aiding in the interpretation of radiological images and facilitating early detection of diseases.

- **Finance:** DRL has found applications in financial domains, such as algorithmic trading and portfolio management. By learning optimal trading strategies from historical market data, DRL agents can make informed decisions on buying, selling, and managing financial assets. DRL's ability to adapt to changing market conditions and complex patterns provides an advantage in the dynamic and uncertain financial landscape.

- **Resource Management:** DRL is valuable for optimizing resource allocation and management in various domains. For example, in energy systems, DRL can learn to control and optimize the energy consumption of buildings or allocate energy resources efficiently. In wireless communication networks, DRL can optimize spectrum allocation and manage network resources for improved performance and user experience.

These applications only scratch the surface of the potential of DRL. The combination of deep learning and reinforcement learning opens doors to solving complex decision-making problems in various domains, making DRL a powerful and versatile approach with far-reaching implications.

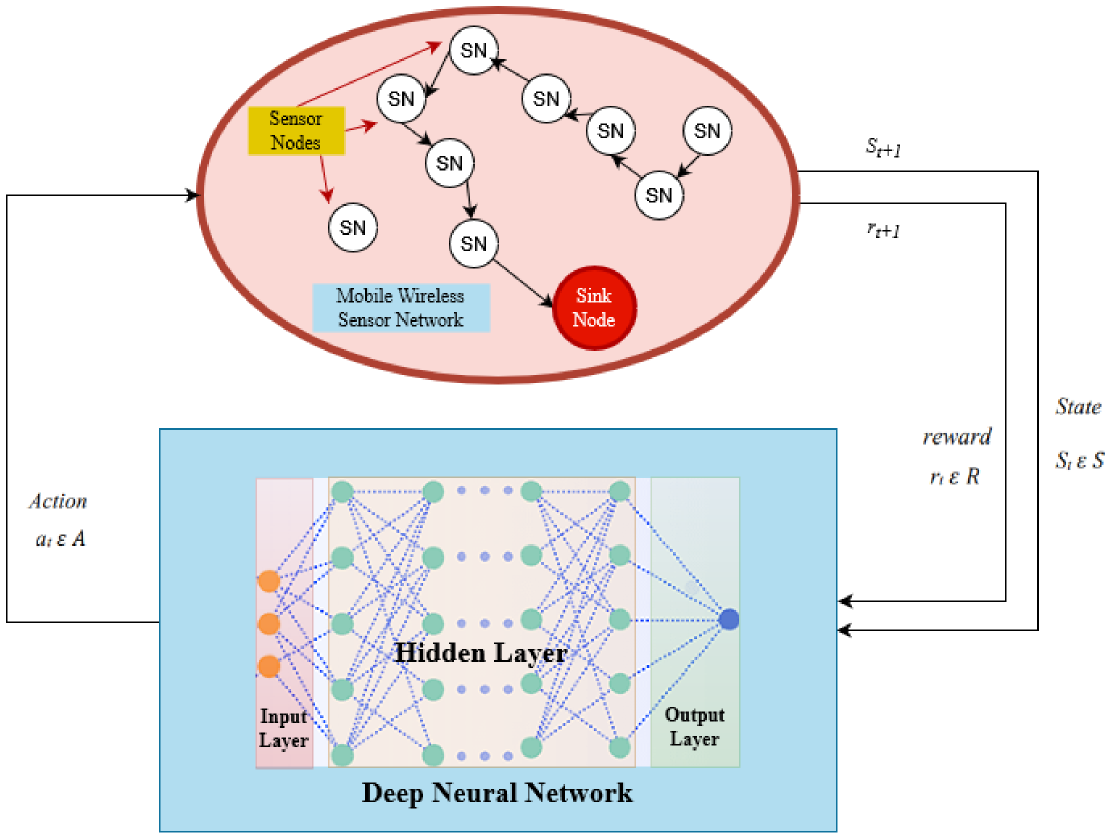


Figure 2.6: Deep RL Process

The DRL model addresses this issue by combining RL and deep learning (DL) techniques. The DRL model uses a deep neural network (DNN) to approximate the Q-values functions. DRL parameters (state and reward) can be assigned based on different 5G system objectives, such as power consumption, state of RRHs, user demand, channel gains or throughput maximization, etc. The DQN operates similarly to the Q-value function except for the addition of neuron and replay memory. All input states are transferred to different NN layers, each with different weight factors θ. Finally, DQN generates the Q-value outputs with respect to possible actions. Furthermore, an experience replay memory is also used, where the network training is done by sampling a small batch of tuples.

DRL approach fundamentally makes use of deep neural networks function approximation properties to effectively overcome problems of high dimensionality and complexity. Furthermore, it plays a significant role to develop an appropriate correlation between each state-action combination and its corresponding value (i.e., cumulative reward) by learning network parameters and training samples as depicted in Figure 2.6.

2.10.3 The key features of DRL

- DRL can enable MWSN to learn the most energy-efficient routing paths that extend the lifespan of MWSNs that are constantly changing (Mohammad Abu Alsheikh et al., 2014).

- Simplify complex routing problems by breaking them down into smaller sub-problems. In each sub-problem, nodes create graph structures by only considering their nearby neighbors, resulting in low-cost, efficient, and real-time routing.

- Use straightforward computational methods and classifiers to meet Quality of Service (QoS) requirements in routing problems.

Chapter 3

Literature Review

This Chapter talks about Deep Reinforcement Learning (DRL) as this lays the necessary foundation for the proposed work. The introduction begins, followed by a discussion on routing protocols for MWSNs, which presents an overview of different types of routing protocols. In particular, the Optimized link state routing (OSLR) protocol is discussed. Subsequently, routing protocols based on DRL with reference to WSNs are elaborated; several DRL-based routing protocols are reviewed for MWSNs that exist in the literature. After that, some well-known techniques that are the basis of DRL are discussed, such as ANNs, CNNs, and RNNs, along with their architecture. The pros and cons are also studied of using DRL in MWNS. Finally, it provides details of the performance metrics used in evaluating various routing protocols in the context of MWSNs and which metrics are selected to evaluate routing protocol.

3.1 Introduction

Deep Reinforcement Learning (DRL) is a ML approach used for classification, and it is a subcategory of Artificial Neural Networks (ANNs).DRL approaches are data learning representation methods with multi-layer representations (between the input layer and output layer). It composes of simple non-linear modules that transform the representation from the lower layer to the higher layer to achieve the best solution (LeCun, Bengio, and G. Hinton, 2015). Communication patterns and information processing inspire it in human nerve systems which has some differences from the functional and structural properties of human brains (Marblestone, Wayne, and Kording, 2016). The key benefits of DRL are extracting features from the data, working with or without labels, and it can be trained to fulfill multiple objectives. It can be helpful in various domains such as bio-informatics, business intelligence, medical image processing, social network analysis, speech recognition, and handwriting recognition. The advantages of deep learning are recently attracting the WSNs researchers. Embedding deep learning in WSNs has solved various issues such as routing (Lee, 2017), data quality estimation

(Yuzhi Wang et al., 2017), and energy harvesting (F. Chen, Z. Fu, and Z. Yang, 2019).

3.1.1 Routing Protocols For MWSNs

Routing protocols are typically categorized based on various characteristics, such as network structure and application, which suggests that no one routing mechanism can effectively and practically serve all types of WSN applications (Ketshabetswe et al., 2019; Biradar et al., 2009). As a result, routing protocols can be divided into several categories, including network structure-based protocols, operation-based protocols (Kaganurmath and Ganashree, 2016), and path establishment-based protocols.

3.1.2 Network Structure-Based Protocols

The network structure-based approach consists of various routing methods, such as flat-based routing which is suitable for large networks with numerous sensor nodes, where it is not feasible to assign global identifiers (IDs) to every node (S. K. Singh, 2010). Consequently, data-centric routing approaches are used, where all nodes in the network are considered equal and perform the same function. Some of the well-known flat-based routing protocols include Flooding (Ketshabetswe et al., 2019), Gossiping (CHOKSI and DESAI, 2012), Directed Diffusion (Intanagonwiwat, Govindan, and Estrin, 2000), Rumor Routing (Echoukairi, Bourgba, and Ouzzif, 2016), Minimum Cost Forwarding Algorithm (Nikolaos A Pantazis, Stefanos A Nikolidakis, and Dimitrios D Vergados, 2012), and Active Query forwarding in sensor networks (P. Kumar, M. Singh, and Triar, 2012). The network structure-based protocols can be further categorized into flat-based, hierarchical-based, and location-based protocols.

Hierarchical or cluster-based routing is a popular routing approach that divides the network into clusters controlled by a cluster head to optimize energy consumption. This type of routing operates in two layers, where cluster heads collect data from cluster members, process it, and forward it to the sink through other cluster heads. Clustering-based routing has proven effective in optimizing power consumption and extending the network's lifetime. Examples of clustering-based routing techniques include Low-Energy Adaptive Clustering Hierarchy (LEACH) [33], Power-Efficient Gathering in Sensor Information Systems (PEGASIS) (Ketshabetswe et al., 2019), Threshold-sensitive Energy Efficient Protocols (TEEN) (Nikolaos A Pantazis, Stefanos A Nikolidakis, and Dimitrios D Vergados, 2012), and Adaptive Periodic Threshold-sensitive Energy Efficient protocol (APTEEN).

Finally, location-based routing uses nodes' location information to make routing decisions. Nodes know their location information using the Global Positioning System (GPS) or other technologies. This location information can be used by nodes to optimize routing paths by considering energy consumption (A.

Kumar et al., 2017). Examples of location-based routing protocols include Geographical Adaptive Fidelity (GAF), Geographical and Energy Aware Routing (GEAR), and Minimum Energy Communication Network (MECN) (Chaudhary and Vatta, 2014).

3.1.3 Operation-Based Protocols

Operation-based protocols can be divided into multipath-based (Radi et al., 2012), query-based, negotiation-based, QoS-based, and coherent/non-coherent-based protocols. This protocol is classified into two main approaches based on operation and functionality. The first approach is focused on the routing algorithm and its characteristics. The second approach, however, considers how the protocol operates and functions. This approach includes several categories of routing protocols, such as multipath routing protocols, query-based routing protocols, and negotiation-based routing protocols. Multipath routing protocols utilize multiple paths to transmit data packets to the base station. By using alternative paths, these protocols can reduce the end-to-end delay. Examples of multipath routing protocols include DD and SPIN (Almesaeed and Jedidi, 2021). Query-based routing is another category of protocols that allows the base station to send query messages to nodes requesting specific information. Negotiation-based routing protocols minimize data redundancy by negotiating between neighboring nodes and selecting the optimized route. SPIN is an excellent example of a negotiation-based routing protocol (Dwivedi and Vyas, 2010).

3.1.4 Path Establishment-Based Protocols

The final group of protocols is path establishment-based routing protocols. These protocols determine routing paths using one of three methods: proactive, reactive, or hybrid. Proactive protocols calculate all possible routing paths and save them in a routing table in each node, even when no data is being transmitted. Conversely, reactive routing protocols calculate routing paths only when they are needed. Hybrid routing protocols combine elements of both proactive and reactive approaches (P. Kumar, M. Singh, and Triar, 2012). Routing protocols are classified based on the process they used to discover the routes.

Proactive protocols They are also known as table-driven routing protocols, because they maintain the routing tables for the complete network by passing the network information from node to node and the routes are pre-defined prior to their use and even when there is no traffic flow. The most commonly proactive routing protocol used in Mobile Wireless Sensor Networks (MWSNs) is Optimized link state routing (OLSR) as detailed in 3.1.4.1.

Reactive protocols Reactive routing protocols do not maintain the whole network topology rather, they are activated just on demand when any node wants to send data to any other node. So the routes are created on demand when queries are initiated. The most commonly used reactive routing protocols are Ad-hoc on-demand distance vector (AODV) and Dynamic source routing (DSR) as discussed in 3.1.5.

Hybrid protocols Hybrid Routing Protocols have the merits of proactive and reactive routing protocols by neglecting their demerits. Routing is one of the primary challenges in MWSNs because of the limited power supply, low transmission bandwidth, less memory capacity, and processing capacity. In WSNs, sensors are deployed randomly in the environment, and each sensor node collects the data from the environment and transmits it to the BS for further processing. Figure 6.3 shows the multi-hop transmission from SNs to BS. In general, the nodes which are near the BS consume more energy because they serve as relay nodes. The goal of the routing protocol design is to reduce the energy consumption of SNs and increase the network lifetime. Recently, several routing methods (Hammoudeh and Newman, 2015; X. Liu, 2017; Asif et al., 2017) are developed for WSNs by the researcher using different approaches.

3.1.4.1 Optimized Link State Routing Protocol(OLSR) It is a proactive and table-driven routing protocol used inused in Mobile Ad hoc Networks (MANETs) and Mobile Wireless Sensor Networks (MWSNs). It is designed to efficiently route data packets in a network with a large number of nodes and frequent changes in the network topology due to mobility (Mouiz et al., 2019). The OLSR operates by maintaining a topology database that contains information about the nodes in the network and the links between them. Each node periodically broadcasts information about its neighbors and the links to those neighbors. This information is used to update the topology database and to calculate the shortest path between any two nodes in the network. The OLSR protocol uses a multipoint relaying (MPR) technique to reduce the number of broadcast messages and minimize the network overhead. Each node selects a set of MPRs, which are responsible for forwarding the broadcast messages to their respective destinations. This reduces the number of duplicate messages and minimizes the transmission delays.

The OLSR protocol also includes a mechanism for detecting and repairing broken links in the network. When a link failure is detected, the affected nodes update their topology databases and recalculate their routes to avoid the broken link. There are variants of OLSR protocol that are designed to be highly efficient and scalable, making it well-suited for large-scale MWSNs with high mobility and frequent topology changes (Sabor et al., 2017).

3.1.4.2 Ad-hoc on-demand distance vector (AODV) AODV is reactive on request protocol. AODV is engineered for Mobile infrastructure-less networks. It employs the on-demand routing methodology for the formations of

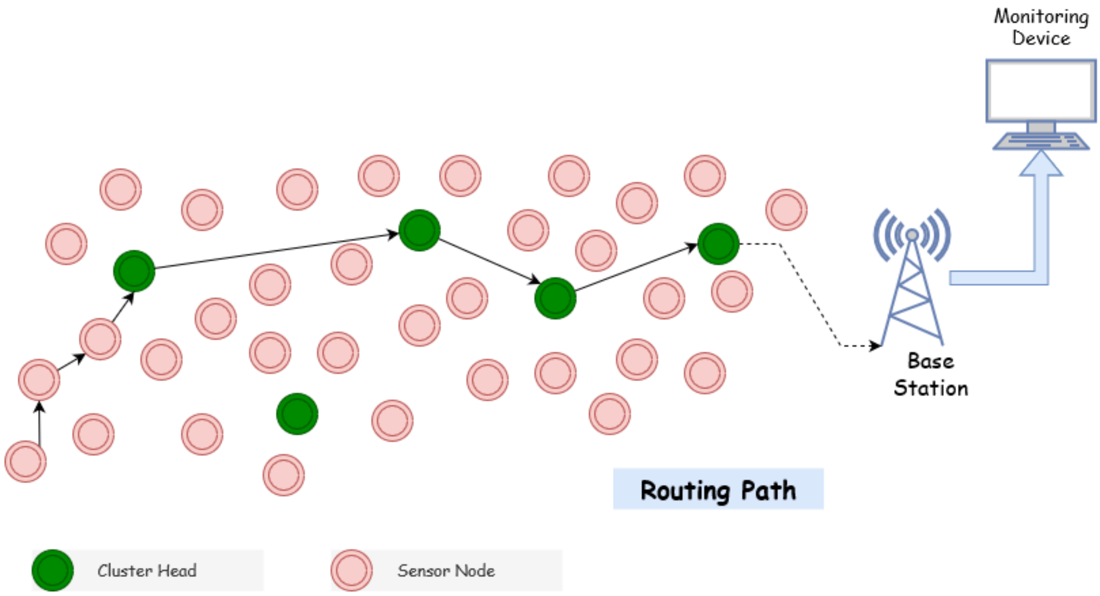


Figure 3.1: Routing in MWSNs

routes among network nodes. The path is established solitary when the source node wants to direct packs of data and pre-set route is maintained as long as the source node needs.

3.1.4.3 Dynamic source routing (DSR)

DSR can be reactive or on-demand as its name shows it uses source routing instead of routing tables. Routing in DSR is divided into two parts: route discovery, and route maintenance. It is designed to allow nodes in a network to dynamically discover and maintain routes for data packets to be transmitted between source and destination nodes without needing a fixed infrastructure or centralized control. When a source node wants to send data to a destination node and does not have a route to the destination in its routing table, it initiates a route discovery process. During this process, the source node broadcasts a route request (RREQ) packet. This packet contains the source and destination addresses of the data packet. Intermediate nodes that receive the RREQ packet check if they have a route to the destination in their routing tables. If they do, they send a route reply (RREP) packet back to the source node. The RREP packet contains the route information the source node can use to reach the destination. The RREP is sent hop-by-hop back to the source node.

As the source node receives the RREPs, it constructs a route to the destination by following the path indicated in the RREP. The source node then caches this route for future use. During data transmission, the source node encapsulates the data packets with the route information and sends them to the next hop according to the learned route. Intermediate nodes forward the packet based on the route information until the destination is reached. The DSR protocol is "on-demand," meaning that routes are only established when needed. Routes are not maintained when not in use, which makes the protocol more adaptable to changes in network topology, mobility, and connectivity.

3.1.5 DRL-Based Routing in MWSN

In (Lee, 2017), a DL-based routing protocol has been introduced with the BS as an infrastructure. It means the route is maintained, assigned, and recovered by the BS. This work proposed a DL-based algorithm that adopts dynamic routing in a mobile sensor network. The BS initially creates a list of virtual routing paths, and from them, it identifies the optimal route. This algorithm overcomes congestion and packet loss and power management. In (Kazemeyni et al., 2014), a Bayesian learning method based optimal routing prediction model has been developed for both decentralized and centralized versions. This approach also performs the scheduling approach while routing the data to balance energy consumption. This algorithm is much more suitable for decentralized than centralized.

Youssif, A. A *et al.,* in (El, Youssif, Ghalwash, et al., 2016) have used a well-known k-means classification algorithm to find optimal clustering in WSNs for routing. This algorithm provides a better packet delivery ratio, and throughput, lowering energy consumption and controls the traffic overhead. In (Ray and De, 2016), authors have proposed energy efficient clustering protocol using a k-means (EECPK-means) algorithm to find the optimal center point from the cluster from a random initial center point. It selects optimal CHs based on the Euclidean distance and residual energy of the SNs in WSNs. EECPK-means algorithm finds the efficient multi-hop communication path from the CHs to BSs. This algorithm avoids data loss and balances the energy consumption of the SNs.

In (Shashikala and Kavitha, 2018), a secure cluster-based routing protocol has been developed to enhance the network lifetime for WSNs. In this approach, cluster heads are selected based on their distances and residual energy. This algorithm mainly focuses on the isolated cluster head and edge node to balance the node energy consumption. In (Kuila and Jana, 2014), authors have presented an efficient routing mechanism based on the transmission range of the SNs and the data forwarding load. In this mechanism, an efficient clustering method was used, which is based on the PSO, to balance the load of the SNs in the network. An ACO-based routing algorithm has been presented in (Y. Sun, Dong, and Y. Chen, 2017) for WSNs. To find the optimal routing, they consider various parameters such as the residual energy of a node, transmission distance, transmission path, and the shortest path between the source nodes to the BS. This algorithm results in minimum energy consumption and prolongs the network lifetime.

The main advantages of DL include its ability to extract high-level characteristics from data, work with or without labels, and be trained to achieve various goals. Many different fields, including bio-informatics, corporate intelligence, medical image processing, social network analysis, speech recognition, and handwriting identification can benefit from it. The Table **??** summarizes existing works based on DL/DRL routing in nutshell.

The literature study indicates that the features of DL/DRL can be utilized for the performance improvement of the WSN. To deal with the energy hole problem in clustering, researchers have reported various methods. However, the deep learning approach is one of the effective methods and can be used (Chang, Yuan, et al., 2019). The main advantages of DL/DRL include its ability to extract high-level characteristics from data, work with or without labels, and be trained to achieve a variety of goals. Many different fields, including bioinformatics, corporate intelligence, medical image processing, social network analysis, speech recognition, and handwriting identification, can benefit from it.

3.2 Reinforcement Learning

There are various methods that can be used for the optimization of various parameters of the MWSNs. Now days Reinforcement learning (RL) is very very popular for intelligent systems. Deep reinforcement learning is gaining research attention due to its unique features and powerful optimizations. RL encompasses a unique collection of tasks that apply to many real-world scenarios and differ from standard machine learning domains in fundamental ways. Generally, machine learning tasks are classified as either supervised (learning from labeled training data) or unsupervised (discovering labels from unlabeled data).

RL, however, is unique in that it does not use labeled training data or seek to discover any labels. Rather, RL tasks focus solely on maximizing a reward obtained from one's environment(14). Examples of RL tasks include playing board or video games, controlling robotic limbs, finding optimal paths through an environment, and many more. In fact, many learning tasks we face as humans every day can be modeled as RL tasks.

The general model for an RL task involves an agent that observes an environment and takes actions based on that observation. For simplicity, it is assumed that the observation-action cycle to occur as discrete timesteps. To facilitate training an agent for an RL task, the agent is provided with a reward value after taking an action at each timestep. Positive rewards can be used to indicate that a desirable action was taken, while negative rewards can indicate less favorable actions. The goal of a machine learning algorithm in the context of RL is to learn a policy that maximizes the total reward over an entire run of the simulation (often called an episode).

To further improve the performance, the Deep learning approach is introduced with Reinforcement learning, i.e., Deep Reinforcement Learning. Deep Reinforcement Learning (DRL) is an evolution of traditional Reinforcement Learning (RL) that incorporates deep neural networks to handle high-dimensional state spaces. DRL uses techniques like Deep Q-Networks (DQN) and policy gradients to approximate value functions and policies, enabling it to tackle complex tasks,

Table 3.1: Literature Review of Deep Reinforcement Learning

Ref	**Techniques**	**Outcomes**	**Features**	**Drawback**
B. Liu et al., 2017	Researchers utilized a DBN, they uncover the correlations between the demand for multi-commodity flow in wireless networks and link usage.	Based on authors' predictions, they eliminate links that are unlikely to be utilized, shrinking the data size for demand-constrained energy optimization. Their approach leads to a 50% reduction in runtime without sacrificing optimality.	The relationship between the input and output in their case is intricate and not readily defined. To unravel this relationship, they employ deep learning techniques, which enable them to deduce the latent or hidden relationship embedded within the complex structure..	Require a large amount of data for training to achieve optimal results
Y. He, C. Liang, et al., 2017 Y. He, Z. Zhang, et al., 2017	The authors of this paper applied deep reinforcement learning to tackle the challenges of caching and interference alignment in wireless networks.	The authors specifically treat the time-varying channels as finite-state Markov channels and use deep Q networks to determine the optimal user selection policy. This innovative framework shows a substantial improvement in both sum rate and energy efficiency compared to existing methods	The proposed method involves training a model to evaluate links based on flow demand vectors. Extraneous links are excluded from the optimization problem through the estimated link values to minimize computation time and storage costs. The approach's effectiveness is evaluated through test samples, and the results illustrate how removing unnecessary links significantly reduces computation time.	Require significant computational resources

Ref	Techniques	Outcomes	Features	Drawback
L. Chen et al., 2018	An automatic traffic optimization technique utilizing a deep reinforcement learning method is presented. The authors designed a two-layer DRL framework that mimics the Peripheral and Central Nervous Systems in animals to resolve scalability issues in data centers.	The authors have implemented multiple peripheral systems at all end-hosts for making local decisions on brief traffic flows. A central system has also been utilized to optimize long traffic flows, which can endure longer delays. The experiments conducted on a testbed of 32 servers demonstrate that the proposed design significantly decreases the traffic optimization turnaround time and the flow completion time, compared to previous methods.	AuTO's scalability owes its success to the separation of time-consuming decision-making processes from quick actions for short tasks, which is achieved through a specific approach called DRL.	Training DRL models typically involves a complex and time-consuming process.
B. Mao et al., 2017b	The authors used a Deep Belief Architectures DBA to determine the next routing node and construct a software-defined router.	Their approach, which considers Open Shortest Path First as the optimal routing strategy, has achieved an accuracy of up to 95% while significantly reducing overhead and delay. Additionally, it results in higher throughput with a signaling interval of 240 milliseconds.	In this paper, the authors propose a supervised deep learning system that constructs routing tables and demonstrate how it can be seamlessly integrated with programmable routers equipped with CPUs and GPUs.	The paper does not explicitly mention scalability.

Ref	Techniques	Outcomes	Features	Drawback
Lee, 2017	Lee et al. utilized a three-layer deep neural network to enhance the efficiency of routing rules by classifying the node degree based on comprehensive information of the routing nodes.	The Viterbi algorithm generates virtual routes based on the classification results and temporary routes.	The proposed technique is a hybrid wireless ad-hoc network routing solution that leverages collaboration between wireless ad hoc networks composed of infrastructure-based wired networks. The approach utilizes the node degree classifier (NDC) results produced by deep learning in conjunction with the Viterbi algorithm to determine the most effective route.	Integrating a deep learning-based routing solution with existing network infrastructure and protocols can be challenging.
B. Mao et al., 2017a	The authors enhance the routing performance by using tensors to represent the hidden layers, weights, and biases in the Deep Belief Networks.	The results illustrate that the proposed approach outperforms the conventional Open Shortest Path First (OSPF) protocol regarding overall packet loss rate and average delay per hop.	In this paper, the authors employ Tensor-based Deep Belief Architectures (TDBAs), an advanced technology, to make decisions based on multiple network traffic factors.	The additional computational overhead introduced by the deep neural network .

Ref	**Techniques**	**Outcomes**	**Features**	**Drawback**
B. Zhao and X. Zhao, 2022	The proposed approach tackles challenges in wireless sensor networks (WSNs) by creating localized subnetworks equipped with amplified relay nodes and a carefully designed operational time cycle. Resource allocation policies are developed using deep reinforcement learning (DRL), treating the optimization problem as a Markov decision process.	The implementation of the suggested approach yields enhanced communication within WSNs. By addressing issues such as channel fading, irregular energy supply, and suboptimal sensor deployment, the proposed method leads to improved overall system performance. Simulation results demonstrate that the developed transmission policies outperform greedy, random, and conservative policies, resulting in higher throughput within localized networks and contributing to the network's overall efficiency.	The wireless sensor network (WSN) is structured into multiple localized subnetworks, each comprising relay nodes with amplification capabilities. The subnetworks operate on a specialized time cycle, ensuring synchronized and efficient data transmission. Deep reinforcement learning (DRL) is used to devise resource allocation strategies that optimize both power and time resources for maximum throughput.	High time complexity

such as playing video games or controlling robots, where traditional RL methods may struggle due to the curse of dimensionality. The details of DRL are discussed in the next section.

3.3 DRL Techniques

Deep reinforcement learning architectures take inspiration from the structure of the human brain, which has a deep architecture. The brain organizes concepts hierarchically, starting with simple concepts and building up to more abstract ones. Researchers have applied this hierarchical learning approach to computers, using multiple levels of abstraction and processing to solve computational problems. There are three main types of deep architectures: generative, discriminative, and hybrid. A generative deep architecture characterizes the high-order correlation properties of input data for synthesis purposes, while a discriminative deep architecture is used for pattern recognition or classification. A hybrid model combines the generative and discriminative architectures to aid in discrimination tasks, using the optimized outputs obtained from the generative architecture (Fadlullah et al., 2017).

It's important to understand that the hybrid deep architecture is not the same as using the outputs of a traditional neural network as inputs for a Hidden Markov Model (HMM) (Morgan, 2011). However, before 2006, except Convolution Neural Networks (CNNs), deep architectures couldn't be effectively trained for any purpose. Despite this, current deep reinforcement learning algorithms rely on multi-layer architectures, as described in Bengio et al.'s research (Bengio, Courville, and Vincent, 2013). The primary difference is the introduction of Greedy Layer-Wise unsupervised pre-training, designed to learn a hierarchy of features from a massive, unlabeled dataset one level at a time.

To put it simply, at each level of the hierarchy, the learned features undergo a new transformation that serves as the input for the next level (G. E. Hinton, Osindero, and Teh, 2006; Bengio, Lamblin, et al., 2006). These features can then be used as input for a standard supervised ML predictor, such as Support Vector Machines (SVMs) or Conditional Random Field (CRF), or for a deep supervised neural network. In the case of deep learning architectures, focused on the latter. Each iteration of unsupervised feature learning produces a set of weights that create a new layer in the deep neural network. Ultimately, the layers of learned weights can be used to initialize a deep supervised predictor, such as a neural network classifier or a deep generative model like a Deep Boltzmann Machine (DBM) (Srivastava, Salakhutdinov, and G. E. Hinton, 2013).

Below, a brief overview is provided of pertinent deep reinforcement learning techniques, including Artificial Neural Networks and General Architecture of Artificial Neural Networks,

3.3.1 Artificial Neural Networks (ANN)

Artificial neural networks (ANNs) possess a remarkable ability to emulate human intelligence. They draw inspiration from the structure and functionality of biological neural networks, enabling them to learn from complex or imprecise data. In wireless communications, ANNs play a crucial role in investigating and forecasting network and user behavior. This information proves invaluable in addressing a wide range of wireless networking challenges, such as cell association, spectrum management, computational resource allocation, and cached content replacement, as elaborated in subsequent sections.

Furthermore, the proliferation of smart devices and mobile applications has significantly elevated human interaction with mobile systems. By leveraging trained ANNs, which can be likened to "experts" in processing human-related data, wireless networks gain the ability to anticipate users' future behaviors. Consequently, they can devise optimal strategies to enhance the quality of service (QoS) and reliability. While the utilization of ANNs in real-time wireless sensor network (WSN) applications necessitates higher computational resources, it offers tremendous potential for enhancing efficiency across various aspects of WSNs, including localization (Banihashemian, Adibnia, and Sarram, 2018; Phoemphon, So-In, and Niyato, 2018), detecting faulty sensor nodes (Chanak and Banerjee, 2016), routing (Mehmood et al., 2017; Gharajeh and Khanmohammadi, 2016).

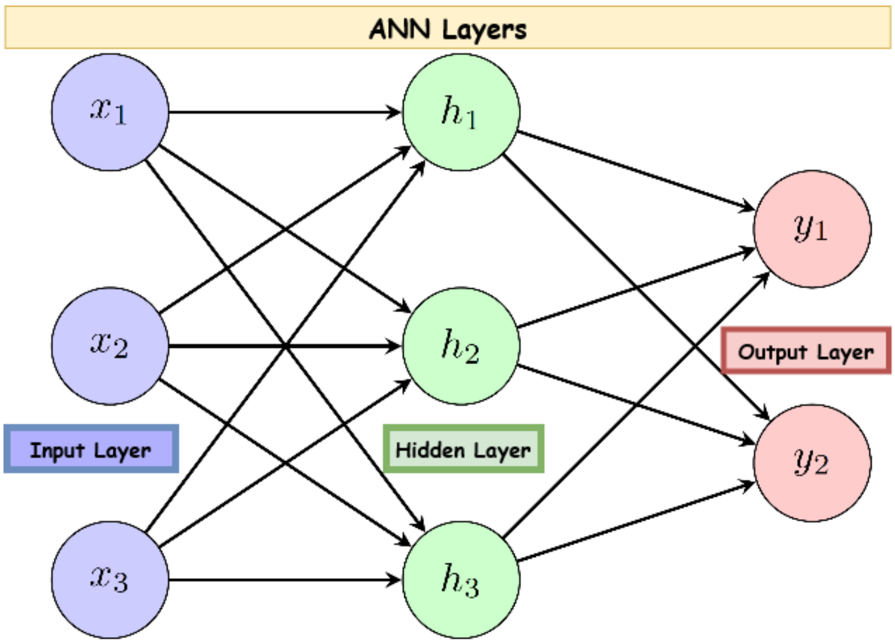


Figure 3.2: A simple ANN architecture with different layers

The architecture of ANNs consists of several simple, highly interconnected processing elements known as neurons, which are used to mimic how the human brain learns. ANNs are essentially an artificial model of a human nervous system whose base elements are also neurons used to process information in the sense of cognition and transmit this information signal in the nervous system.

Figure 3.2 shows the basic layered structure of an ANN. Each ANN contains three types of layers called input layers, one or more hidden layer(s), and output layers. ANN classifies complex and non-linear data sets very easily, and there is no restriction for the inputs like other classification methods.

A neuron consists of a nucleus, dendrites, and axons. Neurons are connected by dendrites and axons, as shown in Figure 3.2. The connection point between two neurons is known as a synapse. The information signal transmitted to a neuron will change its membrane potential. If this change makes the neuron's membrane potential exceed a certain value, it will send a signal to all its connected neurons. This is how signals propagate through the human nervous system. ANNs use artificial neurons to mimic this operation of the human nervous system, thus enabling artificial intelligence. Mathematically, an artificial neuron consists of the following components:

In general, the main components of an ANN that consists of multiple neurons will include the following:

- Input layer that consists of a number of neurons used to represent the input signal which will be transmitted in the neurons.
- Output layer that consists of a number of neurons used to represent the output signal.
- Hidden layer that consists of a number of neurons used to mimic the human brain.
- Input weight matrix that represents the strength of the connections between the neurons in the input layer and the neurons in the hidden layer.
- Neuron weight matrix that represents the strength of the connections between the neurons in the hidden layer.
- Output weight matrix that represents the strength of the connections between the neurons in the output layer and the neurons in the hidden layer.

3.3.2 Convolution Neural Networks (CNNs)

CNNs are a kind of deep neural network that follows a feedforward approach and includes convolutional layers, pooling layers, and fully connected layers. They are intended to analyze data with multiple arrays, such as color images, audio spectrograms, and videos, and benefit from the properties of these signals such as local connections, shared weights, pooling, and multi-layer architecture (Fadlullah et al., 2017).

CNNs draw inspiration from the simple and complex cells in visual neuroscience and can automatically identify important features without any human supervision, making them well-suited for tasks such as image recognition, speech processing, and face recognition, among others. Unlike traditional fully connected (FC) networks, CNNs utilize shared weights and local connections to make optimal use of 2D input-data structures like image signals, and this process requires fewer parameters, simplifying the training process and speeding up the network. The primary benefit of CNNs compared to their predecessors is that they can identify relevant features automatically, without human intervention (Alzubaidi et al., 2021).

CNNs are particularly useful in image recognition since they allow image-specific features to be encoded into the architecture, making the network more adept at handling image-based tasks while minimizing the parameters needed to create the model. This stands in contrast to traditional ANNs, which face difficulties in handling the computational complexity of processing image data, particularly with larger image dimensionality, such as high-resolution images. To address this, convolution layers were introduced to exploit shared weights and local connections, which considerably simplifies the training process and speeds up the network (Y. Li, 2017).

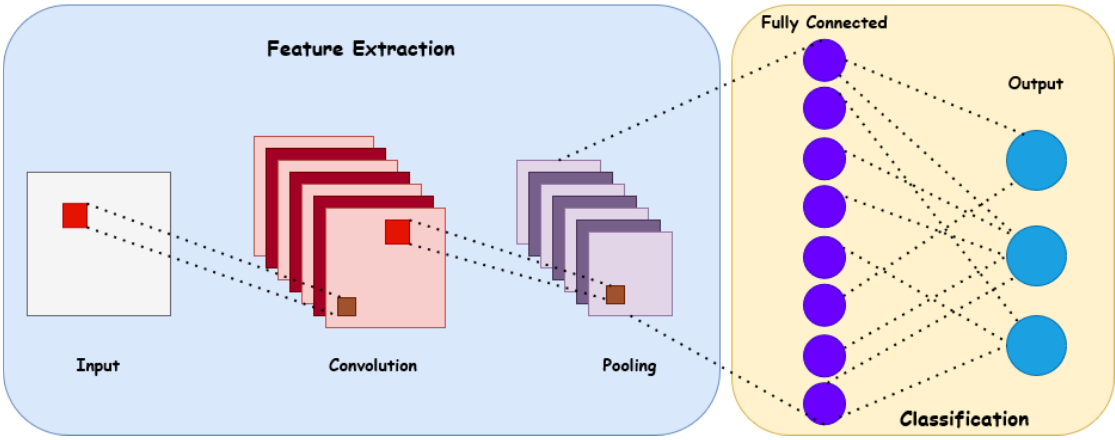


Figure 3.3: A simple CNN architecture with different layers

A CNN architecture consists of two primary components: a convolution tool that identifies and isolates different features of an image through feature extraction, and a fully connected layer that utilizes the output of the feature extraction process to predict the image's class. The feature extraction process typically comprises several pairs of convolutional and pooling layers. By generating new features that summarize the original set of features, the CNN model aims to decrease the number of features present in a dataset. The CNN architecture diagram, depicted in Figure 3.3, shows the various layers involved in the feature extraction process as detailed below.

Convolutional layers: These are the key building blocks of CNNs, and they perform feature extraction by convolving the input data with a set of learnable filters, also known as kernels or weights. These filters are slid over the input data to detect local patterns, such as edges, corners, and blobs, in different input regions. The output of the convolutional layer is a feature map, which represents the activation of each filter across the input data. The convolutional layers also employ shared weights and local connections to fully use the 2D input-data structures like images.

Pooling layers: They are used to reduce the spatial dimensionality of the feature maps generated by the convolutional layers. The most common type of pooling is max pooling, where the maximum value within a small spatial neighborhood is selected and downsampled, which helps to make the network more robust to translation invariance.

Fully connected layers: These layers are typically used at the end of the network to produce a classification output based on the features extracted from the earlier layers. These layers are similar to the layers in traditional artificial neural networks (ANNs), and they connect all the neurons from one layer to the neurons in the next layer, which provides a rich representation of the input data.

Combining convolutional layers, pooling layers, and fully connected layers allows CNNs to automatically learn and identify relevant features from the input data without human supervision. CNNs are particularly well-suited for image-related tasks, but they have also been successfully applied to other domains, such as speech processing, natural language processing, and reinforcement learning.

3.3.3 Recurrent Neural Networks (RNNs)

RNNs are a type of neural network commonly used for processing sequential data such as time series, speech, and text. Unlike traditional feedforward neural networks, RNNs have a feedback loop that allows information to be passed from one-time step to the next. RNNs can use the output from a previous time step as input to the current time step (Fadlullah et al., 2017).

The basic building block of an RNN is called a recurrent cell. This cell contains a hidden state updated at each time step using the current input and the previous hidden state. The hidden state captures information about the context of the input sequence up to the current time step. The output of the recurrent cell can be fed into another layer or used as a prediction for that time step (Alzubaidi et al., 2021).

RNNs are trained using backpropagation through time (BPTT), which is a variation of the backpropagation algorithm used in traditional neural networks. BPTT is used to calculate the gradient of the loss function with respect to the model parameters at each time step. The gradients are then used to update the model parameters using an optimization algorithm such as gradient descent.

One of the main advantages of RNNs is their ability to handle variable-length sequences. They can process sequences of any length, unlike traditional neural networks that require fixed-length inputs. This makes RNNs particularly useful for tasks such as speech recognition, natural language processing, and time series prediction. However, RNNs can suffer from the vanishing gradient problem, which occurs when the gradients become very small as they are propagated back through time. This can make it difficult to train RNNs to capture long-term dependencies in sequential data. Several variations of RNNs have been developed to address this issue, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), which use more sophisticated update rules for the hidden state.

The basic architecture of an RNN is shown in Figure 3.4, which consists of a single recurrent layer that contains recurrent connections, allowing information to be passed from one time step to the next. The input at each time step is processed by the recurrent layer, which produces an output and also updates its hidden state. The output can be fed back into the network at the next time step, allowing the network to take into account its previous output when processing the next input.

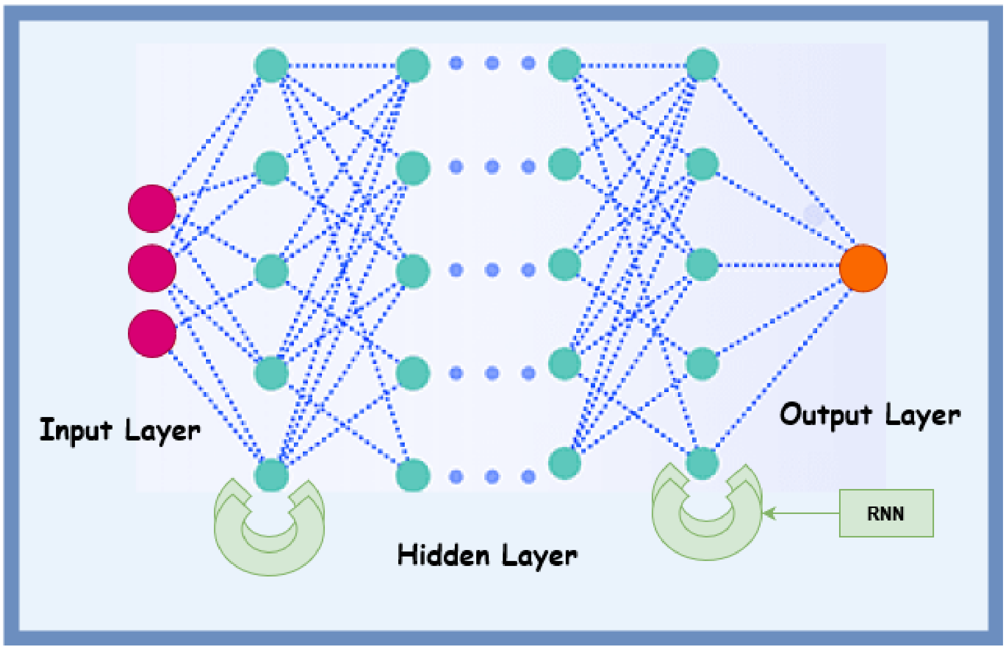


Figure 3.4: A simple RNN architecture with different layers (Fadlullah et al., 2017).

3.4 Challenges in DRL

Despite the remarkable advantages and broad applications of Deep Reinforcement Learning (DRL), several significant challenges must be addressed to maximize its potential. One of the primary issues is the high computational complexity and resource requirements necessary for training DRL models. These models typically demand substantial computational power, often necessitating advanced hardware like GPUs or TPUs, which can be costly and inaccessible for some users. Additionally, the training process is time-intensive, frequently requiring extensive data and prolonged periods to achieve convergence. This intensive resource demand poses a considerable barrier, particularly for applications in academia or small enterprises. Moreover, DRL suffers from sample inefficiency, relying heavily on large numbers of interactions with the environment to learn effective policies, which can be impractical in real-world scenarios where data acquisition is costly and limited, such as in bioinformatics and medical image processing (LeCun, Bengio, and G. Hinton, 2015), (Marblestone, Wayne, and Kording, 2016).

Another critical challenge in DRL is ensuring stability and convergence during training. The interdependent nature of reinforcement learning, where actions influence subsequent states and rewards, often leads to instability and the risk of catastrophic forgetting. Techniques like experience replay and target networks help mitigate these issues, but achieving consistent and reliable convergence remains a hurdle. Additionally, the exploration-exploitation dilemma complicates the learning process, requiring a delicate balance to avoid suboptimal policies. Beyond these technical challenges, DRL models often lack interpretability, making it difficult to understand and trust their decisions, especially in sensitive fields like business intelligence and medical diagnostics. This opacity can hinder the adoption of DRL in critical applications where decision transparency is essential. Furthermore, ensuring that DRL models generalize well to new environments and tasks is challenging, limiting their broader applicability. These issues are compounded by domain-specific constraints, such as energy efficiency in Wireless Sensor Networks (WSNs), which require tailored solutions to optimize performance and resource usage (Ma et al., 2016; C. Li et al., 2015), (Lee, 2017), (Yuzhi Wang et al., 2017), (F. Chen, Z. Fu, and Z. Yang, 2019).

3.5 Pros and Cons of DRL in MWSN Routing

There exist both benefits and challenges of utilizing DRL for MWSN routing as detailed below.

Benefits

- **Enhanced Routing Performance:** DRL can help MWSNs to learn from past experiences and optimize their routing strategies, leading to improved routing efficiency. By learning to make better routing decisions, the network can reduce latency, packet loss, and energy consumption.

- **Routing Flexibility in Dynamic Environments:** MWSNs operate in dynamic environments where network topology and traffic patterns can change rapidly. DRL algorithms can help MWSNs to adapt to these changes quickly by learning from new experiences and adjusting their routing strategies accordingly.

- **Improved Network Resilience:** MWSNs are often deployed in harsh and challenging environments where nodes can fail or become disconnected. DRL can help the network to identify alternative routes and adapt to changes in network conditions, increasing overall network resilience.

- **Scalability:** As the size and complexity of WSNs increase, traditional routing algorithms may struggle to efficiently manage the network due to the large number of nodes and the high degree of mobility. In contrast, DRL algorithms can handle large networks with many nodes and can adapt to changes in network topology and traffic patterns. This makes them well-suited for complex mobile WSNs, where the ability to scale to larger networks is crucial for effective routing.

- **Optimal Resources Utilization:** Traditional routing protocols often require frequent communication among nodes, leading to high overhead and communication costs. Deep reinforcement learning can help to reduce these costs by enabling nodes to make more informed routing decisions based on local information and past experiences.

Challenges

- **Computational Complexity:** DRL algorithms can be too computationally demanding and require high processing power and memory, which can be a challenge for resource-limited MWSN nodes.

- **DRL Training Demands:** Gathering large amounts of training data for DRL algorithms can be challenging in MWSNs due to communication and energy constraints. In addition, training may take a considerable amount of time, which could hinder the prompt deployment of the routing algorithm in real-world MWSNs."

- **Overfitting Risk:** DRL models can overfit to the training data, causing inadequate generalization to new, unseen data. This issue can be even more significant in mobile WSNs with highly variable network conditions, resulting in suboptimal routing decisions and poor performance.

- **Limited Transparency:** DRL algorithms are hard to interpret as they are often seen as "black boxes," leading to difficulties in understanding the decision-making process, which could make optimizing or debugging the system challenging.

Our proposed approaches involve training a DRL model to optimize routing decisions in OLSR networks by leveraging historical network data and performance metrics (Donta, Srirama, et al., 2023; Sah, Cengiz, et al., 2021). These approaches have the capability to optimize performance metrics including throughput, energy efficiency, and end-to-end delay in MWSNs. Through continuous learning and adaptation, these algorithms can therefore identify the most efficient routes that fulfil both network and application requirements. This thesis focuses on

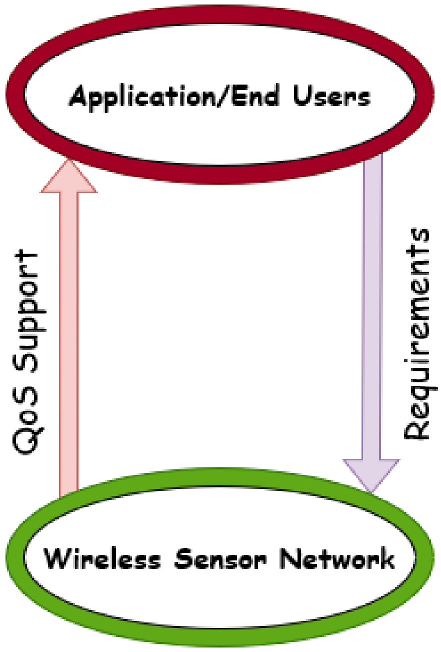


Figure 3.5: QoS functionality for WSNs (Asif et al., 2017)

researching and creating novel algorithms for routing and determining paths among sensor nodes. The proposed algorithms aim to achieve a balance in energy consumption during the path construction process.

Specifically, the following objectives are targeted.

- The DRL does not require any predetermined data sets to train the system unlike other ML approaches such as supervised or unsupervised.
- It provides the best decisions based on the trial and error methods by considering the exploitation or exploration algorithms with the previous optimal decisions (Xiao, S. Mao, and Tugnait, 2019).
- Unlike RL, DRL does not require additional space to maintain a Q-table and thus it is also not required to compute all the Q-values associated with each state.

3.6 Performance Parameters

In WSNs, several performance metrics are used for measuring the QoS of the network. In this section, definitions and formulae are targeted and used as performance metrics for the majority of the works in the literature. There is a strong relationship between QoS and application requirements as shown in Figure 3.5. The important QoS parameters are discussed below.

3.6.1 Connection Probability

The significance of connection probability lies in its direct impact on the overall functionality and performance of mobile sensor networks. Here are a few key aspects where connection probability is important:

- **Network Coverage**: Connection probability is closely related to network coverage. It indicates the extent to which the sensor nodes are capable of establishing connections with each other, forming a connected network. A higher connection probability implies a broader coverage area, ensuring that more nodes can communicate effectively. This is crucial in scenarios requiring comprehensive coverage, such as monitoring large-scale environments or tracking mobile entities.

- **Data Reliability and Quality**: A high connection probability enhances data transmission reliability and quality within the network. When nodes have a greater chance of connecting with each other, they can exchange information more consistently and accurately. This is vital in applications where the integrity and timeliness of data are critical, such as environmental monitoring, disaster response, or surveillance systems. Reliable data transmission contributes to better decision-making and improves the overall performance of the mobile sensor network.

- **Energy Efficiency:** Connection probability is closely tied to energy efficiency in mobile sensor networks. Nodes expend energy in establishing and maintaining connections. A higher connection probability means that nodes can establish connections more efficiently, reducing energy consumption. This becomes particularly significant in resource-constrained environments where sensor nodes operate on limited battery power. By maximizing the connection probability, energy can be conserved, prolonging the network's lifespan and reducing the need for frequent battery replacements or recharging.

- **Network Robustness**: Connection probability also influences the robustness of mobile sensor networks. In scenarios where nodes move or the environment undergoes changes, maintaining a reliable connection becomes challenging. However, a higher connection probability increases the chances of maintaining connectivity even in dynamic or unpredictable conditions. This ensures that the network remains operational and resilient, adapting to changes and providing valuable data and services.

3.6.2 End-to-End Delay

The end-to-end delay refers to the time taken for data to travel from the source sensor node to the destination node, encompassing all intermediate nodes and network components. It encompasses the transmission, propagation, and processing delays that occur during data transfer.

The significance of end-to-end delay lies in its impact on the real-time nature of mobile sensor network applications. For instance, in a healthcare system where sensor nodes monitor patients' vital signs, any delay in transmitting critical data could have severe consequences. Similarly, in environmental monitoring, timely data delivery is crucial for detecting and responding to events like natural disasters or pollutant outbreaks. A low end-to-end delay ensures that sensor data is delivered promptly, enabling timely decision-making and response. It facilitates real-time monitoring, analysis, and control, which is particularly important in applications requiring immediate action or feedback. By minimizing delays, mobile sensor networks can operate efficiently and provide accurate and up-to-date information to the users or the central processing system. Reducing the end-to-end delay in mobile sensor networks poses several challenges. These challenges include limited bandwidth, network congestion, variable channel conditions, and energy constraints of the sensor nodes. Researchers and engineers explore various techniques and protocols to mitigate these challenges and optimize network performance.

To address the end-to-end delay in mobile sensor networks, researchers focus on designing efficient routing protocols, optimizing data aggregation and fusion techniques, and implementing quality of service (QoS) mechanisms. These approaches aim to minimize delays by prioritizing critical data, reducing data redundancy, and dynamically adapting network parameters.

Eventually, the significance of end-to-end delay in mobile sensor networks lies in its impact on the real-time nature and effectiveness of data transmission. By minimizing delays, mobile sensor networks can deliver timely and accurate information, enabling prompt decision-making and action in applications such as healthcare, environmental monitoring, and surveillance. Efforts to reduce end-to-end delay involve designing efficient routing protocols and implementing optimization techniques to overcome the challenges posed by limited bandwidth, network congestion, and energy constraints. By addressing these challenges, mobile sensor networks can achieve improved performance and provide valuable services in a wide range of domains.

3.6.3 Routing Overhead

One of the primary reasons routing overhead is significant in mobile sensor networks is its direct influence on energy consumption. Sensors in these networks are often battery-powered, and conserving energy is a critical factor for prolonging the network lifetime. The additional control messages required for routing, such as route discovery, maintenance, and update messages, consume precious energy resources. High routing overhead can lead to increased energy drain, reducing the overall network lifespan and requiring frequent battery replacements or recharging.

Another important aspect affected by routing overhead is network bandwidth utilization. The transmission of routing control packets adds to the data traffic, competing for limited bandwidth resources in the network. As a result, excessive routing overhead can lead to congestion and decreased throughput, hampering the overall network performance. In mobile sensor networks, where data from various sensors needs to be relayed to a sink node or a central server, efficient utilization of network bandwidth is crucial for timely and reliable data delivery.

Furthermore, routing overhead impacts the scalability of mobile sensor networks. As the number of nodes increases, the control traffic required for routing also grows. This can lead to network congestion and increased contention for resources, limiting the network's ability to accommodate a larger number of sensors. Efficient routing protocols with minimal overhead are vital to ensure the scalability of mobile sensor networks, allowing for seamless expansion and accommodating a larger number of mobile sensors.

Additionally, routing overhead affects the network's resilience and adaptability to changing conditions. Mobile sensor networks operate in dynamic environments where sensors can move, join or leave the network, or experience link fluctuations due to obstacles or interference. Routing protocols must continually adapt to these changes, leading to additional control signaling and overhead. The ability to manage and minimize routing overhead while maintaining network connectivity and resilience is crucial to ensure the network's adaptability and robustness.

Eventually, the significance of routing overhead in mobile sensor networks lies in its impact on energy consumption, network bandwidth utilization, scalability, and network adaptability. Minimizing routing overhead is vital to extend the network's lifetime, improve data transmission efficiency, enhance scalability, and ensure the network can adapt to changing conditions. By employing efficient routing protocols and techniques that reduce unnecessary control signaling, researchers and network designers can mitigate the adverse effects of routing overhead and optimize the overall performance of mobile sensor networks.

3.6.4 Throughput

It is one of the significant parameters in the overall development of the network performance because it deals with the successful packets received with respect to the total number of packets sent. It can be used in various ways mentioned below.

- **Data Reliability**: In applications where real-time data is crucial, such as environmental monitoring or emergency response systems, high throughput ensures that data is transmitted promptly and reliably. Timely delivery of data enables quick decision-making and appropriate responses to events or conditions being monitored.

- **Energy Efficiency**: Sensor nodes in mobile sensor networks are typically resource-constrained, relying on limited battery power. Throughput optimization techniques aim to reduce energy consumption during data transmission, allowing sensor nodes to conserve their energy resources and prolong their operational lifetimes. Efficient data transmission helps in achieving energy-efficient network operation.

- **Network Scalability**: Mobile sensor networks often involve a large number of sensor nodes distributed over a wide area. Throughput plays a crucial role in maintaining network scalability. By ensuring high throughput, the network can accommodate a larger number of nodes and handle increased data traffic, allowing for the seamless expansion of the network without compromising its performance.
- **Quality of Service (QoS)**: Many applications in mobile sensor networks require specific (QoS) guarantees, such as minimum throughput requirements or delay constraints. High throughput is necessary to meet these (QoS) demands, ensuring that the network satisfies the application's performance criteria and meets the expectations of end-users.

To enhance throughput in mobile sensor networks, various techniques can be employed. These include optimizing routing protocols, employing data aggregation and compression techniques, utilizing adaptive transmission power control, and employing efficient medium access control (MAC) protocols. These techniques aim to minimize packet loss, reduce interference, and optimize the utilization of network resources to maximize throughput.

3.6.5 Energy Consumption

: Energy is a critical resource in mobile sensor networks as the sensor nodes are often powered by limited-capacity batteries or energy harvesting mechanisms. Since these networks are typically deployed in remote or inaccessible areas, recharging or replacing the batteries frequently is impractical or even impossible. Therefore, it becomes imperative to carefully manage and optimize the energy consumption of sensor nodes to prolong their operational lifespan and ensure the uninterrupted functioning of the network.

Optimizing energy consumption in mobile sensor networks offers several benefits. First and foremost, it enhances network longevity. By efficiently utilizing the available energy resources, the lifetime of the network can be significantly extended, ensuring that the sensor nodes remain operational for an extended period. This is particularly crucial in applications where continuous monitoring or data collection is required over an extended duration.

Moreover, minimizing energy consumption reduces the frequency of maintenance or node replacement, resulting in cost savings. Deploying and maintaining sensor networks often involve logistical challenges and financial costs. By conserving energy and reducing the need for frequent battery replacements, the overall maintenance and operational costs can be significantly reduced, making mobile sensor networks more economically viable. Additionally, energy-efficient mobile sensor networks contribute to environmental sustainability. The efficient use of energy resources reduces the number of batteries disposed of in the environment, minimizing the ecological footprint of these networks. This is particularly relevant as the deployment of mobile sensor networks continues to grow, and their impact on the environment becomes increasingly significant.

Efficient energy management in mobile sensor networks also enables improved network performance and reliability. By optimizing energy consumption, the network can allocate resources effectively, enhance data transmission rates, reduce latency, and improve overall network coverage. This, in turn, enhances the quality and accuracy of the collected data and ensures the reliability of the network for real-time applications.

Table 3.2: Comparison of various metrics and their trade-offs along with techniques to balance metrics

Metric	**Impact**	**Balance**	**Technique**
Average Energy Consumption	Lower energy consumption is desirable and Higher energy consumption leads to shorter network lifetime	Trade-off between energy consumption and network lifetime	Energy-efficient routing protocols, duty cycling, power management techniques, reinforcement learning for routing optimization (e.g. Q-learning, SARSA, actor-critic), neural networks for power control and energy prediction
Connection Probability	High connection probability leads to better network performance, Low connection probability may lead to data loss or delay	Trade-off between connection probability and network coverage	Optimal node placement, adaptive transmission range, multi-hop communication, reinforcement learning for node localization and link prediction (e.g. Q-learning with function approximation, deep reinforcement learning), neural networks for link scheduling.

Table 3.2: Comparison of various metrics and their trade-offs along with techniques to balance metrics

Metric	Impact	Balance	Technique
Routing Overhead	Lower routing overhead leads to better energy efficiency and network performance, Higher routing overhead leads to increased energy consumption and network congestion	Trade-off between routing overhead and routing efficiency	Hierarchical routing, clustering, adaptive routing, reinforcement learning for routing optimization (e.g. Q-learning, deep reinforcement learning), neural networks for traffic prediction and load balancing
Throughput	Higher throughput leads to better network performance, Lower throughput leads to network congestion and reduced data delivery	Trade-off between throughput and energy consumption	Multi-channel communication, adaptive transmission rate, congestion control, reinforcement learning for channel allocation and modulation selection (e.g. Q-learning, deep reinforcement learning), neural networks for traffic prediction and rate adaptation
End-to-End Delay	Lower end-to-end delay leads to faster data delivery, Higher end-to-end delay leads to increased energy consumption and reduced network performance	Trade-off between delay and energy consumption	Routing optimization, congestion control, quality of service (QoS) provisioning, reinforcement learning for delay-sensitive routing and scheduling (e.g. Q-learning, deep reinforcement learning), neural networks for traffic prediction and QoS-aware routing

Eventually, the significance of energy consumption in mobile sensor networks cannot be overstated. Efficient energy management not only prolongs the operational lifespan of sensor nodes but also reduces costs, improves network performance, and contributes to environmental sustainability. By employing energy-efficient strategies, researchers and practitioners can unlock the full potential of mobile sensor networks and enable their widespread adoption in diverse domains.

3.7 State-of-Art for Aggregation Methods

The research contribution of the researchers in the related field is analyzed for addressing challenges related to data transmission and data aggregation in Wireless Sensor Networks (WSNs) using machine learning and deep reinforcement learning techniques. The related works are explained in Table 3.3.

In (N. Kaur, D. R. Kaur, and D. R. Sharma, 2022), the author discusses the difficulty of data transmission in WSNs due to handling significant amounts of data. It emphasizes using data aggregation (DA) techniques to manage this issue. In (R. Kaur, Sandhu, and Sapra, 2020), the author explores the potential of machine learning approaches to improve WSN performance and efficiency, particularly in dealing with the restricted interoperability of sensors. In (Sudha, Suresh, and Nagesh, 2021), the authors propose an enhanced machine learning data aggregation model to overcome resource restrictions and efficiency problems in WSNs. In (H. Li, Wan, and H. He, 2020) and (F. Zhang and Q. Yang, 2020), the authors discuss deep reinforcement learning-based approaches for home energy management systems and energy trading in smart grids, respectively, to optimize power use and lower costs.

The (Z. Zhu, Ye, and L. Fu, 2020) addresses energy-efficient transmission in underwater acoustic communications using deep reinforcement learning. The (Y. Liu et al., 2020) presents a privacy-preserving data aggregation game in crowdsensing using deep reinforcement learning. The (Deepakraj and Raja, 2021) proposes a hybrid data aggregation algorithm to increase energy efficiency and extend the network lifetime in WSNs. The (Kokilavani, N. S. Kumar, and Narmadha, 2022) studies energy-efficient data aggregation techniques and their impact on various aspects of WSNs. The (Krishna and Vashishta, 2013) categorizes energy-efficient data aggregation algorithms based on their structure, search, and time-based techniques, with cluster-based protocols showing better energy efficiency and throughput rate performance. The detailed discussion is explained below in Table 3.3 regarding the objective, contribution, finding, and conclusion.

3.8 Importance of SoM and DRL in Routing

In summary, (Philip Paul Arunodhayam et al., 2023) presents a compelling case for using the Chimp Optimization Algorithm and Self-Organizing Map (SOM) to enhance energy efficiency and network performance. Their findings demonstrate significant improvements in network lifetime, packet delivery ratio (PDR), and packet loss ratio (PLR) compared to traditional routing protocols such as AODV, DSDV, and DSR. However, the integration of SOM with the Optimized Link State Routing (OLSR) protocol offers distinct advantages, particularly for Mobile Wireless Sensor Networks (MWSNs), which operate under different constraints and dynamics.

Table 3.3: Related works on data aggregation in WSN

Literature Study	
Reference	(N. Kaur, D. R. Kaur, and D. R. Sharma, 2022)
Objective	This paper discusses the difficulty of data transmission in Wireless Sensor Networks (WSNs) while handling significant amounts of data. To manage data aggregation and transmission effectively, the study focuses on Data Aggregation (DA) techniques.
Contribution	Examining numerous ML methods used in current DA research provides a thorough list of intelligent methods that have been applied to solve the data transmission problem.
Findings	The study discovers that when dealing with massive data quantities, data transmission in WSNs becomes difficult [1]. DA schemes have been suggested to solve this problem. However, maintaining QoS and security is still a challenge.
Conclusion	The evaluation and survey of the literature offer insightful information and establish the groundwork for future developments in WSN data transmission technology. WSNs may develop further and continue to handle the issues brought on by enormous data volumes in dynamic contexts by using ML methods.
Reference	(Sudha, Suresh, and Nagesh, 2021)
Objective	This research proposes a unique Machine Learning Data Aggregation Model (EML-DA) to solve the resource restrictions and efficiency issues in Wireless Sensor Networks (WSNs).
Contribution	The emphasis on robust data aggregation with ICA and hybrid CH selection with ANN highlights the significance of intelligent methods to maximize energy consumption and data processing efficiency in WSNs.
Findings	When data is aggregated at CH nodes, the computational effectiveness of ICA and its use of differential entropy help reduce duplicated data and enhance energy usage.
Conclusion	In conclusion, the EML-DA model's use of ANN and ICA illustrates the potential of intelligent data-driven solutions to provide reliable and effective data transmission in WSNs.

Literature Study	
Reference	(Y. Liu et al., 2020)
Objective	To develop the best tactics for the dynamic payment-PPL game, the article uses reinforcement learning approaches to determine the game's Nash equilibrium point, especially Q-learning and deep Q network (DQN).
Contribution	The suggestion of using reinforcement learning strategies like Q-learning and DQN to figure out the payment-PPL strategy shows how effective dynamic learning methods are at dealing with the unidentified payment-PPL model.
Findings	Even in situations where the payment-PPL model is unclear, the use of reinforcement learning techniques, particularly Q-learning and DQN, makes it easier to determine the best payment-PPL strategies.
Conclusion	DQN, in particular, outperforms standard Q-learning in terms of performance overall, resulting in better platform and participant utilities and improved data aggregation accuracy in the crowd sensing environment.
Reference	(Deepakraj and Raja, 2021)
Objective	To acquire data from many types of sensors for applications like agriculture and security, wireless sensor networks (WSNs) face an issue of redundancy. This research aims to solve this problem. To analyze data in real-time, consume less energy, and eliminate communication delays, the study suggests a Hybrid Data Aggregation Algorithm (HDAA). This would eventually increase the network lifespan.
Contribution	By eliminating redundant data in the collected data, data aggregation techniques strive to increase the accuracy and dependability of data processing in real-time applications.
Findings	An energy-efficient data aggregation procedure is produced by the clustering method, and cluster heads are chosen based on sensor rankings.
Conclusion	Overall, the proposed HDAA shows improved performance in cutting down on energy use and communication delays, resulting in a longer network lifetime.

Literature Study	
Reference	(Kokilavani, N. S. Kumar, and Narmadha, 2022)
Objective	To save costs and manage energy effectively, this research aims to emphasize the value of data aggregation in Wireless Sensor Networks (WSNs). To tackle difficult objectives like eliminating redundancy, improving energy efficiency, extending network lifetime, protecting privacy, and improving communication efficiency, the article suggests aggregation methods.
Contribution	Data aggregation's function in effective data forwarding to the base station, which results in increased energy efficiency, is highlighted by the focus on the process of condensing data from source nodes and eliminating duplicate information.
Findings	In the context of WSNs, the introduction of aggregation protocols solves several issues, including the elimination of redundancy, energy efficiency, network longevity, privacy protection, and communication efficiency.
Conclusion	In comparison to traditional algorithms, the suggested data aggregation algorithms show higher performance in privacy protection and communication efficiency, resulting in a longer network lifetime and reduced energy usage.
Reference	(Krishna and Vashishta, 2013)
Objective	This article's goal is to draw attention to the value of data aggregation protocols in Wireless Sensor Networks (WSNs) for effectively organizing data and sparing node energy to extend the network's lifespan.[10] According to their methods—structure-free, structure-based, distance-based, and time-based—energy-efficient data aggregation algorithms are categorized in this paper innovatively.
Contribution	A thorough study of the various protocol types is provided by suggesting a unique method for categorizing data aggregation methods based on their structure, search-based, and time-based techniques.
Findings	According to the research, cluster-based data aggregation protocols outperform structure-less, time-based, or search-based protocols in terms of energy efficiency and throughput rate.
Conclusion	The simulation findings show that in terms of energy consumption and throughput rate, cluster-based data aggregation protocols perform better than structure-free, time-based, and search-based protocols.

One of the primary advantages of integrating SOM with OLSR in MWSNs is the enhanced adaptability to dynamic network topologies. OLSR is designed to efficiently manage link state information and quickly disseminate routing updates throughout the network. By incorporating SOM, the protocol can leverage SOM's ability to cluster and organize data effectively. This integration allows OLSR to maintain optimal routing paths even as the network topology changes, ensuring consistent throughput and reducing the overhead associated with frequent route recalculations.

Energy efficiency is critical in MWSNs due to the limited battery life of sensor nodes. While the proposed algorithm used in the study focuses on optimizing routing for energy efficiency, integrating SOM with OLSR can further refine this optimization. SOM's clustering capabilities can help identify the most energy-efficient routes by grouping nodes based on their energy levels and connectivity patterns. This targeted approach to routing decisions can significantly reduce energy consumption across the network, prolonging the operational lifetime of sensor nodes and enhancing overall network sustainability.

The integration of SOM with OLSR can also lead to improved throughput and overall network performance. SOM's ability to process and analyze large datasets enables OLSR to make more informed routing decisions, optimizing the flow of data across the network. This can result in higher packet delivery ratios and lower packet loss ratios, as observed in the study with the Chimp Optimization Algorithm. In an MWSN context, these improvements are crucial for maintaining high-quality communication and data transmission, particularly in applications requiring real-time data monitoring and analysis.

The combination of SOM with OLSR enhances the scalability and robustness of MWSNs. OLSR's proactive nature, coupled with SOM's adaptive clustering, allows the network to scale efficiently, accommodating a growing number of nodes without compromising performance. Additionally, the robustness of SOM in handling noisy and incomplete data can help OLSR maintain reliable communication links even in challenging environments, such as those encountered in underwater or remote sensing applications.

Integrating SOM with OLSR in MWSNs offers several advantages over traditional routing protocols and optimization techniques. By enhancing adaptability to dynamic topologies, optimizing energy consumption, improving throughput, and ensuring scalability and robustness, this integration can significantly advance the state-of-the-art in MWSN routing. While the study by Philip Paul Arunodhayam et al. highlights the potential of combining optimization algorithms with SOM, further research and empirical validation are necessary to fully realize the benefits of integrating SOM with OLSR in diverse MWSN scenarios. This approach provides important insights for researchers and practitioners seeking to enhance the performance and sustainability of MWSNs.

In addition, (J. Tang, Mihailovic, and Aghvami, 2022) et al. demonstrate the efficacy of DRL in reducing maximum link utilization (MLU) and end-to-end delay through a Multi-Plane Routing (MPR) approach within a Dueling

Deep Q-Network (DDQN) framework. Building on this, integrating DRL with OLSR can enhance routing efficiency and network performance in various ways. The integration of Deep Reinforcement Learning (DRL) with the Optimized Link State Routing (OLSR) protocol presents significant opportunities for addressing the inherent challenges in traffic engineering (TE) within dynamic network environments.

OLSR is a proactive routing protocol that continuously updates and maintains network routes by periodically exchanging topology information. This ensures that routing decisions are always based on the most recent network state. By integrating DRL, OLSR can further enhance this proactive nature. DRL can predict future network states and traffic patterns, allowing OLSR to make more informed and anticipatory routing decisions, thereby reducing latency and improving throughput.

DRL's adaptive learning capabilities allow it to optimize routing decisions in real time, adapting to changing network conditions and traffic demands. In a dynamic network setting, where traffic load can vary significantly, DRL can help OLSR dynamically adjust routing paths to balance the load across the network. This can prevent bottlenecks and reduce MLU, as shown in the study where DRL-MPR achieved lower MLU compared to OSPF and standard MPR. Such dynamic traffic management is crucial for maintaining high performance in all-IP access networks.

Networks often experience link failures and varying link quality, especially in wireless environments. DRL can enhance OLSR's robustness by continuously learning from the network's operational history and adjusting routes accordingly. This capability ensures that the network can quickly adapt to failures and maintain optimal performance. Integrating DRL with OLSR can improve the network's fault tolerance, leading to fewer disruptions and more reliable communication.

As evidenced by the simulation results in (J. Tang, Mihailovic, and Aghvami, 2022) et al.'s study, DRL-MPR significantly outperforms traditional routing protocols in terms of MLU and end-to-end delay. To validate the integration of DRL with OLSR, similar empirical studies should be conducted. By comparing the performance of DRL-OLSR with traditional OLSR and other routing methodologies, we can quantify improvements in network efficiency, throughput, and latency. This empirical validation is critical for demonstrating the practical benefits of integrating DRL into proactive routing protocols like OLSR.

Chapter 4

SOM proposed Model

In this chapter, self-organizing maps (SOMs) are applied in the context of mobile sensor networks. SOMs are unsupervised learning algorithms that have gained significant attention for their ability to organize and represent complex data patterns. Within mobile sensor networks, SOMs offer a promising approach to tackling the challenges associated with resource allocation, data transmission, and network performance optimization.

The introduction section of this chapter provides an overview of the fundamental concepts of self-organizing maps and their underlying principles. SOMs enable sensors to be explored in mobile networks to create a topological representation of their environment and organize themselves based on spatial relationships and data patterns. By leveraging this self-organizing capability, the sensors can enhance data aggregation, improve decision-making processes, and achieve more efficient resource allocation.

Furthermore, the potential applications are highlighted of SOMs in mobile sensor networks. The versatility of SOMs makes them suitable for a wide range of use cases, including environmental monitoring, surveillance systems, and disaster management. By employing SOMs in these applications, the ability can be leveraged to adapt to changing network conditions, facilitate collaborative decision-making among sensors, and optimize data transmission for improved network performance.

Lastly, the results and discussions section presents the findings obtained through implementing SOMs in our mobile sensor network framework. The experimental setup is discussed, and data collection and performance metrics are used to evaluate the effectiveness of SOMs. The results showcase the impact of SOMs on network throughput, data aggregation efficiency, and overall network performance. Additionally, the results are analyzed and interpreted the obtained results, providing insights into the strengths, limitations, and potential areas of improvement for the SOM-based approach in mobile sensor networks.

This chapter aims to shed light on the capabilities and potential of self-organizing maps in addressing the challenges mobile sensor networks face. By harnessing the power of unsupervised learning, SOMs offer a valuable tool for optimizing resource allocation, improving data transmission efficiency, and enhancing overall network performance. The following sections will delve deeper into the applications, implementation, and results of employing self-organizing maps in the mobile sensor network.

4.1 Introduction

Self-organizing maps (SOMs) are a type of unsupervised artificial neural network (ANN) that was introduced by Finnish scientist Teuvo Kohonen in 1982. These networks are inspired by the structure and function of the human brain's visual cortex and have been widely used for clustering and visualization tasks. The main goal of SOM is to reduce the dimensionality of high-dimensional data to a low-dimensional representation while preserving the topological structure of the data.

During training, the neurons' weights in a two-dimensional grid are iteratively adjusted to match the input data. A competitive learning rule compares each input vector to the weight vectors of all the neurons, and the neuron with the closest weight vector is selected as the winner. The winning neuron and its neighboring neurons are then updated based on a Gaussian function that decreases with distance from the winner neuron. SOMs have many advantages, including handling large datasets with high-dimensional input data, computational efficiency, and unsupervised learning. They can also adapt to input data distribution changes, making them suitable for online learning applications.

SOMs have been successfully applied in many fields, including image processing, speech recognition, bioinformatics, and finance. In image processing, SOMs have been used for tasks such as image segmentation and object recognition, while in speech recognition, they have been used for phoneme recognition and speaker identification. In bioinformatics, SOMs have been used for gene expression analysis and protein structure prediction. SOMs have been used in finance for stock market prediction and credit risk assessment.

They work by taking a set of high-dimensional input vectors and mapping them onto a lower-dimensional grid of output nodes or neurons. Each neuron is associated with a weight vector randomly initialized to a value close to the input vectors. During the learning process, the SOM iteratively adjusts the weight vectors based on the similarity between the input vectors and the weight vectors of the neighboring neurons, Figure 4.1 shows SOM process in action.

4.1.1 Applications of SOM in MWSNs

Self-organizing maps (SOMs) have found numerous applications in sensor networks due to their ability to perform data analysis and visualization in a distributed manner. Here are some of the applications of SOMs in sensor networks:

- **Anomaly detection** SOMs can be used for anomaly detection in sensor networks. They can be trained on normal operating conditions of the sensor

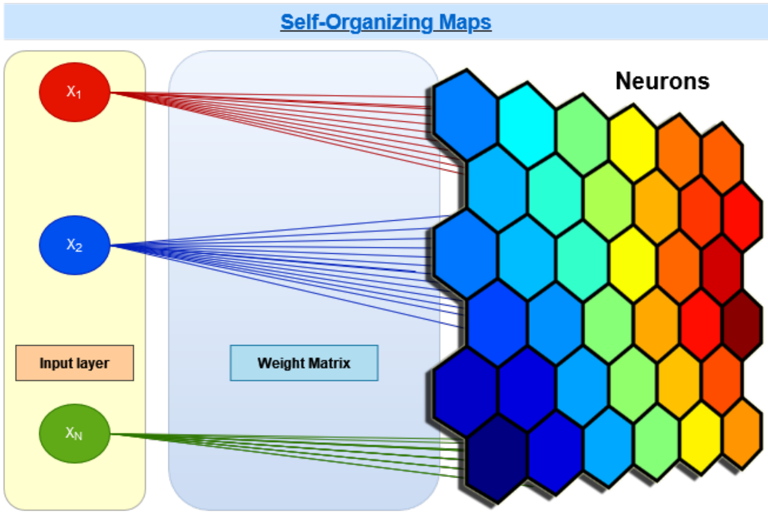


Figure 4.1: SOM Process

network, and any deviation from the learned patterns can be flagged as an anomaly. This approach can detect faults or intrusions in the sensor network.

- **Data compression** SOMs can be used for data compression in sensor networks. The weight vectors of the neurons in the output layer can be used to represent the input data in a lower-dimensional space. This can be useful for reducing the size of the data transmitted by the sensors, which in turn saves energy and bandwidth. Mobile sensor networks are characterized by small and lightweight sensors connected wirelessly to a central unit, such as a mobile phone or base station.

  These networks generate a large amount of data that needs to be transmitted over wireless channels with limited bandwidth and power resources. SOMs can be used to reduce the amount of data transmitted by the sensors while preserving the important features. Here are some ways SOMs can be used for data compression in mobile sensor networks.

  - **Dimensionality Reduction:** SOMs can be used to reduce the dimensionality of the sensor data. In this approach, the SOM is trained on the sensor data, and the weight vectors of the output neurons are used to represent the input data in a lower-dimensional space. The size of the SOM output layer determines the dimensionality of the compressed data. The compressed data can be transmitted to the central unit, where it can be decompressed using the inverse mapping of the SOM.
  - **Quantization:** SOMs can be used for vector quantization, a lossy data compression technique that maps high-dimensional vectors to a set of discrete codebook vectors. In this approach, the SOM is trained on the

sensor data, and the weight vectors of the output neurons are used as codebook vectors. The sensor data is then quantized by mapping each input vector to the nearest codebook vector. The quantized data can be transmitted to the central unit, where it can be decompressed by replacing each codebook vector with the corresponding weight vector of the SOM.

  - **Hybrid Approaches:** Hybrid approaches that combine SOMs with other data compression techniques have also been proposed. For example, a hybrid approach that combines SOMs with wavelet transforms has been proposed for image compression in mobile sensor networks. In this approach, the image is first transformed using a wavelet transform, and the resulting coefficients are then compressed using a SOM-based vector quantization technique.

    Using SOMs for data compression in mobile sensor networks have several advantages, including reduced data transmission and storage requirements, reduced power consumption, and improved data analysis and visualization.Additionally, SOMs are well-suited for mobile sensor networks, characterized by limited processing and storage capabilities, dynamic network topologies, and varying environmental conditions.

- **Visualization** SOMs can be used for visualizing sensor data in a distributed manner. Each sensor can train a SOM on its local data, and the resulting SOMs can be merged to form a global SOM that represents the entire sensor network. The global SOM can be used to visualize the similarities and differences between the sensor data, which can be useful for monitoring the sensor network and identifying patterns.

- **Clustering** SOMs can be used for clustering sensor data. The SOM can be trained on the sensor data, and the resulting clusters can be used for data aggregation, anomaly detection, or decision-making tasks. Clustering in mobile sensor networks is challenging due to the dynamic nature of the network topology and the limited resources of the sensors. SOMs can overcome these challenges and perform clustering in a distributed manner. Here's how it can be done:

  - **Initialization:** The SOM is initialized by randomly selecting a set of neurons. Each neuron represents a cluster, and its weight vector is set to the centroid of the assigned data points.
  - **Sensor data acquisition:** Each sensor acquires data from its surroundings and assigns it to the closest SOM neuron. This is done using a competitive learning rule similar to the one used in the unsupervised training of SOMs.
  - **SOM update:** The weight vectors of the neurons are updated based on the input data assigned to them. The update uses a learning rate

  that decreases with time and a neighborhood function that decreases with the distance between the neuron and the winning neuron.

  - **Cluster formation:** After a certain number of iterations, the SOM stabilizes, and the neurons' weight vectors represent the cluster centroids. The sensors are then assigned to the closest neuron in the SOM, and the resulting clusters represent the groups of sensors with similar data.

    SOM-based clustering in mobile sensor networks has several advantages. First, it can be performed in a distributed manner, which reduces the communication overhead and the energy consumption of the sensors. Second, it can adapt to changes in the network topology and the data distribution, which makes it suitable for dynamic sensor networks. Finally, it can handle high-dimensional data and provide a low-dimensional representation of the data, which can be helpful for data visualization and compression.

    SOM-based clustering in mobile sensor networks has been applied in various domains, including environmental monitoring, disaster management, and healthcare, For e.g., SOM-based clustering can be used in environmental monitoring to group sensors that measure.

    similar environmental parameters, such as temperature and humidity. In disaster management, SOM-based clustering can identify areas with high concentrations of pollutants or detect the presence of hazardous materials. In healthcare, SOM-based clustering can be used to group sensors that measure physiological parameters, such as heart rate and blood pressure, to monitor the health of patients.

- **Predictive maintenance** SOMs can be used for predictive maintenance in sensor networks. They can be trained on historical sensor data, and the resulting SOM can be used to predict when a sensor or a component of the sensor network will likely fail. This approach can help prevent unplanned downtime and reduce maintenance costs. Mobile sensor networks consist of mobile nodes equipped with sensors that can move around and gather data. These networks can be used for monitoring various systems, such as machines in a manufacturing plant, vehicles on a highway, or infrastructure in a city.

  To apply SOMs for predictive maintenance in mobile sensor networks, the SOM is first trained on historical sensor data to learn the typical patterns of the system. This training process involves inputting the historical sensor data into the SOM and adjusting the neurons' weights to represent the patterns in the data. Once the SOM is trained, it can be used to predict when a system component is likely to fail. When the mobile sensor network operates, the sensors gather real-time data from the system.

  This data is then input into the SOM, which compares the current patterns to the historical patterns learned during training. If the current patterns deviate significantly from the learned patterns, the SOM can flag the deviation as a potential fault or failure. The maintenance team can investigate the flagged component to prevent unplanned downtime and reduce maintenance costs.

By using SOMs for predictive maintenance in mobile sensor networks, the maintenance team can take a proactive approach to maintenance rather than a reactive one. This can help minimize the system's downtime, increase the system's reliability, and reduce maintenance costs. Furthermore, since SOMs are trained on the sensor data in a distributed manner, they can be used for predictive maintenance in large-scale mobile sensor networks with many sensors and nodes.

- **Energy-efficient routing** SOMs can be used for energy-efficient routing in sensor networks. The SOM can be used to find the optimal path for transmitting data from the sensors to the sink node while minimizing energy consumption. This approach can help extend the lifetime of the sensor network. This application uses the SOM to find the optimal path for transmitting data from the sensors to the sink node while minimizing energy consumption (Xu et al., 2019) (Ahmed Elsmany et al., 2019).

4.1.2 The key features of SOMs

- One of the standout attributes of SOMs is their ability to perform unsupervised learning. Unlike supervised learning algorithms that require labeled data, SOMs can uncover patterns and relationships within unlabeled data. This makes them particularly useful for tasks such as clustering, where data points with similar characteristics are grouped. By identifying clusters within the data, SOMs enable exploratory data analysis and can provide valuable insights into complex datasets.

- Another crucial feature of SOMs is their capability to preserve the topological structure of the input data. When trained, a SOM organizes a grid of neurons in the output space. This grid mimics the input data's topology, meaning that similar input vectors are mapped to adjacent neurons on the grid. This preserves the relationships and similarities in the original data, allowing for efficient visualization and analysis of data patterns. Observing the neurons' proximity on the grid can infer the proximity and similarity of the corresponding input vectors.

- SOMs also excel at dimensionality reduction, which is mapping high-dimensional data onto a lower-dimensional space. High-dimensional datasets can be challenging to visualize and comprehend, but SOMs offer an effective solution. By projecting the input vectors onto a lower-dimensional output grid, SOMs enable the representation of complex data in a more manageable form. This dimensionality reduction simplifies data exploration, enabling researchers and analysts to gain valuable insights from large, intricate datasets.

- In addition to their robustness, SOMs are capable of handling noisy and incomplete data. Real-world datasets are often imperfect, with missing values or errors. However, SOMs possess inherent robustness, allowing them to process and analyze such data effectively. By capturing the underlying patterns and relationships within the data, SOMs can provide reliable results even in the presence of noise or incomplete information. This attribute makes SOMs well-suited for real-world applications where data quality may vary.

The key features of SOMs make them a powerful tool for data analysis and visualization. Their ability to perform unsupervised learning enables the exploration of unlabeled data, leading to valuable insights and discoveries. The preservation of topological structures facilitates understanding data relationships, while dimensionality reduction aids in the visualization and comprehension of complex datasets.

Furthermore, the robustness of SOMs allows them to handle noisy and incomplete data, enhancing their applicability in real-world scenarios. By leveraging these features, researchers and analysts can harness the power of SOMs to uncover hidden patterns, gain insights, and make informed decisions in various domains.

4.2 The SOM-based Routing Algorithm

- First, the SOM is trained on the sensor data to learn the topological structure of the network. Each node in the SOM represents a region in the network where sensors with similar data are located. The nodes in the SOM are connected to their neighboring nodes, forming a two-dimensional grid.
- When a sensor node wants to transmit data to the sink node, it calculates the Euclidean distance between its data and the weight vectors of the nodes in the SOM.
- The node in the SOM with the closest weight vector to the sensor data is selected as the destination node. Using a multi-hop approach, The sensor node transmits its data to the destination node. Each hop is directed toward the neighboring nodes of the destination node, which leads to the sink node.

The SOM-based routing algorithm has several advantages over traditional routing algorithms. It can adapt to network topology changes, making it is suitable for mobile sensor networks. It can also reduce the amount of data transmitted by the sensors, saving energy and prolonging the network's lifetime.

In addition, the SOM-based routing algorithm can be combined with other energy-efficient techniques, such as duty cycling and data aggregation, to reduce energy consumption further.

Duty cycling involves turning off the sensors during idle periods, while data aggregation involves combining multiple data packets into a single packet to reduce the number of transmissions. Eventually, SOMs are a powerful tool for clustering, visualization, and data compression tasks. They are based on a competitive learning rule and a Gaussian neighborhood function, which allows them to preserve the topological structure of the input data. SOMs are computationally efficient, capable of handling large datasets, and suitable for online learning applications. They have been applied in various fields, including image processing, speech recognition, bioinformatics, and finance, and are likely to continue to be essential tool.

4.3 Methodology used in SOM in MWSN

In the proposed work, SOMs are used to optimize sensor networks by helping to identify patterns and relationships in nodes connectivity. Training a SOM using the connectivity data of a sensor network can provide valuable insights that can contribute to optimizing the OLSR routing protocol in the network. Initially, the SOM is trained using the connectivity data of the sensor network. This data typically includes information about which nodes are connected and the strength of their connections. The SOM algorithm learns to organize and map this connectivity data onto a lower-dimensional grid.

SOM-OLSR routing algorithm is propoed, which is based on SOM algorithm. SOM is used to optimize sensor networks by helping to identify patterns and relationships in node connectivity. It smoothes and scales the execution of the requests, and it maps high-dimensional information onto a low-dimensional space as a result of which the network trains itself to identify patterns and relationships in the network connectivity and creates a map that represents these patterns in a more manageable way.

An integration of Self-Organizing Map (SOM) with the Optimized Link State Routing (OLSR) protocol becomes a powerful solution in terms of improving performance of Mobile Wireless Sensor Networks (MWSNs) as well as the whole sequence is deployed in partial as mobile agent nodes in monitoring scenarios. The learning and adaptability of SOM to the network topology based on sensor data and communication patterns enables a deeper and more valuable understanding of the environment's dynamic nature. In a continuous way, SOM analyzes the collected data and assists OLSR in making informed routing decisions, optimizing the paths based on node movement, energy levels, and data transmission needs. This dynamic routing optimization improves the efficiency of data transmission by in addition to offering load balancing so that network congestion is eliminated and communication in the monitoring area is assured. Moreover, SOM empowers the network with fault detection and self-healing functionalities through network monitoring for anomalies, which consequently triggers corrective measures like traffic rerouting or node reconfiguration to sustain data delivery and network integrity despite challenges.

Noteworthy, the amalgamation of SOM with OLSR increases the self-management of the network, thus eliminating human intervention and increasing the scalability of multi-wireless sensor networks. By recursively learning and adapting to environmental changes, the SOM network is capable of responding to node failure, changes in the environment, and communication patterns without much human intervention. That autonomy feature stands out in the area of monitoring applications where a remote and large-scale deployment of sensor nodes is a common practice. The integration of SOM and OLSR brings self-organization and adaptation to MWSNs, which helps the networks to perform efficiently and reliably in the dynamic environment of monitoring applications, thus leading to the improvement in the effectiveness of data collection and decision-making processes.

Let $\mathbf{w}_{ij}$ be the weight of neuron i in the SOM, where i ranges from 1 to the dimension of the input space, and let x_j be the jth element of the input vector $\mathbf{x}$. Let $\mathbf{m}_i$ be the location of neuron i in the SOM.

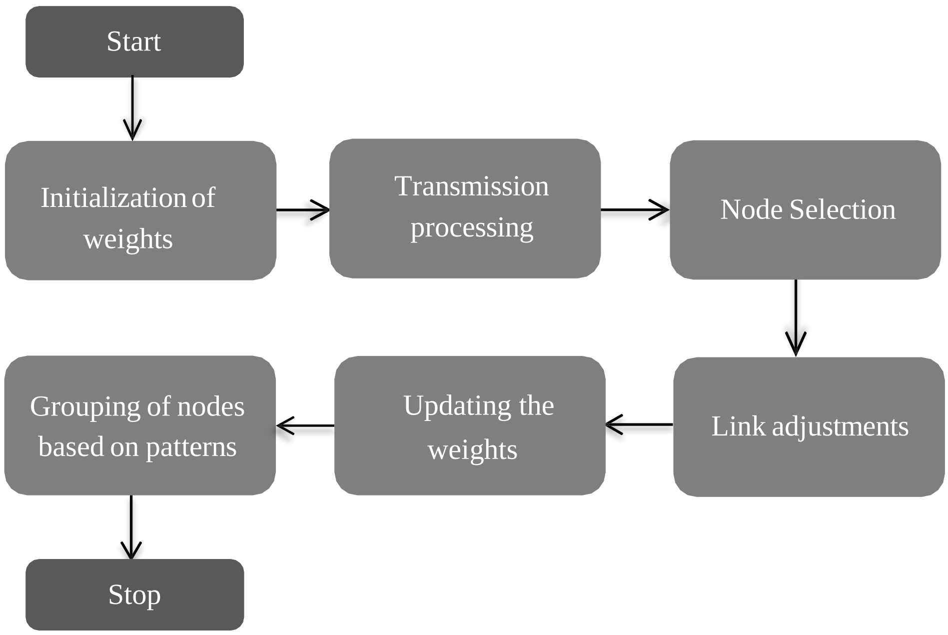


Figure 4.2: SOM in mobile sensor networks

The SOM-based routing algorithm works as follows:

- The SOM-OLSR is trained on the sensor data to learn the topological structure of the network. Each node in the SOM represents a region in the network where sensors with similar data are located. The nodes in the SOM are connected to their neighbouring nodes, forming a two-dimensional grid.

Algorithm 1 SOM-OLSR Algorithm

Initialization of parameters
Deployment of Nodes in Network Area (N,R)
Broadcast Link State Packet (LSP) to all the Neighboring Nodes
Cost Function Estimation for connection
Initialization:
Normalize weight matrix:
initialize random weights()
Calculate
$W_{Nor}(i,j) = \frac{(W_{i,j} - min_{val})}{max_{val} - min_{val}}$
for i to N **do**
  $d_i = \sqrt{\sum((x_j - w_i)^2}$
  Finding the closest neuron
  $closest_{neuron} = argmin_{(distances)}$
  return $closest_{neuron}$
  Update Weights For Selected Node
  $W_{ij}(t+1) = W_{ij}(t) + \eta(t) \times h_{ij}(X_j - W_{ij}(t))$
  Adjust Weights For Neighboring Node
  $h_{ji}(t) = \exp\left(\frac{(c_i(t) - c_j(t))^2}{2\sigma^2(t)}\right)$
end for

Let's assume that the distance between neuron i and neuron j is denoted as $d(i,j)$. The cost function *C(d)* is used to calculate the cost based on the distance. Cost function is formulated as

$$C(d) = k * d \tag{4.1}$$

where k is a constant representing the cost per unit distance. Once the cost function is defined, the cost of the connection between neuron i and neuron j can be estimated as

$$C(i, j) = C(d(i, j)) \tag{4.2}$$

By evaluating the distance between the neurons and applying the cost function, a cost value is obtained for each connection in the SOM.

SOMs and OLSR are used in energy-efficient routing in sensor networks through a two-phase approach, as the following:

- In the first phase of SOM training, the SOM algorithm is employed to learn. Let's assume, a sensor node denoted by $N = N_1, N_2, ..., N_i$, where each sensor node N_i has associated features x_i, e.g. distance, coordinates, data rate, etc. The features of each sensor node are represented as a vector: $x_i = [x_{1i}, x_{2i}, ..., x_{mi}]$, where m is the number of features.

SOM features can be trained by presenting the feature vectors x_i to the network. The SOM consists of a set of neurons organized in a grid. Each neuron j in the SOM is represented by a weight vector $w_j = [w_{1j}, w_{2j}, ..., w_{mj}]$, where m is the number of features. Initially, the weights are randomly assigned. The SOM adjusts its weight vectors during training based on a learning algorithm.

The transmission dimensions can be represented as a weight matrix W, where each element $W_{i,j}$ represents the weight (transmission dimension) between node i and node j.

- Calculate the minimum value (min_{val}) and maximum value (max_{val}) from the weight matrix W, where W is an $m \times n$ matrix representing the transmission dimensions between nodes in the MWSN. Then normalize the values of the weight matrix W using a common normalization technique as

$$W_{Nor}(i, j) = \frac{(W_{i,j} - min_{val})}{(max_{val} - min_{val})} \tag{4.3}$$

  After this step, the normalized values in W_{Nor} will be in the range $[0, 1]$. This ensures that all weights are proportionally adjusted based on their relative differences.

- Each node in the SOM has a weight vector W of the same dimension as the input data vector. Calculate the Euclidean distance between the input data vector x and the weight vector W of each node in the SOM. Let W_{ij} be the weight of neuron i and j dimension of the input space, and let x_j be the j^{th} element of the input vector x. The distance between the input vector x and the weight vector w_i is given by:

$$d_i = \sqrt{\sum_j (x_j - w_i)^2} \tag{4.4}$$

- Identify the node with the smallest Euclidean distance as the best matching unit (BMU). Compare the calculated Euclidean distances for each neuron in the SOM and find the neuron that has the smallest distance to the input data vector x. This neuron is considered as BMU or the neuron that is closest to the input data vector x.

- Update the weights of neighboring nodes based on their distance from the selected node.

$$W_{ij}(t+1) = W_{ij}(t) + \eta(t) \times h_{ij}(X_j - W_{ij}(t)) \tag{4.5}$$

where t is the iteration number, $\eta(t)$ is the learning rate at iteration t, and $(X_j - W_{ij}(t))$ is the error between the input vector and the old weight vector. The step size of weight updates is determined by the learning rate $\eta(t)$ which is usually set to a large value at the beginning and gradually decreased during the course of the algorithm to ensure that it converges. The neighbourhood function $h_{ij}(t)$ is a Gaussian function that decreases with distance from the winning neuron.

$$h_{ji}(t) = \exp\left(\frac{(c_i(t) - c_j(t))^2}{2\sigma^2(t)}\right) \tag{4.6}$$

where c_i and c_j are the locations of neurons i and j, and $\sigma(t)$ is the neighborhood radius at iteration t. The neighboring radius refers to the maximum distance within which nodes or devices can communicate directly with each other. This process is depicted in Figure 4.2 and the associated proposed algorithm is presented in Algorithm 1.

- Repeat steps 3 and 4 iteratively for each transmission in the network to achieve similar data patterns.

- Analyze the resulting node weights to identify clusters of similar transmission patterns, thereby reducing redundancy in the network.

  During Training, the SOM adapt the weights of the neurons in the output map based on the input data using a learning algorithm as discussed below step wise.

- The second phase performs energy-efficient routing for SOM process. Once the SOM is trained, it is used for energy-efficient routing in the sensor network. SOM can be utilized to find the optimal path for data transmission from a source sensor node to a sink node. Each neuron j in the SOM represents a region in the network. Each region is associated with a set of sensor nodes within its vicinity. Neurons in the SOM can be identified for a given source and sink node. The path between the neurons corresponding to the source and sink nodes can also be identified by traversing the connections in the SOM grid. This path on the SOM grid represents the optimal path in the sensor network for data transmission from the source to the sink by considering energy efficiency.

4.4 Results & Discussion

The Result and Discussion chapter of a thesis serves as a critical component for presenting and analyzing the findings obtained through the research process. It allows for an in-depth exploration and interpretation of the data collected, providing an opportunity to address the research questions, evaluate the hypotheses, and discuss the implications of the study. The MWSN network is simulated using the MATLAB environment, with the simulation parameters listed in Table 4.1 and Table 4.2. The network comprises multiple nodes that are randomly distributed as depicted in Figure 4.3.

Table 4.1: Simulation Parameters.

Parameter	Value
Nodes	100
Network Length	5000 m
Network Width	5000 m
Bite per Sec	1000
Minimum Communication Probability	0.2
Minimum Radius	10 m
Maximum Radius	60 m
Coverage Distance	150 m
Total Iterations	50
RNN States	8
Maximum Steps per Episode(RNN)	50
Epsilon	0.9
Epsilon Decay	0.9

In this chapter, the research outcomes are thoroughly examined, compared with existing literature, and contextualized within the broader research field. By delving into the results and engaging in thoughtful discussion, researchers can offer valuable insights, draw conclusions, and contribute to the existing body of knowledge. The proposed implementation is done using a MATLAB environment.

4.4.1 Result and Discussion using SOM

TheFigure 4.4 shows the connection probability which should be high for effective communication in mobile sensor networks. Connection probability plays a critical role in determining the performance and effectiveness of mobile sensor networks. It affects network coverage, data reliability, energy efficiency, and network robustness. By focusing on improving and maintaining a high connection

Table 4.2: Simulation Parameters.

Parameter	Description	Default Value
Node Speed	Rate of sensor node movement	1 m/s
Node Direction	Initial direction of node movement	Randomly determined
Pause Time	Duration for which a node remains stationary	0 seconds
Random Waypoint Model		
Maximum Speed	Maximum speed of nodes in the model	1 m/s
Minimum Speed	Minimum speed of nodes in the model	0 m/s
Waypoint Pause Time	Duration for node pauses at a waypoint	0 seconds
Gauss-Markov Model		
Mean Speed	Mean speed of nodes in the model	1 m/s
Standard Deviation	Standard deviation of node speed	0.1 m/s
Correlation Time	Time constant for speed correlation	100 seconds
Random Walk Model		
Step Length	Length of each step in the model	1 meter
Step Time	Duration of each step in the model	1 second

probability, researchers and practitioners can enhance the functionality and practicality of mobile sensor networks, enabling them to fulfill their intended purposes effectively in various applications. It represents the likelihood or probability of successful communication or connectivity between sensor nodes within the network. The higher the connection probability, the greater the chances of establishing reliable and efficient communication links among the nodes.

TheFigure 4.5 shows the end-to-end delay in ms for the mobile sensor network. The end delay must be low as much as possible for low bit error rates. The end-to-end delay is a crucial metric in mobile sensor networks as it plays a significant role in determining the efficiency and effectiveness of data transmission. In mobile sensor networks, which consist of numerous sensors connected wirelessly, timely and reliable data delivery is essential for various applications such as environmental monitoring, healthcare systems, and surveillance.

To have a high packet receiving rate, the end delay must be as minimal as possible. There is a significant impact on end delay when the chance of connection declines, as in the case of SOM can be observed by the fact that end delay is

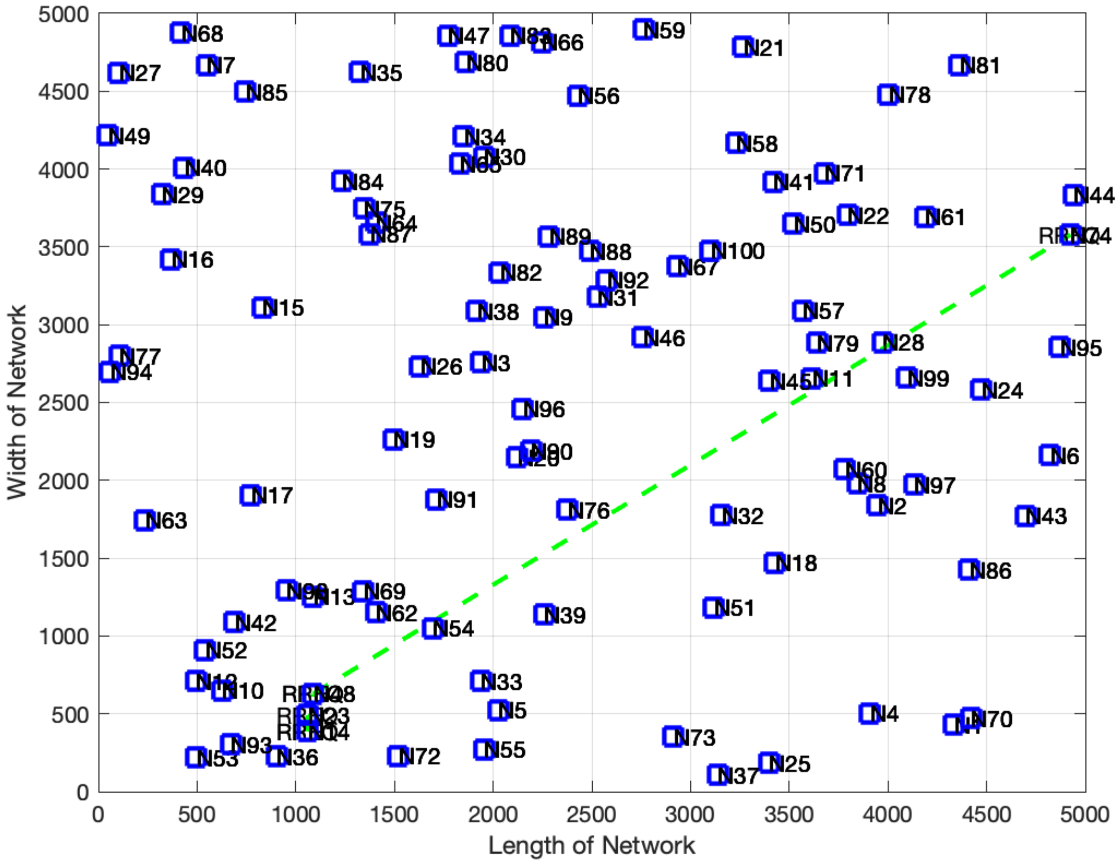


Figure 4.3: MWSN network simulation setup.

decreasing and can be further reduced by training the network with more iterations with dense layers.

Figure 4.6shows the routing overhead also improved, which should be controlled to achieve high control of packets with high data rates with high mobility. Routing overhead indicates that a large number of packets are at the maintenance level, increasing the likelihood of failures and significant packet dropouts. Additionally, the network's throughput, which must be high also improved in the case of neural networks and indicates high successful packet deliveries at the receiver side, and can be observed.

Routing overhead plays a crucial role in mobile sensor networks, and its significance stems from the impact it has on network performance and resource utilization. In mobile sensor networks, where sensors are deployed in a dynamic environment and can move autonomously, efficient routing is essential for reliable and timely data delivery. However, routing overhead refers to the additional control and signaling information that is necessary for establishing and maintaining routing paths, but adds to the overall communication load and resource consumption.

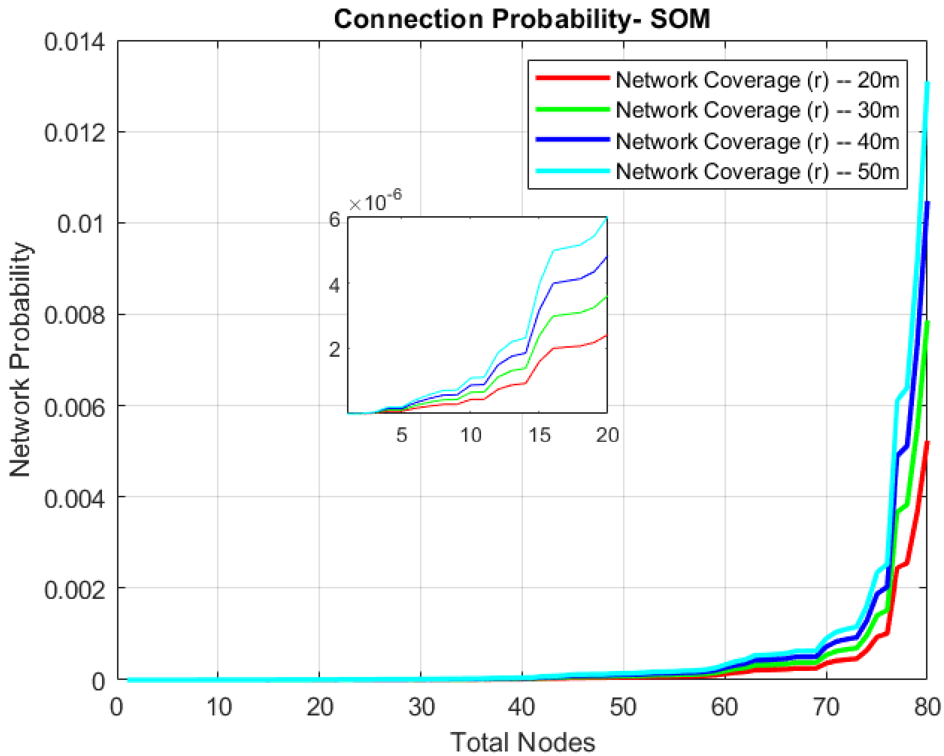


Figure 4.4: Connection Probability using SOM

Figure 4.7 shows the throughput in the proposed evaluation and shows that if throughput becomes less then there will be high chances of network degradation which is not the desired output of the proposed work. The significance of throughput in mobile sensor networks is of utmost importance as it directly affects the overall performance and efficiency of the network.

Throughput refers to the amount of data that can be successfully transmitted within a given time period in a network. It is a critical metric that determines the network's capacity to handle data traffic effectively. In mobile sensor networks, the ability to achieve high throughput is vital for ensuring timely and reliable transmission of data from the sensor nodes to the intended destination, such as a base station or a central server.

The Figure 4.8 shows the energy consumption of the network through the proposed evaluation. As it can be seen from the evaluation that energy consumption is also improved and must be low for the successful use of resources in the mobile sensor network. If the energy consumption is high then there can be more failure of nodes and nodes may dead and halt the operation of execution of requests in the network.

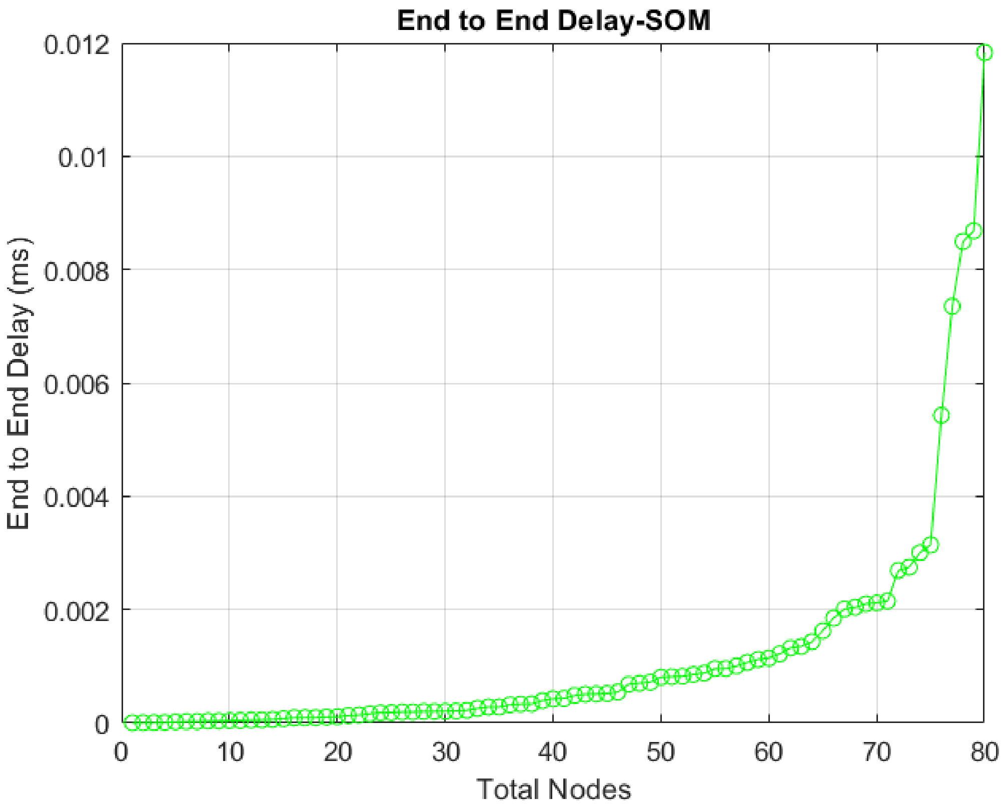


Figure 4.5: End to End Delay using SOM

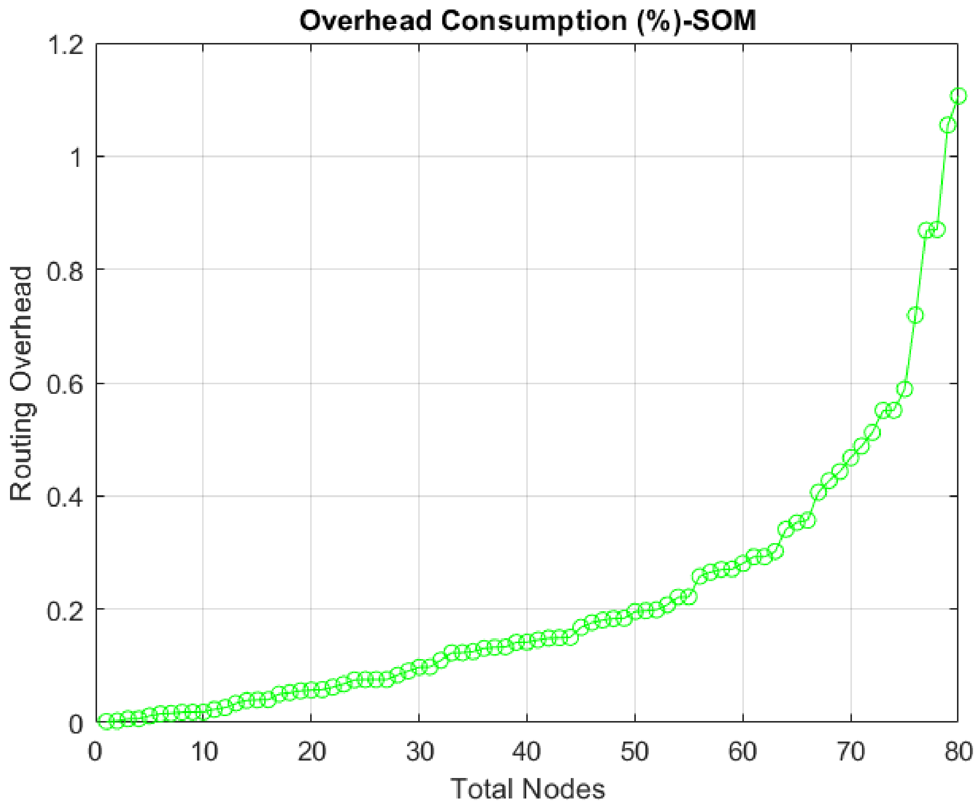


Figure 4.6: Overhead Consumption using SOM

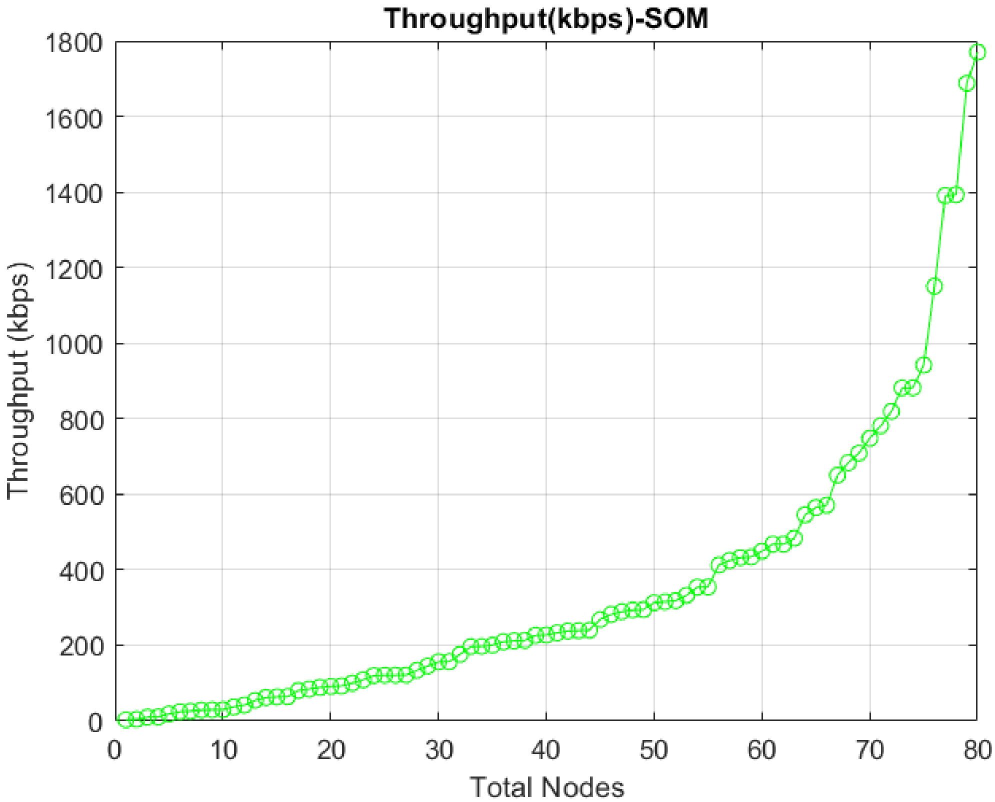


Figure 4.7: Throughput using SOM

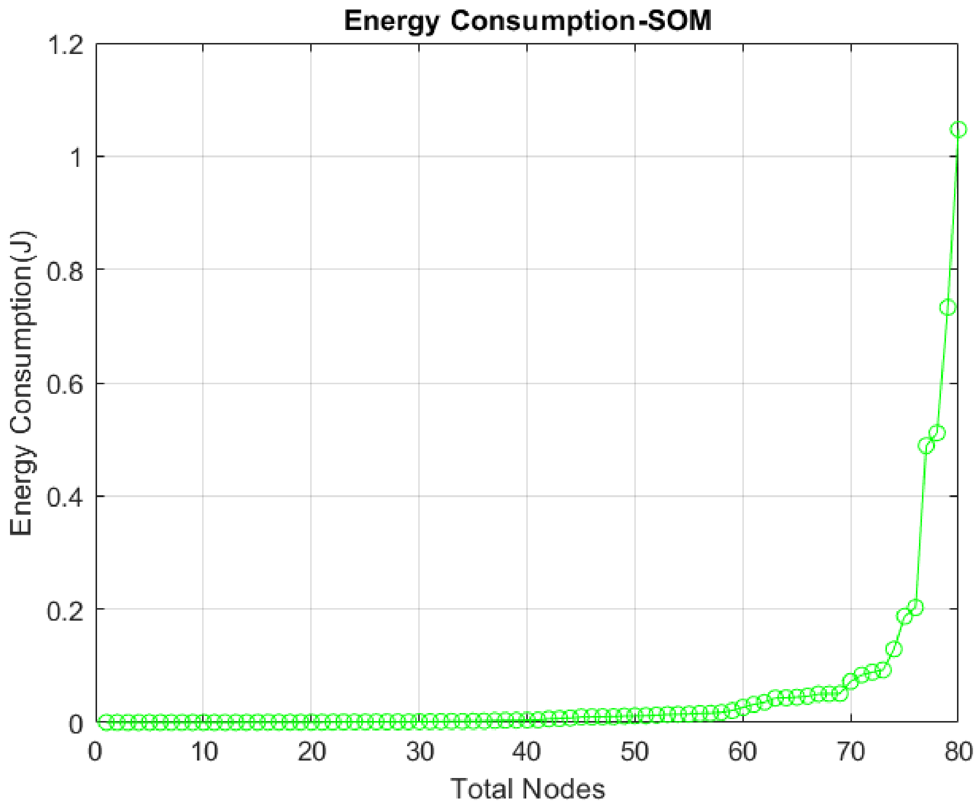


Figure 4.8: Energy Consumption using SOM

Chapter 5

DRL Proposed Model

In this chapter, the realm of DRL is discussed and explored its components, and the results achieved through its implementation.

The introduction sets the stage for understanding the significance of DRL in addressing various challenges in different domains. The growing interest is highlighted in leveraging DRL techniques to tackle complex problems that involve sequential decision-making. The ability of DRL to learn from interactions with the environment and to optimize decision-making processes based on rewards has resulted in breakthroughs in diverse areas, including robotics, gaming, autonomous systems, and even healthcare.

Next, the applications of DRL are discussed, where the versatility of this approach is showcased in solving real-world problems. From autonomous driving and robotic control to financial portfolio management and resource allocation, DRL has demonstrated its effectiveness in domains where traditional optimization methods fall short. By learning from experiences, DRL agents can adapt to changing environments and make intelligent decisions in dynamic and uncertain situations. Moving on, it is explored the components of DRL that contribute to its success.

The key elements are examined such as the agent, environment, and reward system, highlighting their roles in shaping the learning process. Furthermore, the integration of deep neural networks is discussed, which enables DRL models to handle high-dimensional input data, extract meaningful representations, and generalize knowledge across similar scenarios. Understanding these components provides a foundation for comprehending the inner workings of DRL and its implications in mobile sensor networks.

Lastly, the results and discussions are discussed which is obtained through the implementation of DRL techniques in the context of mobile sensor networks. By evaluating the performance of DRL models in comparison to traditional optimization methods, insights into the efficacy of DRL is gained in enhancing network throughput and resource allocation. The discussion encompasses the achieved results, highlighting the strengths and limitations of the approach, and provides valuable insights for future research and applications in the field.

5.1 The DRL based Routing Algorithm

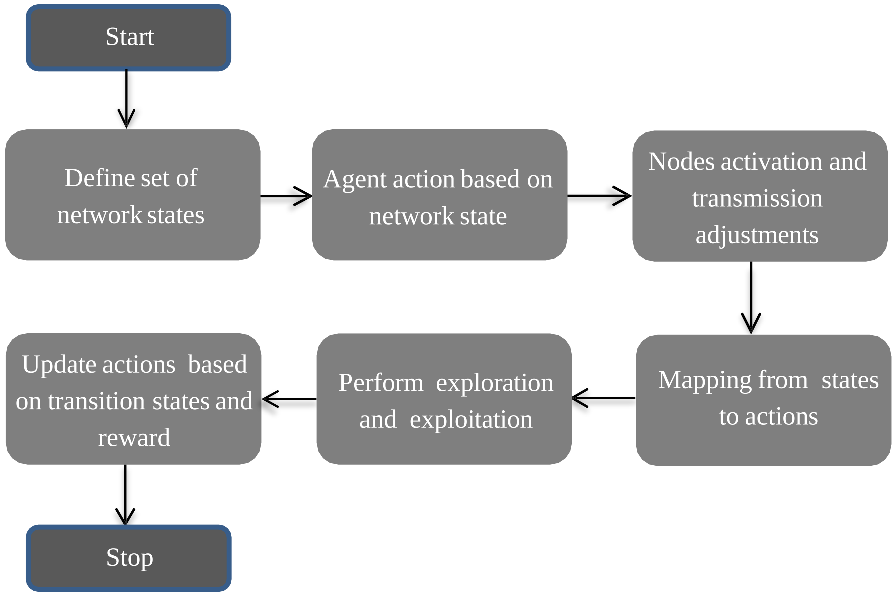


Figure 5.1: Deep RL in mobile sensor networks (Alsalmi, Navaie, and Rahmani, 2024)

Our second proposed algorithm is DRL-OLSR, which employs deep reinforcement learning to optimize routing in MWSN. In this case DRL can be represented as a class of Markov Decision Problems has been the most thoroughly investigated. The agent can only travel to a certain number of states, and for each state visited, a numerical reward will be collected; negative numbers may indicate penalties. There is a variable value associated with each state. There are further states that can be reached through various acts from each state. The averaged future reward that is accumulated by choosing activities from a given state determines that state's value. Actions are chosen following a policy that is also subject to modification.

Training OLSR in MWSNs using DRL involves several steps, including defining the state, action, reward, and training the DRL model. The description of these steps is provided below.

- **State:** The state represents the current state for each node in the environment includes factors such as the location of sensors, their signal strength, the quality of the wireless link, and the traffic load on the network S_i.

- **Action:** As the nodes perform actions such as transmitting packets and executing route requests based on the signal strength between two nodes A_i.

- **Reward:**The reward is used to evaluate the node's performance. it is the immediate feedback to the node by the environment after the node performs an action according to the current state.

5.1.1 DRL-OLSR Mathematical Model

The DRL-OLSR algorithm utilizes DRL to optimize routing in MWSN. This process is depicted in Figure 5.1 and the associated proposed algorithm is presented in Algorithm 2. In the proposed work, the case where Reinforcement Learning (RL) can be represented as a class of Markov decision problems has been extensively studied.

A Markov Decision Process (MDP) consists of four essential components, i.e., state, action, reward, and transition probabilities. In each iteration, the present state is denoted by i, and the agent receives an observation of the environmental state s_i from the set of possible states S. Subsequently, the agent chooses an action a_i from the available actions based on this observation. The set of possible actions for the state s_i is denoted as $A_s i$. When the agent executes an action, it receives a reward value r_i in response. Finally, the agent transitions to the next state with a certain probability known as the transition probability. The transition probability from present state s_i to the next state s_{i+1}, given that the current state is s_i and the action taken is a_i is represented as $P(s_{i+1}|s_i, a_i)$ (Alsalmi, Navaie, and Rahmani, 2024).

The basic illustration of MDP is depicted in Figure 5.2. An episode in this process forms a limited sequence of states, actions, and transition function that returns the next state and rewards in (S, A, δ, R) as given in (5.1).

$$s_0, a_0, r_1, s_1, a_1, r_2, s_2, ..., s_{n-1}, a_{n-1}, r_n, s_n \tag{5.1}$$

Let s_i denote the current state, a_i denote the action taken, r_{i+1} denote the reward obtained after performing action a_i, and (S, A, δ, R) represent the sets of states, actions, and rewards, respectively. The episode concludes when the final state transitions reach s_n. The overall reward is can be represented as,

$$R = r_1 + r_2 + r_3 + ... + r_n \tag{5.2}$$

The ultimate objective of the RL agent is to discover the optimal policy π^* that maximizes the total expected reward, given a set of actions and states (Alsalmi, Navaie, and Rahmani, 2024).

$$\pi^* = \arg\max_{\pi(s)} E\ [R_i + \gamma R_{i+1} + \gamma^2 R_{i+2}] \tag{5.3}$$

where π^* represents the policy of a state for optimal action. $\gamma \in (0, 1)$ denotes the discount factor and shows the importance of immediate and future rewards.

$$Q(s_i, a_i) = \mathbb{E}\left[r_i + \gamma \max_{a'} Q(s_{i+1}, a')\right] \tag{5.4}$$

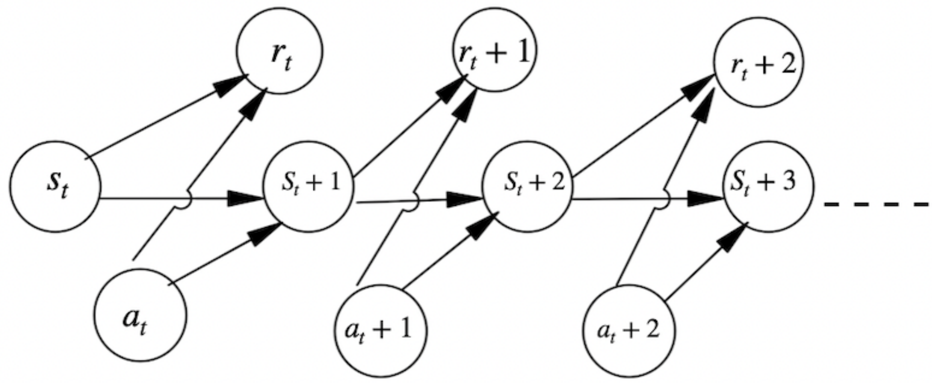


Figure 5.2: The Markov Decision Processes (MDP).

$Q(s_i, a_i)$ stands for the expected immediate reward for acting at in-state s plus the sum of the discount factor and the highest possible expected return in the next state. The definition of this function is based on the intuition that actions should be taken to maximize the expected return at each time step in order to maximize overall reward (Alsalmi, Navaie, and Rahmani, 2024).

Q-learning is one of the popular RL methods to solve MDP. In Q-learning, Bellman's Equation can be used to determine the optimal Q-value function $Q^*(s_i, a_i)$. The DRL model addresses this issue by combining RL and deep learning (DL) techniques. The DRL model uses a deep neural network (DNN) to approximate the Q-values functions.

$$Q_i^*(s_i, a_i) = (1-\alpha)Q_{i-1}(s_i, a_i) + \alpha[r_i + \gamma \max_{a_{i+1}} Q_{i-1}(s_{i+1}, a_{i+1})] \tag{5.5}$$

Here α is the learning rate. The Deep Q-Network (DQN) architecture consists of an input layer, multiple hidden layers, and an output layer. The input layer takes the state of the environment as input, and the output layer produces the Q-value for each action. The hidden layers contain non-linear activation functions that enable the network to learn complex relationships between the input and output (Alsalmi, Navaie, and Rahmani, 2024). The Q-value can be derived as follows:

$$Q_\pi(s_i, a_i) = R(s_i, a_i) + \gamma \sum_{s_{i+1} \in S} P(s_i, a_i, s_{i+1}) Q_i^*(s_{i+1}, a_{i+1}) \tag{5.6}$$

where $R(s_i, a_i)$ represents the reward of action a_i in the state s_i , $P(s_i, a_i, s_{i+1})$ represents the probability of switching to state s_{i+1} after action a_i in the state s_i, and $Q_i^*(s_{i+1}, a_{i+1}) = maxQ_\pi(s_{i+1}, a_{i+1})$ represents the optimal Q-value of action

a_{i+1} in next state s_{i+1} (Alsalmi, Navaie, and Rahmani, 2024). Then, update the Q-value by the following formula:

$$Q_\pi(s_{i+1}, a_{i+1}) = Q_\pi(s_i, a_i) + \alpha \times \Big[R(s_i, a_i) + \gamma \sum_{s_{i+1} \in S} P(s_i, a_i, s_{i+1}) Q_i^*(s_{i+1}, a_{i+1}) - Q_\pi(s_i, a_i)\Big] \quad (5.7)$$

where $\alpha \in [0, 1]$ is the learning rate in(5.7). The optimal action a_i can be obtained as follows:

$$a_i^* = \arg\max\ Q_\pi(s_i, a_i) \quad (5.8)$$

Therefore, the optimal policy can be derived from the optimal action as given in (5.9).

$$L_w = E\left[(Q_\pi(s_i, a_i) - Q_\pi(s_i, a_i^*, w))^2\right] \quad (5.9)$$

where w is the network parameter and the Q-value to be updated up to target Q-value T_Q. Q_π is a predicted Q-value.

For the DRL-OLSR, the actions of a node are restricted to a finite number of states, representing different network conditions. As the node traverses these states, it receives numerical rewards associated with each state visit. It is worth noting that these rewards can be positive, indicating desirable outcomes, or negative, serving as penalties (Alsalmi, Navaie, and Rahmani, 2024).

The main objective of DRL-OLSR is to train the node to make informed decisions on selecting the most appropriate routes based on the observed network states and the associated rewards. By learning from the collected rewards, the agent can optimize the routing decisions and improve the overall performance of the MWSN.

By combining the power of DRL and the benefits of the Optimized Link State Routing (OLSR) protocol, DRL-OLSR aims to enhance the efficiency, reliability, and adaptability of routing in MWSN scenarios.

In the proposed algorithm, each state is associated with a specific variable value. Additionally, there exist multiple states that can be reached through various actions from each state. The value of a particular state is determined by the accumulated average future reward obtained by selecting actions from that state. The selection of actions is guided by a policy, which may be subject to modification as the algorithm progresses.

As mentioned above training the OLSR algorithm in MWSN using DRL encompasses several essential steps. The initial step for training the DRL model is state representation, which captures the current state of each node in the environment. This entails selecting key network parameters such as node energy levels, connection quality, network congestion, and other factors that influence the OLSR algorithm's decision-making process. These parameters together constitute the state information that the DRL agent will use during training.

After this within the network, nodes engage in actions such as transmitting packets or refraining from transmitting them to neighboring nodes. These actions are influenced by factors such as the distance and single strength between two nodes. The distance plays a crucial role in determining the success of packet transmission and significantly impacts network routing behavior. Node actions directly influence the entire decision-making process of the DRL agent, as they contribute to shaping the routing strategy and optimizing network performance.

The most important step is to design the right reward function. Positive rewards are assigned to actions that contribute to improved routing efficiency. These rewards serve as immediate feedback to the nodes from the environment, indicating the positive impact of their actions on the current state. Conversely, negative rewards can be assigned as penalties for actions that lead to routing failures. The reward provides valuable feedback to the nodes, encouraging them to make decisions that optimize network performance and minimize undesirable outcomes.

5.1.2 Training of DRL Model

In the proposed model, the main objective of the learning process is to maximize the agent's predictable cumulative reward. The estimation problem and the control problem are two related calculations that deal with reinforcement learning. The estimation problem deals with the discovery of the value function for the QoS of DRL. At the end of learning, this value function highlights the cumulative sum of the reward that can be predictable when initiating actions at each visited conversion state in the network. The control problem deals with the quality evaluations that maximize reward when moving through state space by relating to the environment. In the end, the network model makes an ideal policy that allows both ideal control and action planning.

Performance improvement by the use of function approximation and utilization of samples is necessary to handle vast environments using two factors that contribute to the effectiveness of reinforcement learning. These two vital elements allow the use of reinforcement learning in vast situations/environments for different purposes. The environment is represented by a simulation model (for simulation-based optimization). Interacting with the environment is the only way to gather information about it. Since there is some kind of model accessible, the issues in reinforcement learning may be classified as planning issues, and some of the issues could be classified as actual learning issues. However, machine learning alters both planning issues through reinforcement learning. Deep learning-based reinforcement learning is explained below:

The Deep reinforcement is evaluated based on equation(5.10).

Algorithm 2 Proposed Algorithm

Step 1: Initialize network specifications such that Net= *F[s]*.
F[s]= All network specifications related to the network length, network radius, number of packers, minimum and maximum radius.
Step 2: Initialize *L(x)* & *W(x)* of mobile sensor networks for the deployment of the network.
L(x) = Length of network in meters
W(x) = Width of network in meters
Step 3: Implement the network deployment $D[n] \subset R[x_i]$ such that $R[x]$ is the iterative nodes locations
Step 4: Implement Nodes placement process such that $ND[x] = \{N_{S_0}, N_{S_1}, ...N_{S_n}\}$

for $x = 1$ to N **do**
  $X_{Loc}(x) = X_{Loc}\ \{Ns(x)\}$
  $Y_{Loc}(y) = Y_{Loc}\ \{Ns(y)\}$
  $Net(p) = f\{Network(X_{Loc}, Y_{Loc}\}$
end for

Step 5: Generate the simulation of the transmission and integration of the nodes with the neighbor nodes in the network.
Step 6: Evaluate the node to node distance
$Dist(n(x) : n(y))$
$Dist = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
where $x \& y$ are the nodes coordinate.

for $x = 1$ *to* c **do**
  if $N(M) \leq Avg(N(M))$ **then**
    Call reinforcement learning network training
    $DRL \Rightarrow (F[s], Q_\pi(s_i, a_i))$
  end if
end for

where M belongs to the node-to-node signal strengths, $N(M)$ belongs to the signal strength, DRL is the deep reinforcement learning, and T_Q is the target Q-value.

Step 7: Evaluate Network Performance such as
$N(p)$=$T(x), C_p(x), E(x), E_{cons}(x), Ov_{cons}(x)$
Step 8: Repeat steps 5 to 7 until all processing gets completed.
Step 9: Stop

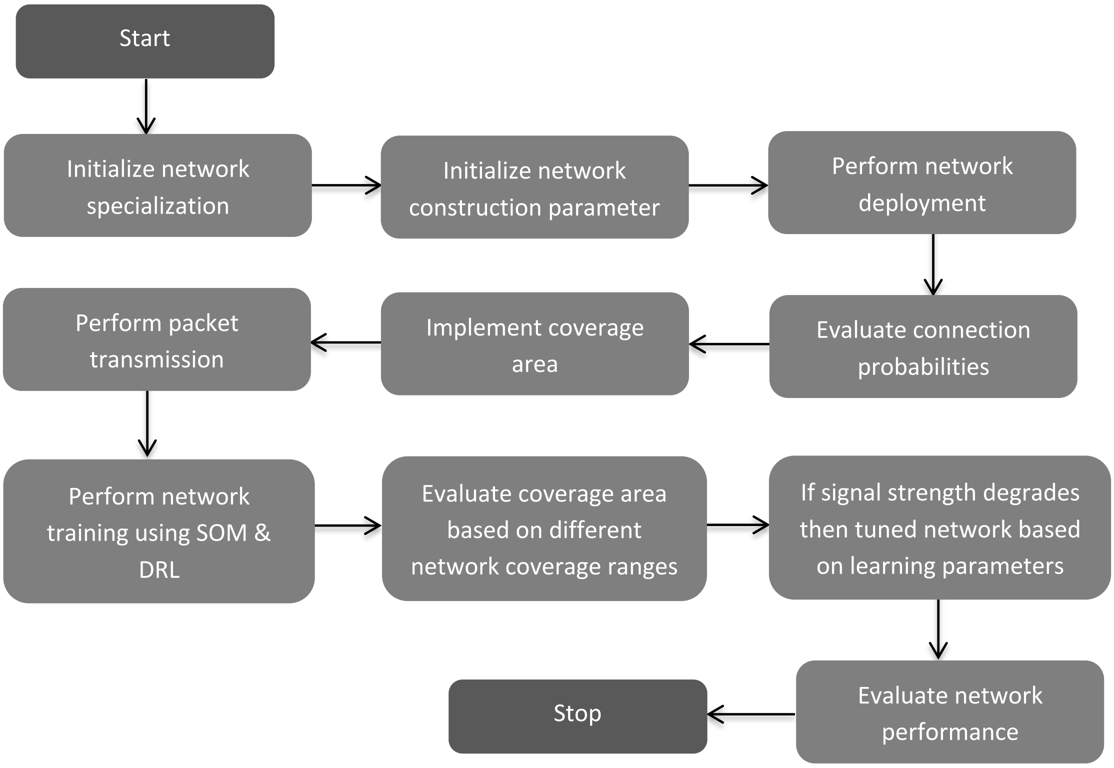


Figure 5.3: The functional block diagram of the proposed method.

$$Q_i\left(s_i, a_i\right) = (1-\alpha) Q_{i-1}\left(s_i, a_i\right) + \alpha\left[r_i + \gamma \max_{a_{i+1}} Q_{i-1}\left(s_{i+1}, a_{i+1}\right)\right] \tag{5.10}$$

where α is the learning rate, r_i is the reward gained γ is a discount factor, and $Q(s,a)$ is the state and actions taken from one transition state to another transition state to attain QoS.

5.2 Result and Discussion

The MWSN network is simulated using the MATLAB environment, with the simulation parameters listed in Table 4.1. The network comprises multiple nodes that are randomly distributed as depicted in Figure 4.3. Using the node-to-node infrastructure, packets are broadcasted to surrounding nodes based on the network's coverage areas and transmission ranges. The final route for packet transmission is indicated by a green dotted line, which is evaluated to assess the network's performance. The simulation was conducted utilizing artificial neural networks and deep learning-based reinforcement learning techniques. The whole proposed methodology used in simulation analysis is illustrated in Figure 5.3.

Figure 5.4 shows the training process using reinforcement learning based on the number of episodes run to achieve low losses and based on which the rewards are

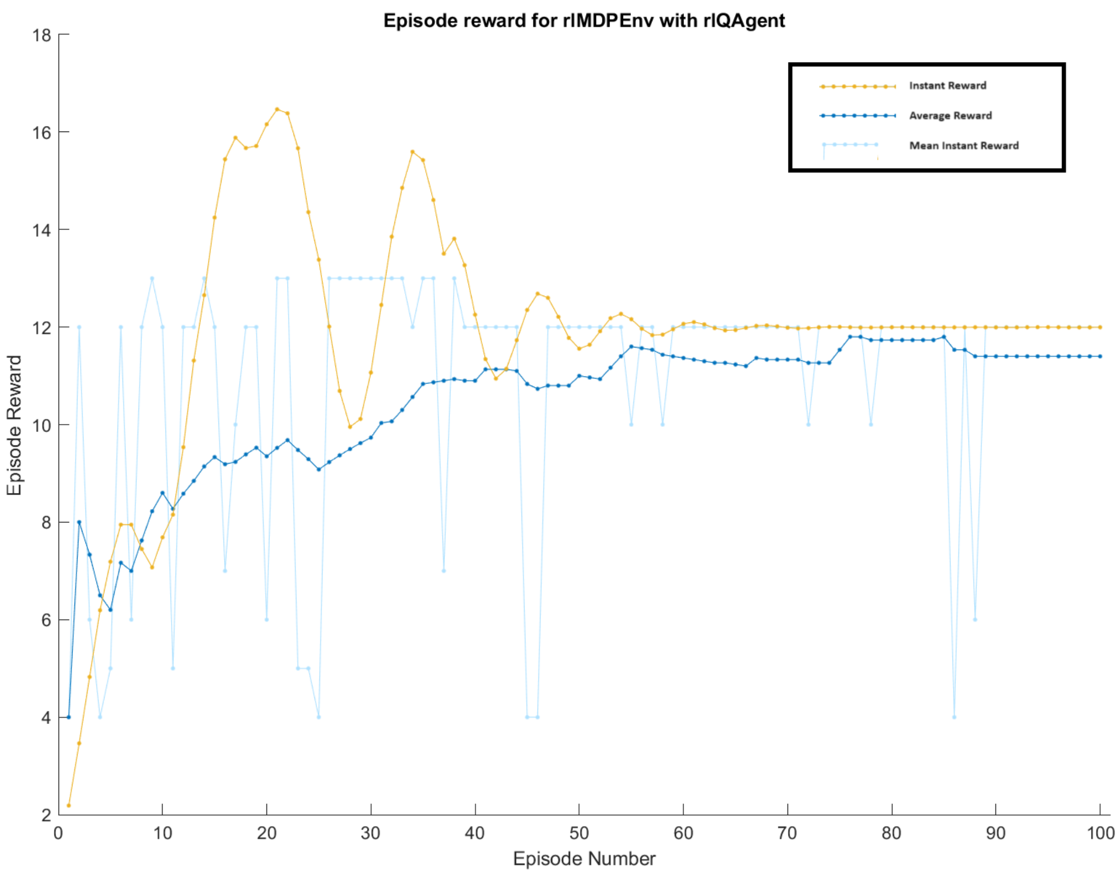


Figure 5.4: Deep Reinforcement Learning Training Process

evaluated. It can be seen from Figure 5.4 the commutative reward is increasing which shows the quality of learning and efficient decision-making process to increase the strength and frequency of the behaviors through which low energy consumption with fewer path losses and low network error rates are achieved.

The blue line signifies the episode reward. The episode is used as a functional part on which the agent performance is evaluated in deep reinforcement learning. Episode reward signifies a single instance of the agent interacting with the environment and completing a task or goal.

Orange dark line signifies the Average Reward shows the average reward received by the agent over a certain number of episodes. The trend of this line over time can give you an idea of how well the agent is learning and improving its performance.

Light yellow line signifies the quality of the current episode (Episode Qo) being executed by the agent. The quality of an episode can be measured in terms of the total reward obtained by the agent during that episode. During each episode, the agent interacts with the environment, takes actions based on its current policy, and receives rewards based on the outcome of those actions. The total reward obtained by the agent during an episode can be used as a measure of how well the agent is performing in that episode. This line is important in reinforcement learning because the goal of the agent is to learn to maximize its expected cumulative reward over the long term, which requires performing well in each individual episode.

5.2.1 Results and Discussion using DRL

The Figure 5.5 shows the analysis carried out in comparison with SOM network, the connection probability in the case of reinforcement learning is high. This analysis focuses on the connection probability, particularly highlighting the higher connectivity observed in the case of DRL. In the context of network analysis, connection probability refers to the likelihood of establishing connections between different nodes or elements within the network. In this comparison, the connection probability is examined in the context of using SOM and DRL techniques. The analysis reveals that in the case of DRL, the connection probability is relatively high compared to that of SOM. This indicates that the DRL approach leads to a higher likelihood of establishing connections between nodes or elements in the network. This observation suggests that DRL is more effective in promoting connectivity and interaction between different components in the system. The higher connection probability in DRL can be attributed to the unique characteristics of the DRL algorithm.

RL is a learning paradigm where an agent interacts with an environment, learns from experiences, and adjusts its decision-making process to maximize rewards. Through this iterative learning process, the DRL agent explores different actions and learns to establish connections that lead to favorable outcomes or rewards. On the other hand, SOM networks, while also capable of learning and organizing data patterns, may exhibit a lower connection probability compared to DRL. SOMs are

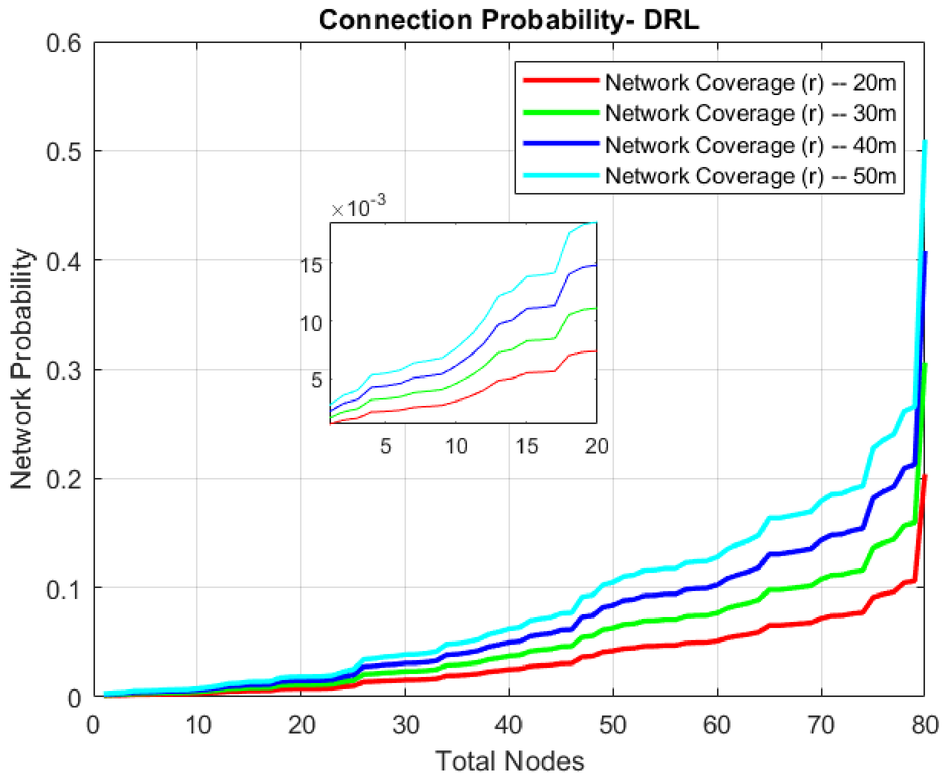


Figure 5.5: Connection probability using DRL

typically designed to preserve the topology of the input data in the output space, emphasizing the organization of data clusters rather than explicit connectivity between individual elements. The contrasting connection probabilities between SOM and DRL highlight these techniques' different approaches and objectives. While SOM focuses on clustering and organizing data patterns, RL emphasizes the establishment of connections that maximize rewards. The higher connection probability in RL suggests that it may be more suitable for tasks that require strong interaction and coordination between different components or agents in the network.

The results demonstrate in Figure 5.6 that the DRL network exhibits significantly lower end-to-end delay compared to the SOM network. End-to-end delay refers to the time a packet or data takes to traverse the network from the source node to the destination node. A lower end-to-end delay is desirable as it signifies faster and more efficient communication within the network. In the analysis, the DRL network's performance in terms of end-to-end delay outperformed the SOM network. This suggests that the DRL-based network communication and decision-making approach is more effective in reducing delays and achieving faster data transmission. The advantage of DRL lies in its ability to learn optimal decision-making policies through trial-and-error interactions with the environment. By continuously learning and adapting its policies based on received rewards, the DRL network can make efficient routing decisions, leading to reduced end-to-end

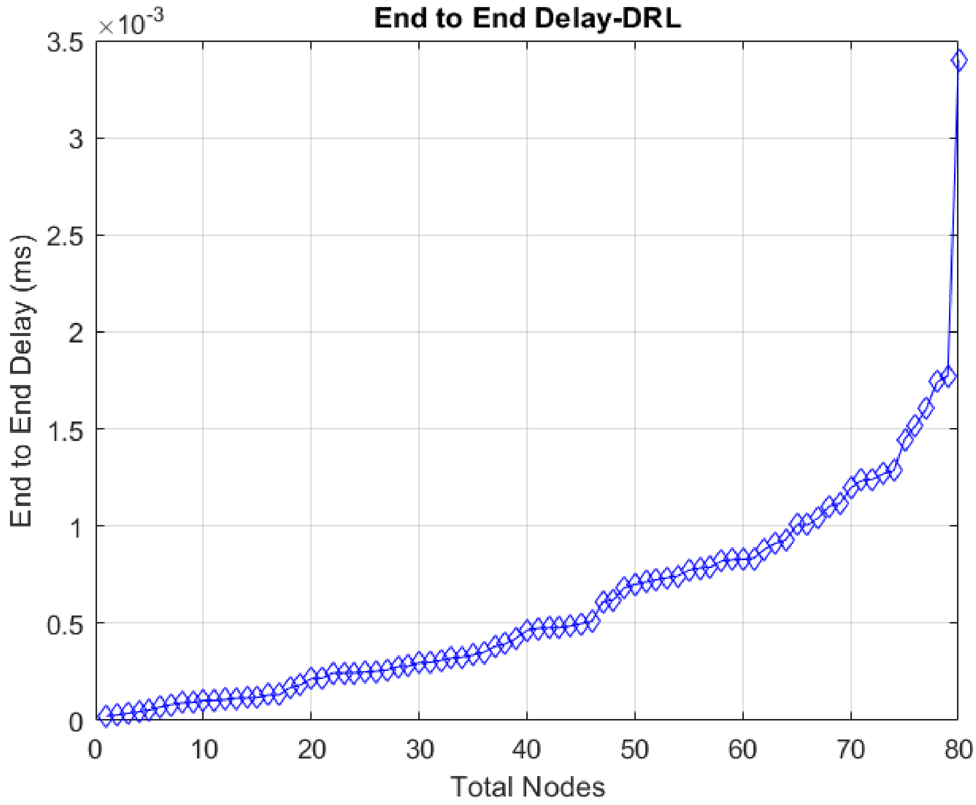


Figure 5.6: End to End Delay using DRL

delay. On the other hand, the SOM network, which utilizes a different approach for communication and decision-making, exhibited higher end-to-end delay.

The SOM network is known for its topological structure preservation and clustering capabilities, but in terms of minimizing delay, it may not be as efficient as the DRL network. The lower end-to-end delay in the DRL network suggests that it has the potential to enhance real-time communication, reduce latency, and improve overall network performance. These advantages make DRL particularly suitable for applications that require fast and timely data transmission, such as real-time monitoring, autonomous systems, and time-sensitive control systems.

TheFigure 5.7 shows the analysis carried out in caomparison with SOM network. Specifically, the overhead consumption in the case of DRL is found to be significantly lower compared to SOM. In the context of this analysis, overhead consumption refers to the additional resources, such as computational power, memory, or time, required by a network or algorithm to perform a task. Lower overhead consumption is advantageous as it indicates that the network or algorithm is more efficient and utilizes fewer resources to achieve the desired outcome. In the case of DRL, the overhead consumption was observed to be remarkably low. This can be attributed to the inherent nature of DRL algorithms, which leverage deep learning and reinforcement learning techniques to train agents to make optimal

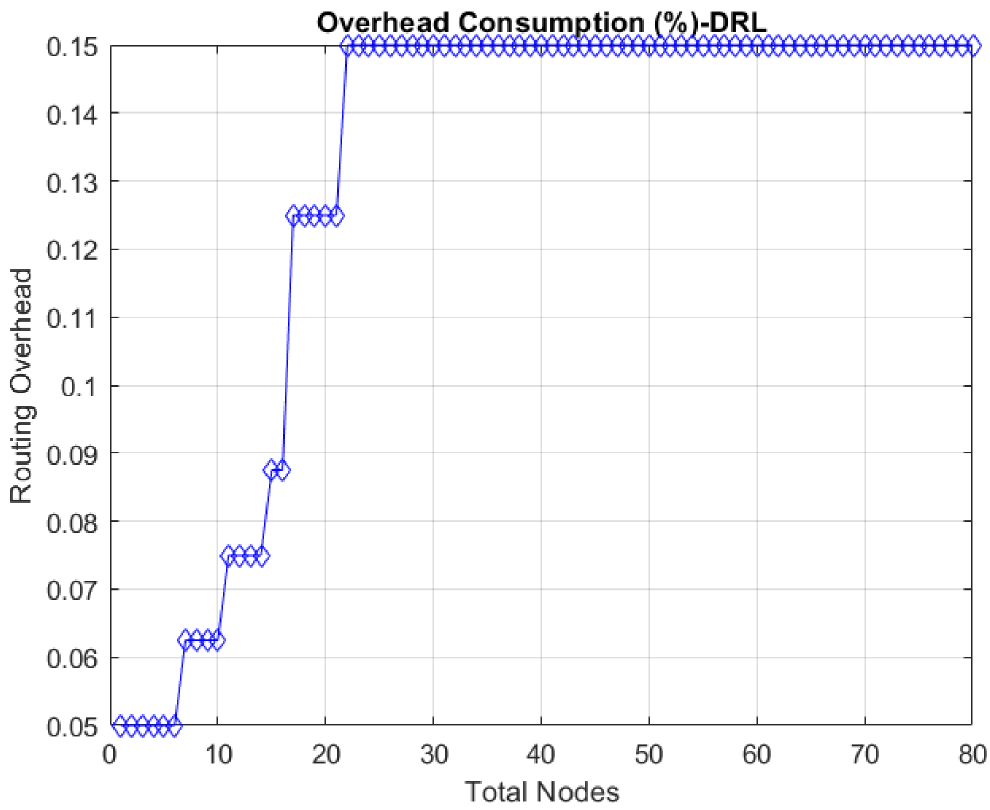


Figure 5.7: Overhead consumption using DRL

decisions. By combining these approaches, DRL agents can learn directly from raw sensory inputs, such as images or sensor data, without the need for explicit feature engineering or preprocessing.

This streamlined process reduces the computational burden and memory requirements, resulting in lower overhead consumption. On the other hand, SOM networks exhibited higher overhead consumption in the comparative analysis. SOMs are unsupervised learning algorithms that organize a grid of neurons to represent the input data's topological structure. While SOMs can be effective in clustering and exploring data patterns, they often require additional computational resources for training and maintaining the grid structure. The computation involved in adjusting the weights, updating the node linkages, and preserving the topology can contribute to higher overhead consumption.

The disparity in overhead consumption between DRL and SOM networks suggests that DRL offers a more efficient solution in terms of resource utilization. The reduced overhead consumption in DRL can have practical implications, such as faster training times, improved scalability, and lower computational costs. These benefits make DRL particularly attractive in scenarios where resource efficiency is crucial, such as in resource-constrained environments or real-time applications.

The analysis focuses in Figure 5.8on the throughput metric, which represents

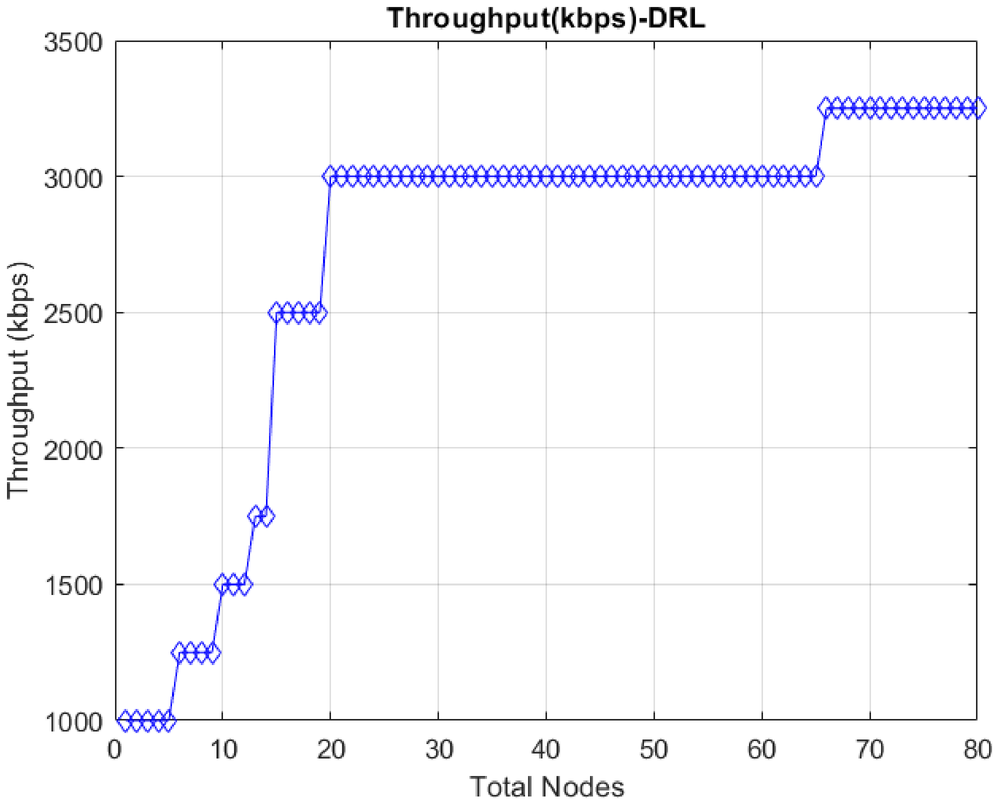


Figure 5.8: Throughput using DRL

the amount of data successfully transmitted in a given time frame. In this comparison, the figure reveals that the throughput achieved by the DRL network is significantly higher when compared to the SOM network. Throughput refers to the rate at which data is transmitted, and a higher value indicates a greater amount of data successfully transmitted within a given time period. The superior throughput observed in the DRL network can be attributed to the inherent capabilities of Deep Reinforcement Learning algorithms.

DRL algorithms leverage the power of deep neural networks combined with reinforcement learning techniques to optimize decision-making policies. By learning from interactions with the environment and receiving feedback in the form of rewards or penalties, the DRL network can adapt its decision-making process to maximize the cumulative reward over time. The DRL network's ability to achieve higher throughput can be attributed to several factors. First, DRL networks can leverage the representation learning capabilities of deep neural networks to extract complex features and patterns from the input data. This enables the DRL network to capture and utilize relevant information, leading to more effective decision-making. Additionally, the DRL network's adaptive nature allows it to dynamically adjust its policy based on the changing network conditions.

Through trial and error interactions with the environment, the DRL network learns optimal strategies to achieve higher throughput, taking into account factors such as channel conditions, congestion levels, and other relevant network parameters. On the other hand, the SOM network, while still a valuable tool for certain applications, may exhibit comparatively lower throughput due to its different underlying principles. SOM networks are primarily designed for clustering and visualization purposes, rather than optimizing network performance metrics like throughput. Although SOM networks can provide valuable insights into data patterns and relationships, they may not be as effective in maximizing data transmission rates.

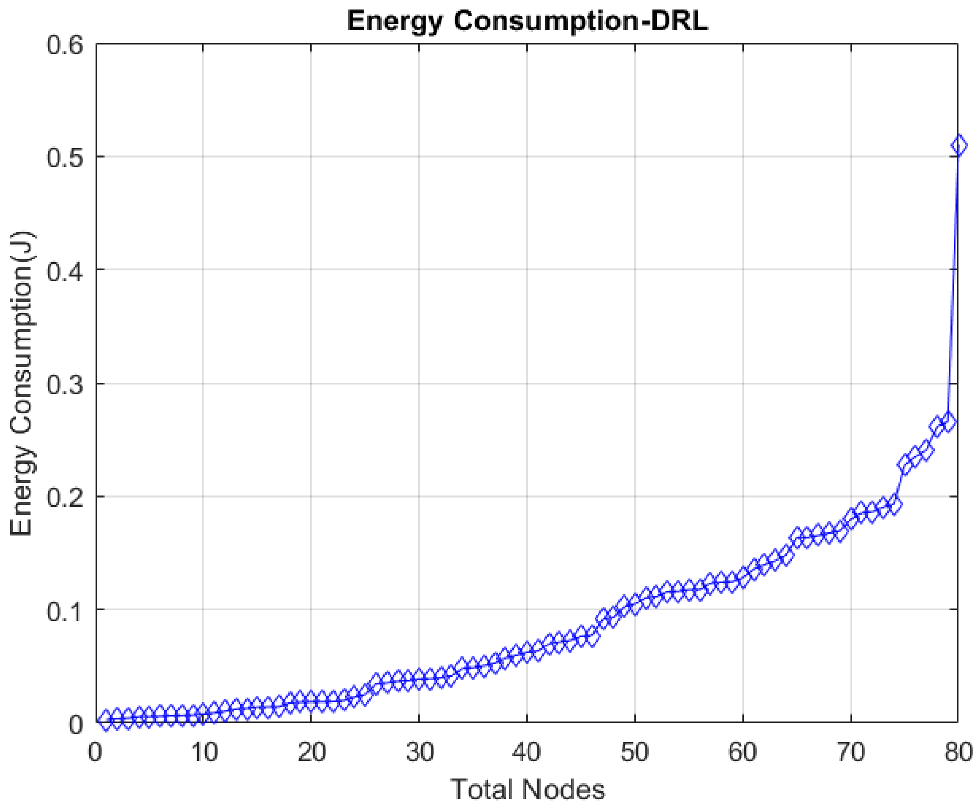


Figure 5.9: Network Energy Consumption using DRL

The findings revealed in Figure 5.9 that DRL exhibits significantly lower energy consumption compared to SOM. Deep Reinforcement Learning utilizes advanced techniques from deep learning and reinforcement learning to train intelligent agents. These agents learn optimal decision-making policies through interactions with their environment, guided by a reward signal. The training process involves trial and error, where the agent gradually improves its policy based on the received rewards. On the other hand, Self-Organizing Map networks are a type of artificial neural network used for clustering and visualization of data patterns. While SOMs are effective for certain tasks, they can consume relatively higher energy due to their inherent architecture and operations.

The analysis demonstrated that DRL requires fewer computational resources and operates more efficiently, resulting in reduced energy consumption. This efficiency can be attributed to several factors. First, DRL employs deep neural networks that have been optimized for training on large-scale datasets, making them more computationally efficient. Secondly, DRL agents learn directly from raw sensory inputs, such as images or sensor data, without the need for extensive preprocessing.

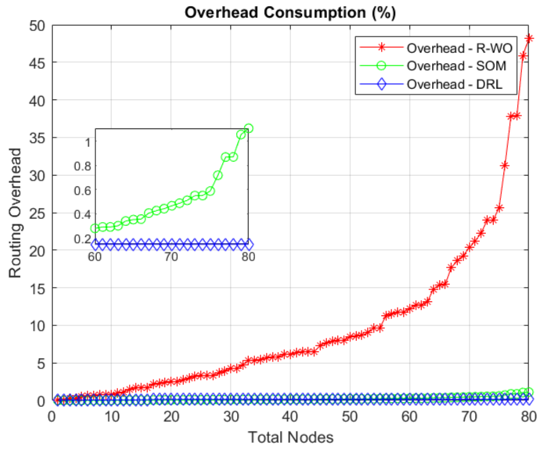
This eliminates the energy-intensive task of feature extraction, which is often required in SOM networks. Furthermore, DRL agents continually adapt their decision-making policies based on real-time feedback, enabling them to optimize their actions efficiently. In contrast, SOM networks rely on a fixed architecture and do not dynamically adjust their operations during runtime, leading to potentially higher energy consumption. The lower energy consumption of DRL has significant implications in various domains. For example, in resource-constrained environments, such as mobile sensor networks or Internet of Things (IoT) devices, minimizing energy usage is crucial for prolonging the operational lifespan and enhancing network efficiency.

Moreover, the reduced energy consumption of DRL can translate into cost savings. By minimizing energy requirements, organizations can lower their electricity bills and optimize the use of computational resources, resulting in increased cost-effectiveness.

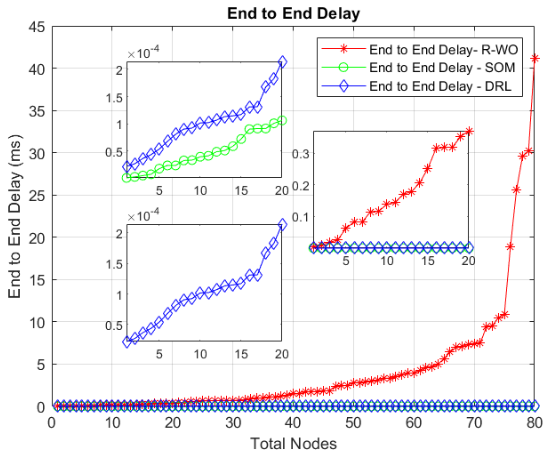
Figure 5.10 shows the performance comparison of our proposed work in a graphical view. It can be clearly seen that the DRL is achieving appropriate performance and shows high network throughput. It shows that the packets are successfully received as per the total transmissions takes place. Also SOM is performing well but slightly less than the DRL as it is a deep quality based learning approach which includes dense arrangement of the neurons in the network. Also the overhead and end delay is less in DRL which shows that out proposed approach is able to achieve high quality of service.

Routing overhead which can be seen in Figure 5.10a is increasing up to some extent but it is not increasing up to the deteriorated condition. If overhead increases then the transmission of the packet failure increases which will increase high bit error rates which should not be the desired output. As per the DRL structure, the training of the network is very densely evaluated which performs high-quality service performance in terms of controlled end delay as shown in Figure 5.10b, high network throughput Figure 5.10c, and energy consumption as seen in Figure 5.10d. The throughput should be high as possible which shows high successful receiving of the packets

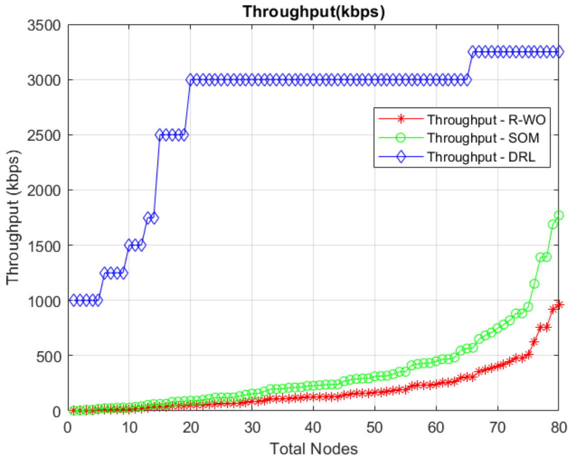
In the performance evaluation in Table 5.1, three scenarios were compared: using Self-Organizing Maps (SOM), using Deep Reinforcement Learning (DRL), and a random scenario as a baseline. The evaluation focused on connection probabilities as a metric to measure the performance of each approach. The results revealed that DRL outperformed both SOM and the random scenario. In the SOM



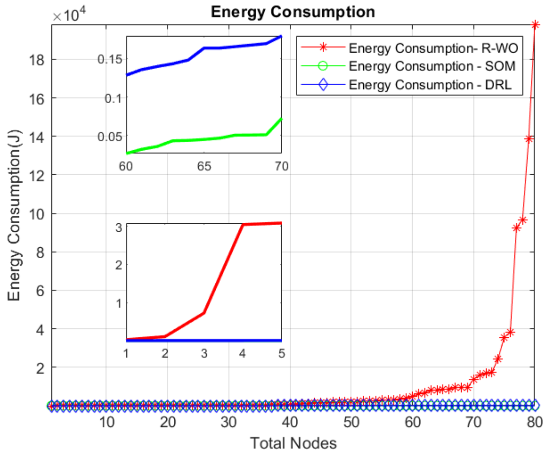
(a) Routing Overhead



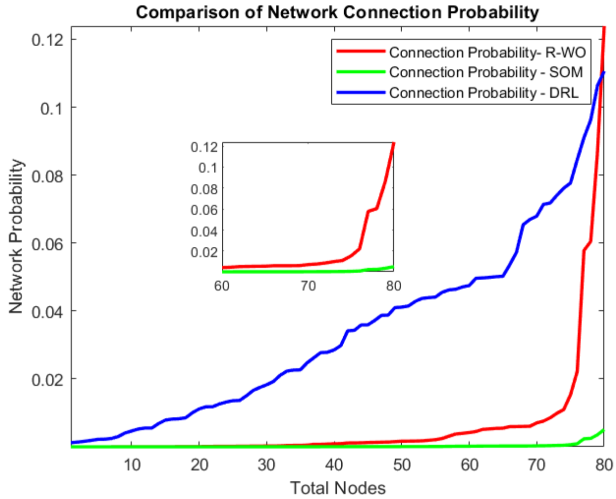
(b) E2E Delay



(c) Network Throughput



(d) Energy Consumption



(e) Probability Comparison

Figure 5.10: Performance comparison of a Routing overhead b E2E delay c Network Throughputd Energy Consumption and e Connection Probability vs. total number of nodes.

Table 5.1: Connection Probability

Number of Nodes	R:WO	SOM	DRL
20	.0008	0.0004	0.01
40	.001	0.0006	0.025
60	.001	0.0007	0.54
80	.12	0.01	0.1

scenario, connection probabilities were calculated based on the algorithm's ability to identify similar transmission patterns and reduce redundancy in the network. However, the performance of SOM in terms of connection probabilities was found to be lower compared to DRL.

This suggests that SOM might struggle in effectively capturing and representing the underlying patterns and optimizing the connections between nodes. On the other hand, DRL exhibited superior performance in terms of connection probabilities. The use of DRL algorithms allowed the agent to learn optimal decision-making policies through trial and error interactions with the environment. By leveraging deep learning techniques, DRL was able to capture complex patterns and make more informed decisions regarding the connections between nodes in the network. This resulted in higher connection probabilities and a more efficient utilization of network resources. In comparison to the without optimization, where connections were made randomly without any optimization, both SOM and DRL demonstrated improved performance. However, DRL surpassed both the random scenario and SOM, indicating its ability to learn and adapt to the network environment in a way that enhances connection probabilities.

The higher connection probabilities achieved by DRL signify its capability to allocate network resources more effectively and reduce redundancy in the network. This optimized connectivity leads to improved data transmission efficiency, reduced energy consumption, and enhanced overall performance of the network.

Table 5.2: End-to-End Delay

Number of Nodes	R:WO	DRL	SOM
20	0.35	0.00015	0.00002
40	2.1	0.0003	0.00025
60	4.3	0.001	0.0005
80	42	0.012	0.002

In the performance evaluation in Table 5.2, three methods are compared: Without Optimization, Self-Organizing Maps (SOM), and Deep Reinforcement Learning (DRL). The evaluation metric used is the end-to-end delay, which measures the time taken for data packets to travel from the source to the destination in the network.

Without Optimization refers to the scenario where no specific optimization techniques or algorithms are applied to reduce the end-to-end delay. This approach typically leads to high delay times and suboptimal performance as the network relies on default routing and transmission mechanisms. SOM, a popular unsupervised learning algorithm, is applied to improve the performance of the network. It provides a structured grid-like representation of the network, where neighboring nodes exhibit similar transmission patterns. This clustering helps reduce redundancy and optimize data transmission. As a result, the end-to-end delay is moderate, indicating an improvement over the Without Optimization scenario. DRL, on the other hand, surpasses both Without Optimization and SOM in terms of performance. DRL combines deep learning and reinforcement learning techniques to train an agent that learns optimal decision-making policies.

In the context of the network, DRL is able to adapt and optimize routing and transmission strategies based on the observed rewards (e.g., low delay, high throughput). By continuously learning and adapting, the DRL agent significantly reduces the end-to-end delay, leading to superior performance compared to both Without Optimization and SOM. The Table 5.2 summarizes the comparison, clearly indicating that DRL outperforms both SOM and Without Optimization in terms of end-to-end delay. The performance improvement achieved by DRL showcases the potential of deep reinforcement learning techniques in optimizing network performance and reducing delays in data transmission.

Table 5.3: Overhead Consumption

Number of Nodes	R:WO	SOM	DRL
10	1	0.05	0.063
20	2.5	0.1	0.125
30	4.8	0.15	0.15
40	6.8	0.18	0.15
50	8.2	0.2	0.15

In the performance evaluation, three approaches were compared: without optimization, using Self-Organizing Maps (SOM), and using Deep Reinforcement Learning (DRL). The evaluation focused on measuring the overhead consumption, throughput, and energy consumption of the network as a key performance metric. The results indicated that DRL outperformed both SOM and the approach without optimization. The approach without optimization served as a baseline, representing the network's overhead consumption in its default state, without any specific optimization techniques or algorithms applied. This baseline allowed for comparison against the other two approaches and provided a reference point for evaluating the improvements achieved by SOM and DRL. SOM, a clustering

Table 5.4: Throughput

Number of Nodes	R:WO	SOM	DRL
10	10	50	1.5×10^3
20	40	100	3×10^3
40	98	220	3×10^3
60	240	450	3×10^3
80	980	1800	3.25×10^3

technique, was utilized as an optimization method to reduce overhead consumption and energy consumption in the network. The results prove the superiority of DRL in terms of overhead consumption as shown in Table 5.3.

SOM aimed to identify clusters of similar transmission patterns, thereby reducing redundancy and improving overall network efficiency. The evaluation measured the energy consumption of the network when SOM was applied and compared it against the baseline. On the other hand, DRL, an advanced technique combining deep learning and reinforcement learning, was also employed for network optimization. DRL agents learned optimal decision-making policies by interacting with the environment and maximizing cumulative rewards. In this case, the objective was to minimize energy consumption without compromising with throughput and maintaining desired network performance. The evaluation measured the throughput and energy consumption when DRL was utilized and compared it against both the baseline and the SOM approach as given in Table 5.4 and Table 5.5.

The results of the evaluation demonstrated that both SOM and DRL outperformed the approach without optimization in terms of energy consumption. However, DRL showed superior performance compared to SOM. The application of DRL resulted in a more significant reduction in energy consumption compared to both the baseline and the SOM approach. The effectiveness of DRL in achieving better energy efficiency can be attributed to its ability to adapt and learn optimal policies through trial-and-error interactions. By continuously optimizing its decision-making process, DRL agents were able to make more informed choices, leading to more efficient resource allocation and reduced energy consumption within the network.

Overall, the performance evaluation highlighted the advantages of employing optimization techniques, such as SOM and DRL, to reduce energy consumption in mobile sensor networks. While SOM showed improvements compared to the approach without optimization, DRL emerged as the most effective method, surpassing both SOM and the baseline in terms of energy efficiency. These findings emphasize the potential of advanced optimization techniques, specifically DRL, in achieving significant energy savings and enhancing the overall performance of mobile sensor networks.

Table 5.5: Performance Comparison

Parameter	Base [1]	Proposed
Energy Consumption	0.15	0.05
Network Probability	0.004	0.11

—

Chapter 6

Aggregation Model

This Chapter discusses aggregation methods for Deep Reinforcement Learning (DRL) based wireless sensor networks. The aggregation algorithm plays a vital role in reducing overall energy consumption. The aggregation method aggregates the data samples, removes the redundant data, and reduces the number of overall packets, which results in a reduction in overall energy consumption. The discussion starts with the introduction, followed by the discussion on routing protocols with aggregation for MWSNs, which presents an overview of different routing protocols with aggregation. The performance of Optimized link state routing (OSLR) without aggregation is compared with the OLSR protocol with aggregation. Subsequently, routing protocols based on SOM with reference to WSNs are elaborated. The SOM-OLSR with aggregation is reviewed and compared with the SOM-based routing protocol without aggregation for MWSNs.Similarly, the DRL with aggregation is reviewed and compared with the DRL-based routing protocol without aggregation for MWSNs.

6.1 Mathematical Model of Aggregation Protocol

The OLSR protocol transmits the data via various routing devices. The nodes aggregate the data at each fully function device (FFD) at an intermediate stage.

A state space model is developed for the proposed OLSR protocol with SOM. Let us assume that the data samples collected for a particular sensor ID are $x_1, x_2, x_3....x_n$. Learning Rate (η) defines a decreasing learning rate schedule. Neighborhood Function ($h(i, c(t))$) defines a neighborhood function that describes how the influence of neighboring neurons changes during training. Input data samples are represented $x(t)$ iteratively to the SOM. For each input data sample $x(t)$, the best matching unit (BMU) is defined as $c(t)$ for the neuron with the closest weight vector:

$$c(t) = \arg\min_i \|x(t) - w_i(t)\| \tag{6.1}$$

Which update the weights of the winning neuron and its neighbors:

$$W_i(t+1) = W_i(t) + \eta(t) \cdot h(i, c(t)) \cdot [x(t) - w_i(t)] \tag{6.2}$$

The data is aggregated for each neuron i during or after training. One common method is to compute the mean of the data samples associated with each neuron:

$$\bar{x}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} X_{ij} \tag{6.3}$$

Where N_i number of data samples associated with neuron i.

The aggregation of all data samples depends upon the aggregation function. The data packets are aggregated using the average value function in the present work. However, the aggregated average value is transmitted only if the deviation is smaller than the deviation threshold (X_{Th}).

$$D_{Agg} = \begin{cases} (1/N) \cdot \sum_{i=1}^{N} x_i, & \text{if } \frac{\sum_{i=1}^{N}\left(x_i - (1/N) \cdot \sum_{i=1}^{N} x_i\right)^2}{(1/N) \cdot \sum_{i=1}^{N} x_i)} \leq X_{Th} \\ x_1, x_2, x_3 ...., x_N, & \text{otherwise} \end{cases} \tag{6.4}$$

The learning rate and neighborhood function parameters are updated during training. Typically, the learning rate decreases, and the neighborhood function narrows over time.

$$\begin{aligned} \eta(t+1) &= \eta_{\text{initial}} \cdot \exp\left(-\frac{t}{time_{constant}}\right) \\ h(i, c(t)+1) &= \exp\left(-\frac{\| \text{ position } (i) - \text{position}(c(t)+1)\|^2}{2 \cdot \text{ neighborhood } (t+1)^2}\right) \end{aligned} \tag{6.5}$$

To develop an aggregation method for OLSR with deep reinforcement learning, a state space model is developed for the proposed OLSR protocol. The aggregation method is analyzed with DRL below. The state space at time step t includes information about the link qualities and queue lengths for each node i. The state of node i is represented at time t as a vector $\mathbf{s}_t^i$:

$$\mathbf{s}_t^i = [qu_t^i, L_Q^i] \tag{6.6}$$

Where qu_t^i is the queue length, L_Q^i is the link quality at node i at time t.

As already discussed, the action space and reward function are The action space includes possible actions that the deep reinforcement learning agent can take at each node i at time step t. The action taken by node i at time t is represented as $\mathbf{a}_t^i$ as given below:

$$\mathbf{a}_t^i = [a_t^i, N_H^i] \tag{6.7}$$

Where a_t^i represents the local action taken at node i, and N_H^i is a next hop for i_{th} node representing the actions related to next-hop node selection or transmission power adjustments.

In reinforcement learning, the reward function is used to evaluate the performance of the corresponding action in a given state. It typically considers energy consumption, data delivery latency, and data aggregation efficiency. Let R_t^i denote the reward obtained by node i at time t as given below:

$$R_t^i = \text{Reward}(s_t^i, a_t^i) \tag{6.8}$$

The OLSR protocol is used with data aggregation to remove redundant data and energy-efficient data transmission. The reinforcement learning algorithm is integrated with OLSR to optimize the cumulative reward over time by learning a policy π that maps states to actions. One common approach is using Deep Q-Learning, which uses a deep neural network with aggregation function D_{Agg} to represent the Q-function, denoted as $Q(s_t^i, a_t^i; D_{Agg})$. The Q-function estimates the expected total reward when taking action a_t^i in state s_t^i and following the policy afterward.

The Q-learning update rule can be represented as follows:

$$\begin{aligned} Q(s_t^i, a_t^i; D_{Agg}) \leftarrow\ & Q(s_t^i, a_t^i; D_{Agg}) + \\ & \alpha \left[R_t^i + \gamma \max_{a'} Q(s_{t+1}^i, a'; D_{Agg}) - Q(s_t^i, a_t^i; D_{Agg}) \right] \end{aligned} \tag{6.9}$$

Where:

- α is the learning rate, determining the step size of the updates.
- γ is the discount factor, representing the importance of future rewards.

The deep neural network is trained by minimizing the Mean Squared Error (MSE) loss between the target Q-values and the predicted Q-values.

6.2 Result Analysis and Discussion

The results are analyzed for three scenarios. The results are analyzed for OLSR (with and without data aggregation) in the first scenario. In scenario 2, the results are analyzed for SOM-OLSR for aggregation condition. Similarly, scenario 3 analyzes DRL-OLSR for aggregation conditions.

6.2.1 Scenario-1

On comparison of the energy consumption without reinforcement learning & without data aggregation (WO-RL-WO-DA) with without reinforcement learning

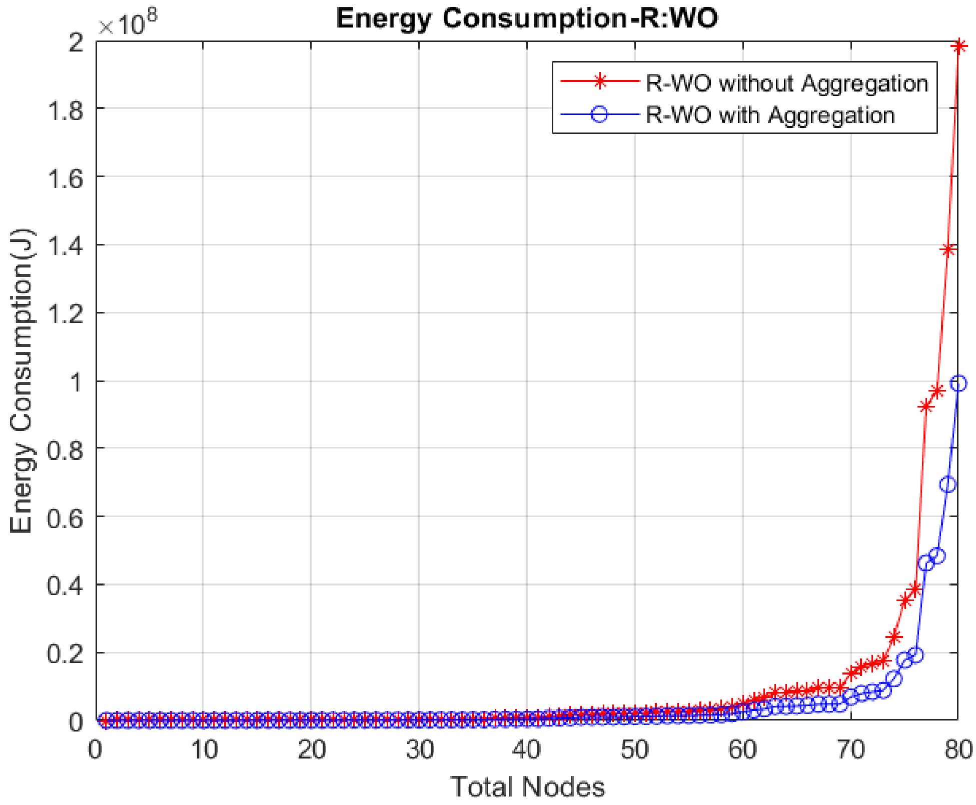


Figure 6.1: Energy Consumption of OLSR with and without aggregation

& with data aggregation (WO-RL-W-DA), The reduction in Energy consumption can be noticed from the graph. The energy consumption starts when the no. of total nodes reaches 60 and beyond. It is observed, particularly when there are 70 total nodes, the consumption of energy of WO-RL-WO-DA is higher when compared to WO-RL-W-DA. The same can be implied as the no. of nodes is increasing, and the power consumption is exponentially increasing at a high rate when there is no Data aggregation. The maximum difference can be observed when there are a total of 80 nodes. With Data aggregation, the consumed energy is 10^8 J. This equals half of 2×10^8 J consumed Without data aggregation. The higher energy consumption results in lesser efficiency. Here statistically proving that WO-RL-W-DA is preferred over WO-RL-WO-DA.

6.2.2 Scenario-2

Self-Organizing Maps Based-Optimized Link State Routing (SOM-OLSR) without data aggregation and SOM-OLSR with data aggregation are compared regarding energy usage. The graph shows how energy use has decreased. When there are 60 or more total nodes, energy consumption begins. It is observed that SOM-OLSR without Aggregation consumes more energy than SOM-OLSR with Aggregation,

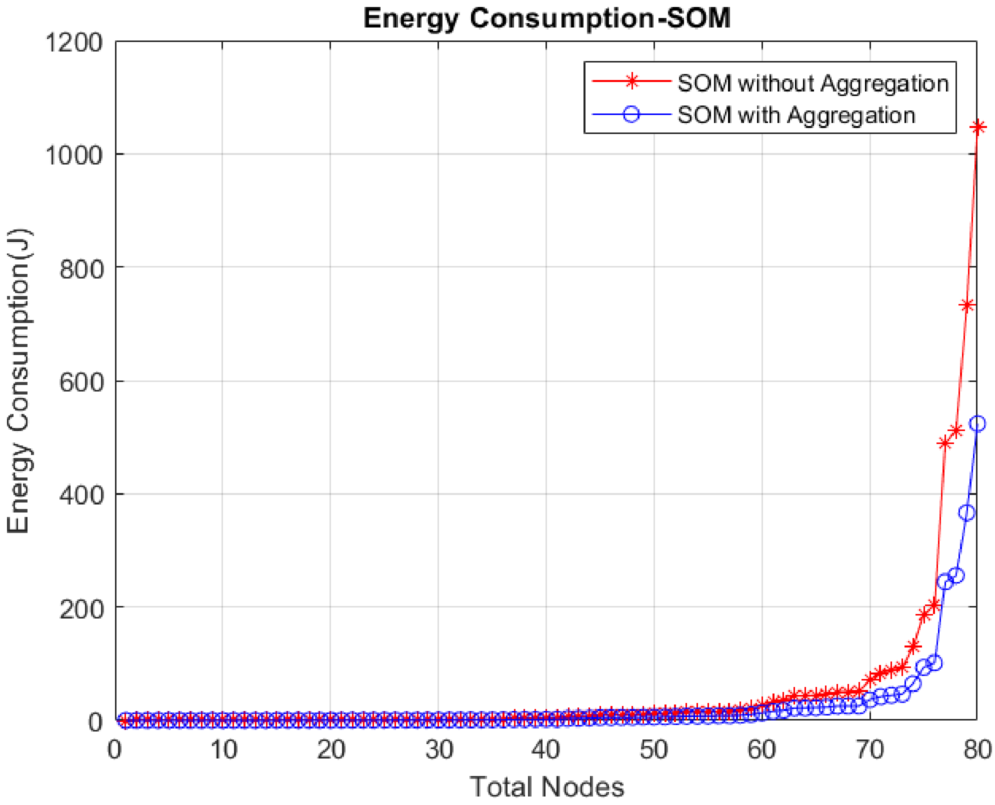


Figure 6.2: Energy Consumption of SOM-OLSR with and without aggregation

especially when there are 70 total nodes. When there is no data aggregation, it may be inferred that as the number of nodes grows, so does the power consumption exponentially and rapidly. The most significant difference may be seen when there are 80 total nodes and 510 J of energy is used for data aggregation. This is equivalent to using half the 1020 J Without data aggregation. Less efficiency is produced by using more energy. The SOM-OLSR with Aggregation is preferable here statistically compared to the SOM-OLSR without Aggregation.

6.2.3 Scenario-3

After implementing the Deep Reinforcement Learning Based-Optimized Link State Routing (DRL-OLSR), There are 2 sets of data collected with Data aggregation and without it. In both cases, the consumption of energy starts when the nodes are above 10. With the increase in no. of nodes, the DRL-OLSR with data aggregation slowly rises at a similar rate as that of DRL-OLSR without data aggregation. The rise in energy consumption is noticeable at the point after which the no. of nodes is increased by more than 30.

We can see both consumptions increment but at different rates. The DRL-OLSR without data aggregation is growing in consumption up to approximately

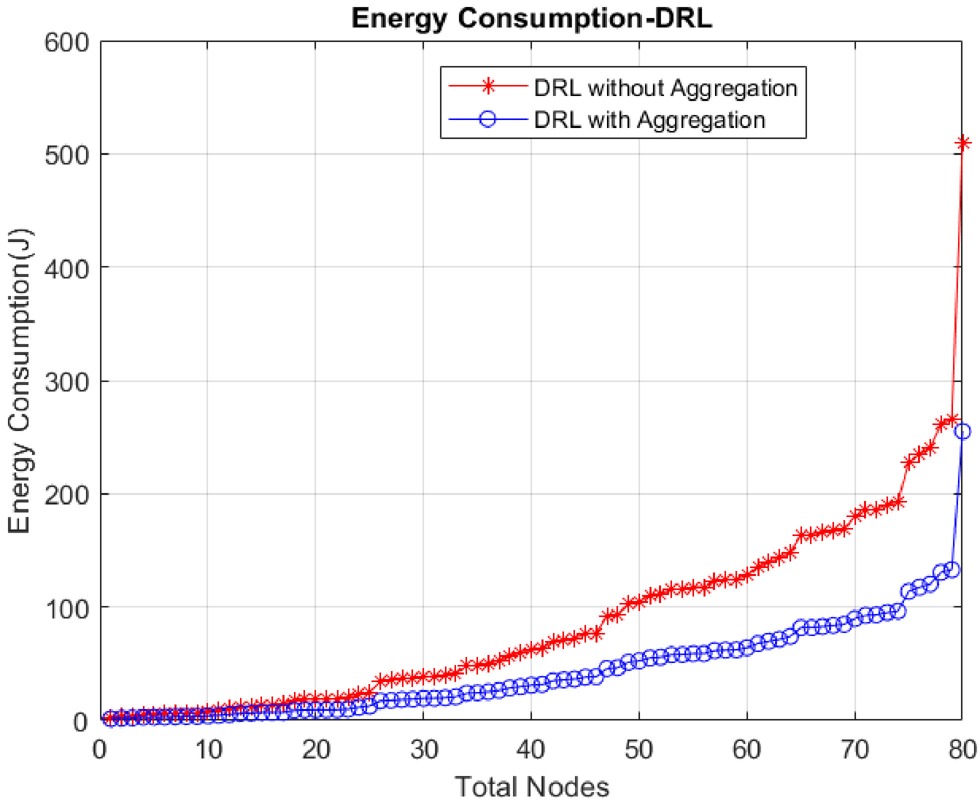


Figure 6.3: Energy Consumption of DRL-OLSR with and without aggregation

500J when there are 80 nodes in total, whereas the DRL-OLSR with data aggregation has consumed only approximately 250J. This graph denotes that the DRL-OLSR with data aggregation consumes significantly less energy when the no. of nodes is increased.

6.3 Summary

The presented analysis involves three distinct routing scenarios aimed at evaluating energy consumption patterns. In the first scenario, a comparison is drawn between Optimized Link State Routing (OLSR) implementations without reinforcement learning and without data aggregation (WO-RL-WO-DA) and with data aggregation (WO-RL-W-DA). Notably, energy consumption becomes significant at around 60 nodes, and a pronounced increase occurs as the node count rises. Particularly, the energy consumption of WO-RL-WO-DA surpasses that of WO-RL-W-DA at 70 nodes, with the divergence becoming most pronounced at 80 nodes, where data aggregation reduces energy consumption by half. This underscores the energy efficiency of WO-RL-W-DA. In the second scenario, the focus shifts to comparing energy use between Self-Organizing Maps Based-Optimized Link State Routing

(SOM-OLSR) without and with data aggregation. A similar trend is observed, where energy consumption becomes noticeable at approximately 60 nodes, and the energy-efficient advantage of data aggregation becomes evident at 70 nodes, with the disparity becoming most pronounced at 80 nodes.

Finally, in the third scenario involving Deep Reinforcement Learning Based-Optimized Link State Routing (DRL-OLSR), energy consumption commences after surpassing 10 nodes, and as node count rises, DRL-OLSR with data aggregation consistently outperforms its non-aggregated counterpart in terms of energy efficiency. These findings collectively highlight the advantageous impact of data aggregation on reducing energy consumption across different routing scenarios and the potential of DRL-OLSR with data aggregation to optimize energy usage as the network size grows.

Chapter 7

Conclusion and Future Directions

The evaluation results demonstrate the significant effectiveness of the proposed deep learning approach in enhancing network lifetime and energy efficiency. Further details on these findings are discussed below.

7.1 Conclusion

In conclusion, the performance evaluation comparing the use of Self-Organizing Maps (SOM), Deep Reinforcement Learning (DRL), and a non-optimized approach revealed that DRL outperformed both SOM and the non-optimized method in terms of energy consumption in the network. Energy optimization has always been a significant challenge in forming wireless sensor networks. The presence of mobile nodes leads to irregular changes in nearby nodes' distance and positions, further complicating the operation of maintaining network connectivity. As a result, addressing these issues becomes critical for efficient and sustainable operation. Based on the results obtained,

it is evident that network connectivity in mobile sensor networks can be enhanced up to a certain level while still maintaining optimal energy usage. The performance evaluation conducted across various metrics, including end-to-end delay, overhead consumption, throughput, and energy consumption, provides valuable insights into the efficacy of different optimization techniques for mobile sensor networks. The comparison among approaches—Without Optimization, Self-Organizing Maps (SOM), and Deep Reinforcement Learning (DRL)—reveals significant improvements in network performance achieved through optimization. While SOM demonstrates moderate enhancements over the baseline approach without optimization, DRL emerges as the most promising method, showcasing superior performance across all evaluated metrics. Specifically, DRL not only effectively reduces overhead consumption and energy consumption but also enhances data transmission efficiency and throughput. These results underscore the potential of advanced optimization techniques, particularly DRL, in addressing the challenges of energy efficiency and network performance in mobile sensor networks.

The findings highlight the critical role of optimization techniques in mitigating energy consumption and improving overall network performance in mobile sensor networks. The superior performance of DRL, as evidenced by its ability to achieve significant reductions in overhead consumption and energy consumption while maintaining high throughput and data transmission efficiency, underscores its effectiveness as a robust optimization approach. By leveraging advanced machine learning techniques like DRL, researchers and practitioners can enhance the sustainability and efficiency of mobile sensor networks, paving the way for the development of more resilient and resource-efficient wireless communication systems.

To further improve the performance, the aggregation method is also used. Data aggregation consistently proves beneficial in reducing energy consumption across scenarios, with the most significant impact observed at higher node counts. Notably, Deep Reinforcement Learning-Based Optimized Link State Routing (DRL-OLSR) with data aggregation consistently outperforms its non-aggregated counterpart, showcasing its potential for optimizing energy usage as the network scales. These findings emphasize the importance of data aggregation and highlight the energy-efficient advantages of DRL-OLSR in various routing scenarios.

While the integration of Self-Organizing Maps (SOM) with the Optimized Link State Routing (OLSR) protocol and Deep Reinforcement Learning (DRL) with OLSR offer promising solutions for enhancing routing efficiency and network performance, they have some limitations. One limitation of SOM-OLSR is its reliance on predefined clustering algorithms, which may not always accurately capture the dynamic nature of network topologies in real time. Additionally, SOM may struggle to scale effectively in large networks, leading to increased computational complexity and overhead. On the other hand, DRL-OLSR may face challenges related to training complexity and convergence time, especially in highly dynamic network environments. Moreover, DRL algorithms require extensive training data and computational resources, which may not be feasible in resource-constrained network scenarios. These limitations underscore the need for further research to optimize and fine-tune the integration of SOM and DRL with OLSR, ensuring their practical applicability and scalability in real-world network deployments.

7.2 Future Scope

Future work in this area could focus on addressing the limitations of the SOM-OLSR and DRL-OLSR integrations to enhance their practical applicability in real-world network deployments. This could involve exploring novel approaches to improve the scalability and adaptability of SOM-based clustering algorithms within the OLSR framework, such as dynamic clustering techniques or hybrid models that combine SOM with other machine learning methods. Additionally, further research could investigate methods to reduce the training complexity and convergence time of DRL algorithms for integration with OLSR, such as exploring more efficient training strategies or leveraging advanced reinforcement learning techniques. By addressing these challenges, future work can advance the integration of SOM and DRL with OLSR, enabling more effective and scalable routing solutions for dynamic network environments.

Several avenues can be explored to enhance further energy consumption optimization in mobile sensor networks using DRL and SOM. These potential directions can contribute to achieving even better performance and efficiency in managing network energy resources. Developing and exploring more advanced DRL architectures could improve energy optimization in mobile sensor networks. Architectures such as Deep Deterministic Policy Gradients (DDPG), Proximal Policy Optimization (PPO), or Trust Region Policy Optimization (TRPO) can be investigated to enhance the learning and decision-making capabilities of the DRL agent.

Combining the strengths of different techniques can lead to further improvements. Hybrid approaches that combine DRL and SOM, or integrate DRL with other optimization algorithms, could potentially achieve better energy consumption optimization. These hybrid models could leverage the SOM's ability to identify clusters and patterns while benefiting from DRL's adaptive decision-making capabilities. Exploring adaptive learning rate mechanisms can help fine-tune the learning process of DRL models. Adaptive learning rates can dynamically adjust the learning rate based on the network's current state, improving convergence speed and optimizing energy consumption.

In future work, introducing contextual awareness to DRL and SOM models can provide valuable information to optimize energy consumption. By considering contextual factors such as network traffic, environmental conditions, or user demands, the models can adapt their decision-making process accordingly, leading to more efficient energy utilization. In addition, investigating transfer learning techniques can enable the knowledge gained from one mobile sensor network to be transferred and applied to a different network with similar characteristics. Transfer learning can expedite the learning process, reduce training time, and optimize energy consumption in new or changing network environments.

Expanding the scope of energy optimization to include edge computing and distributed optimization techniques can further enhance the efficiency of mobile sensor networks. By leveraging edge computing capabilities and distributed optimization algorithms, the energy consumption can be optimized at both the individual node level and the network level. Also, exploring real-time adaptation mechanisms can enable the DRL and SOM models to dynamically adjust their policies based on changing network conditions. By continuously monitoring energy levels, network traffic, and other relevant parameters, the models can adapt their decision-making process in real-time, ensuring optimal energy consumption at all times.

In the proposed thesis work, only free space propagation has been considered. The analysis can be expanded even more in future work by considering the impact of multipath fading and interference between neighboring nodes. These elements can increase the need for higher transmit power to achieve the appropriate levels. For the necessary levels of signal-to-interference noise ratio, these factors may lead to a demand for increased transmit power. In addition, future energy requirements can also be determined in future work by analyzing mobility profiles with swarm intelligent optimization tools.

Bibliography

Abo-Zahhad, Mohammed et al. (2016). "A centralized immune-Voronoi deployment algorithm for coverage maximization and energy conservation in mobile wireless sensor networks". In: *Information Fusion* 30, pp. 36–51.

Ahmad, Awais et al. (2015). "Power aware mobility management of M2M for IoT communications". In: *Mobile Information Systems* 2015, pp. 1–14.

Ahmed Elsmany, Eyman Fathelrhman et al. (2019). "EESRA: Energy Efficient Scalable Routing Algorithm for Wireless Sensor Networks". In: *IEEE Access* 7, pp. 96974–96983. DOI: 10.1109/ACCESS.2019.2929578.

Akbar, Muhammad Sajjad, Hongnian Yu, and Shuang Cang (2016). "TMP: Telemedicine protocol for slotted 802.15. 4 with duty-cycle optimization in wireless body area sensor networks". In: *IEEE Sensors Journal* 17.6, pp. 1925–1936.

Akyildiz, Ian F et al. (2002). "Wireless sensor networks: a survey". In: *Computer networks* 38.4, pp. 393–422.

Almesaeed, Reham and Ahmed Jedidi (2021). "Dynamic directional routing for mobile wireless sensor networks". In: *Ad Hoc Networks* 110, p. 102301.

Alsalmi, N., K. Navaie, and H. Rahmani (May 2024). "Energy and throughput efficient mobile wireless sensor networks: A deep reinforcement learning approach". In: *IET Networks*, n/a–n/a. DOI: 10.1049/ntw2.12126.

Alsheikh, M. Abu et al. (June 2016). "Rate-Distortion Balanced Data Compression for Wireless Sensor Networks". In: *IEEE Sensors Journal* 16.12, pp. 5072–5083.

Alsheikh, Mohammad Abu et al. (2014). "Machine learning in wireless sensor networks: Algorithms, strategies, and applications". In: *IEEE Communications Surveys & Tutorials* 16.4, pp. 1996–2018.

Alzubaidi, Laith et al. (2021). "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions". In: *Journal of big Data* 8, pp. 1–74.

Ambigavathi, M and D Sridharan (2018). "Energy-Aware Data Aggregation Techniques in Wireless Sensor Network". In: *Advances in Power Systems and Energy Management*. Springer, pp. 165–173.

Asif, M. et al. (2017). "Quality of Service of Routing Protocols in Wireless Sensor Networks: A Review". In: *IEEE Access* 5, pp. 1846–1871.

Ayodele, Taiwo Oladipupo (2010). *Introduction to machine learning*. 1st ed. InTech.

Banihashemian, Seyed Saber, Fazlollah Adibnia, and Mehdi A Sarram (2018). "A New Range-Free and Storage-Efficient Localization Algorithm Using Neural Networks in Wireless Sensor Networks". In: *Wireless Personal Communications* 98.1, pp. 1547–1568.

Banoth, Sanjai Prasada Rao, Praveen Kumar Donta, and Tarachand Amgoth (2021). "Dynamic mobile charger scheduling with partial charging strategy for WSNs using deep-Q-networks". In: *Neural Computing and Applications*, pp. 1–13.

— (May 2023). "Target-aware distributed coverage and connectivity algorithm for wireless sensor networks". en. In: *Wirel. Netw.* 29.4, pp. 1815–1830.

Bengio, Yoshua, Aaron Courville, and Pascal Vincent (2013). "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8, pp. 1798–1828.

Bengio, Yoshua, Pascal Lamblin, et al. (2006). "Greedy layer-wise training of deep networks". In: *Advances in neural information processing systems* 19.

Bhasgi, Shivani S and Sujatha Terdal (2021). "Energy and target coverage aware technique for mobile sink based wireless sensor networks with duty cycling". In: *International Journal of Information Technology* 13.6, pp. 2331–2343.

Biradar, Rajashree V et al. (2009). "Classification and comparison of routing protocols in wireless sensor networks". In: *Special Issue on Ubiquitous Computing Security Systems* 4.2, pp. 704–711.

Cao, Li, Yong Cai, and Yinggao Yue (2019). "Swarm intelligence-based performance optimization for mobile wireless sensor networks: survey, challenges, and future directions". In: *IEEE Access* 7, pp. 161524–161553.

Carrano, R. C. et al. (2014). "Survey and Taxonomy of Duty Cycling Mechanisms in Wireless Sensor Networks". In: *IEEE Communications Surveys Tutorials* 16.1, pp. 181–194. DOI: 10.1109/SURV.2013.052213.00116.

Chanak, Prasenjit and Indrajit Banerjee (2016). "Fuzzy rule-based faulty node classification and management scheme for large scale wireless sensor networks". In: *Expert Systems with Applications* 45, pp. 307–321.

Chang, Yuchao, Wen Chen, et al. (2023). *Collaborative Multi-BS Power Management for Dense Radio Access Network using Deep Reinforcement Learning*. arXiv: 2304.07976 [eess.SP].

Chang, Yuchao, Hongying Tang, et al. (July 2017). "Dynamic hierarchical energy-efficient method based on combinatorial optimization for wireless sensor networks". en. In: *Sensors (Basel)* 17.7, p. 1665.

Chang, Yuchao, Xiaobing Yuan, et al. (2019). "Machine-learning-based parallel genetic algorithms for multi-objective optimization in ultra-reliable low-latency WSNs". In: *IEEE Access* 7, pp. 4913–4926.

Chaudhary, Rajesh and Dr Sonia Vatta (2014). "A Tutorial of routing protocols in wireless sensor networks". In: *International Journal of Computer Science and Mobile Computing* 3.6, pp. 971–979.

Chen, Fei, Zhongguang Fu, and Zhiling Yang (2019). "Wind power generation fault diagnosis based on deep learning model in internet of things (IoT) with clusters". In: *Cluster Computing*, pp. 1–13.

Chen, Li et al. (2018). "Auto: Scaling deep reinforcement learning for datacenter-scale automatic traffic optimization". In: *Proceedings of the 2018 conference of the ACM special interest group on data communication*, pp. 191–205.

Chen, Quan, Hong Gao, Zhipeng Cai, et al. (2018). "Distributed low-latency data aggregation for duty-cycle wireless sensor networks". In: *IEEE/ACM Transactions on Networking* 26.5, pp. 2347–2360.

Chen, Quan, Hong Gao, Siyao Cheng, et al. (2017). "Distributed non-structure based data aggregation for duty-cycle wireless sensor networks". In: *IEEE INFOCOM 2017-IEEE Conference on Computer Communications.* IEEE, pp. 1–9.

CHOKSI, CHINTAN and BIJON DESAI (2012). "Comparison of routing protocols in wireless sensor networks". In: *International Conference on Computer Science and Information Technology*, pp. 17–23.

Cottone, Pietro et al. (2016). "A machine learning approach for user localization exploiting connectivity data". In: *Engineering Applications of Artificial Intelligence* 50, pp. 125–134.

Deepakraj, D. and K. Raja (2021). "Hybrid Data Aggregation Algorithm for Energy Efficient Wireless Sensor Networks". In: *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India* 10. Pp. 7–12.

Donta, Praveen Kumar, Tarachand Amgoth, and Chandra Sekhara Rao Annavarapu (Jan. 2022). "Delay-aware data fusion in duty-cycled wireless sensor networks: A Q-learning approach". en. In: *Sustain. Comput. Inform. Syst.* 33.100642, p. 100642.

Donta, Praveen Kumar, Satish Narayana Srirama, et al. (Mar. 2023). "iCoCoA: intelligent congestion control algorithm for CoAP using deep reinforcement learning". en. In: *J. Ambient Intell. Humaniz. Comput.* 14.3, pp. 2951–2966.

Dwivedi, AK and OP Vyas (2010). "Network layer protocols for wireless sensor networks: existing classifications and design challenges". In: *International Journal of Computer Applications* 8.12, pp. 30–34.

Echoukairi, Hassan, Khalid Bourgba, and Mohammed Ouzzif (2016). "A survey on flat routing protocols in wireless sensor networks". In: *Advances in Ubiquitous Networking: Proceedings of the UNet'15 1.* Springer, pp. 311–324.

El, Mohamed Ezz El Dien Abd, Aliaa AA Youssif, Atef Zaki Ghalwash, et al. (2016). "Energy aware and adaptive cross-layer scheme for video transmission over wireless sensor networks". In: *IEEE Sensors Journal* 16.21, pp. 7792–7802.

Elghazel, Wiem et al. (2015). "Random forests for industrial device functioning diagnostics using wireless sensor networks". In: *Aerospace Conference, 2015 IEEE.* IEEE, pp. 1–9.

Elhoseny, Mohamed et al. (2017). "K-coverage model based on genetic algorithm to extend WSN lifetime". In: *IEEE sensors letters* 1.4, pp. 1–4.

Fadlullah, Zubair Md et al. (2017). "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems". In: *IEEE Communications Surveys & Tutorials* 19.4, pp. 2432–2455.

Fang, Wei et al. (2018). "Novel efficient deployment schemes for sensor coverage in mobile wireless sensor networks". In: *Information Fusion* 41, pp. 25–36.

Fei, Zesong et al. (2017). "A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems". In: *IEEE Commun. Surv. Tutor.* 19.1, pp. 550–586.

Fuhao, Zhang and Liu Jiping (2009). "An Algorithm of Shortest Path Based on Dijkstra for Huge Data". In: *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*. Vol. 4, pp. 244–247. DOI: 10.1109/FSKD.2009.848.

Gharajeh, Mohammad Samadi and Sohrab Khanmohammadi (2016). "DFRTP: Dynamic 3D Fuzzy Routing Based on Traffic Probability in Wireless Sensor Networks". In: *IET Wireless Sensor Systems* 6.6, pp. 211–219.

Hammoudeh, Mohammad and Robert Newman (2015). "Adaptive routing in wireless sensor networks: QoS optimisation for enhanced application performance". In: *Information Fusion* 22, pp. 3–15.

He, Ying, Chengchao Liang, et al. (2017). "Optimization of cache-enabled opportunistic interference alignment wireless networks: A big data deep reinforcement learning approach". In: *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6.

He, Ying, Zheng Zhang, et al. (2017). "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks". In: *IEEE Transactions on Vehicular Technology* 66.11, pp. 10433–10445.

Hinton, Geoffrey E, Simon Osindero, and Yee-Whye Teh (2006). "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7, pp. 1527–1554.

Intanagonwiwat, Chalermek, Ramesh Govindan, and Deborah Estrin (2000). "Directed diffusion: A scalable and robust communication paradigm for sensor networks". In: *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 56–67.

Jagannath, Jithin et al. (2019). "Machine learning for wireless communications in the Internet of Things: A comprehensive survey". In: *Ad Hoc Networks* 93, p. 101913.

Kaganurmath, Sharada and KC Ganashree (2016). "Evaluation of Routing Protocols for Wireless Sensor Networks". In: *International Journal Computer and Communication Technology* 2.6.

Kanjo, Eiman, Eman M.G. Younis, and Nasser Sherkat (2018). "Towards unravelling the relationship between on-body, environmental and emotion data using sensor information fusion approach". In: *Information Fusion* 40, pp. 18–31. ISSN: 1566-2535.

Kaur, N., D. R. Kaur, and D. Reecha Sharma (2022). "A Systematic Review of Data Aggregation using Machine Learning Techniques". In: *2022 3rd*

International Conference on Computing, Analytics and Networks (ICAN), Rajpura, Punjab, India 10. Pp. 1–6.

Kaur, R., J. Kaur Sandhu, and L. Sapra (2020). "Machine Learning Technique for Wireless Sensor Networks". In: *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, India* 10. Pp. 332–335.

Kazemeyni, Fatemeh et al. (2014). "Formal Modeling and Analysis of Learning-Based Routing in Mobile Wireless Sensor Networks". In: *Integration of Reusable Systems*. Springer, pp. 127–150.

Ketshabetswe, Lucia Keleadile et al. (2019). "Communication protocols for wireless sensor networks: A survey and comparison". In: *Heliyon* 5.5, e01591.

Kokilavani, S., N. Sathish Kumar, and A. S. Narmadha (2022). "Study and Analysis of Energy Efficient Data Aggregation Techniques for Wireless Sensor Networks". In: *2022 IEEE International Conference on Data Science and Information System (ICDSIS), Hassan, India* 10. Pp. 1–5.

Krishna, M. Bala and N. Vashishta (2013). "Energy efficient data aggregation techniques in wireless sensor networks". In: *5th International Conference and Computational Intelligence and Communication Networks, Mathura, India* 10. Pp. 160–165.

Kuila, Pratyay and Prasanta K Jana (2014). "Energy efficient clustering and routing algorithms for wireless sensor networks: Particle swarm optimization approach". In: *Engineering Applications of Artificial Intelligence* 33, pp. 127–140.

Kumar, Arun et al. (2017). "Location-based routing protocols for wireless sensor networks: A survey". In: *Wireless Sensor Network* 9.1, pp. 25–72.

Kumar, D Praveen, Tarachand Amgoth, and Chandra Sekhara Rao Annavarapu (2019). "Machine learning algorithms for wireless sensor networks: A survey". In: *Information Fusion* 49, pp. 1–25.

Kumar, Prabhat, MP Singh, and US Triar (2012). "A review of routing protocols in wireless sensor network". In: *International Journal of Engineering Research & Technology (IJERT)* 1.4, pp. 1–14.

Kuriakose, Jeril et al. (2014). "A review on localization in wireless sensor networks". In: *Advances in signal processing and intelligent recognition systems.* Springer, pp. 599–610.

Langley, Pat and Herbert A. Simon (Nov. 1995). "Applications of Machine Learning and Rule Induction". In: *Communications of the ACM* 38.11, pp. 54–64.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *nature* 521.7553, pp. 436–444.

Lee, YangMin (2017). "Classification of node degree based on deep learning and routing method applied for virtual route assignment". In: *Ad Hoc Networks* 58, pp. 70–85.

Li, Chunlin et al. (2015). "Distributed data mining based on deep neural network for wireless sensor network". In: *International Journal of Distributed Sensor Networks* 11.7, pp. 1–7.

Li, H., Z. Wan, and H. He (2020). "A Deep Reinforcement Learning Based Approach for Home Energy Management System". In: *2020 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA* 10. Pp. 1–5.

Li, Hongbing et al. (2017). "IM^{2}DCA: Immune mechanism based multipath decoupling connectivity algorithm with fault tolerance under coverage optimization in wireless sensor networks". In: *Applied Soft Computing* 58, pp. 540–552.

Li, Yuxi (2017). "Deep reinforcement learning: An overview". In: *arXiv preprint arXiv:1701.07274*.

Liang, Jiale et al. (2021). "Adaptive cooperative routing transmission for energy heterogeneous wireless sensor networks". In: *Physical Communication* 49, p. 101460. ISSN: 1874-4907. DOI: `https://doi.org/10.1016/j.phycom.2021.101460`.

Lin, Haifeng, Di Bai, and Yunfei Liu (2017). "Maximum data collection rate routing for data gather trees with data aggregation in rechargeable wireless sensor networks". In: *Cluster Computing*, pp. 1–11.

Liu, Bo et al. (2017). "Deep Neural Networks for High Dimension, Low Sample Size Data." In: *IJCAI*, pp. 2287–2293.

Liu, Feng et al. (2017). "A distributed routing algorithm for data collection in low-duty-cycle wireless sensor networks". In: *IEEE Internet of Things Journal* 4.5, pp. 1420–1433.

Liu, X. (2017). "Routing Protocols Based on Ant Colony Optimization in Wireless Sensor Networks: A Survey". In: *IEEE Access* 5, pp. 26303–26317.

Liu, Y. et al. (May 2020). "DeePGA: A Privacy-Preserving Data Aggregation Game in Crowdsensing via Deep Reinforcement Learning". In: *IEEE Internet of Things Journal*. Vol. 7. 5. doi: 10, pp. 4113–4127.

Ma, Tao et al. (2016). "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks". In: *Sensors* 16.10, pp. 1–23.

Mahdavinejad, Mohammad Saeid et al. (2018). "Machine learning for Internet of Things data analysis: A survey". In: *Digital Communications and Networks* 4.3, pp. 161–175.

Mamun, Quazi (2012). "A qualitative comparison of different logical topologies for wireless sensor networks". In: *Sensors* 12.11, pp. 14887–14913.

Mao, Bomin et al. (2017a). "A tensor based deep learning technique for intelligent packet routing". In: *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, pp. 1–6.

— (2017b). "Routing or computing? The paradigm shift towards intelligent computer network packet transmission based on deep learning". In: *IEEE Transactions on Computers* 66.11, pp. 1946–1960.

Marblestone, Adam H, Greg Wayne, and Konrad P Kording (2016). “Toward an integration of deep learning and neuroscience”. In: *Frontiers in computational neuroscience* 10, pp. 1–41.

Mehmood, Amjad et al. (2017). “ELDC: An Artificial Neural Network based Energy-Efficient and Robust Routing Scheme for Pollution Monitoring in WSNs”. In: *IEEE Transactions on Emerging Topics in Computing* PP.99, pp. 1–8.

Mehta, Komal and Raju Pal (2017). “Energy efficient routing protocols for wireless sensor networks: A survey”. In: *International Journal of Computer Applications* 165.3, pp. 41–46.

Mitchell, Thomas M. (1997). *Machine Learning.* 1st ed. New York, NY, USA: McGraw-Hill, Inc. ISBN: 0070428077, 9780070428072.

Mohamed, Shaimaa M., Haitham S. Hamza, and Iman Aly Saroit (Sept. 2017). “Coverage in Mobile Wireless Sensor Networks (M-WSN): A survey”. In: *Computer Communication* 110.C, pp. 133–150. ISSN: 0140-3664.

Morgan, Nelson (2011). “Deep and wide: Multiple layers in automatic speech recognition”. In: *Ieee transactions on audio, speech, and language processing* 20.1, pp. 7–13.

Mouiz, Anas et al. (2019). “Performance evaluation of OLSR and AODV routing protocols with different parameters in mobile ad-hoc networks using NS2 simulator”. In: *Proceedings of the 2019 5th international conference on computer and technology applications*, pp. 134–139.

Pantazis, N. A., S. A. Nikolidakis, and D. D. Vergados (2013). “Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey”. In: *IEEE Communications Surveys Tutorials* 15.2, pp. 551–591. DOI: `10.1109/SURV.2012.062612.00084`.

Pantazis, Nikolaos A, Stefanos A Nikolidakis, and Dimitrios D Vergados (2012). “Energy-efficient routing protocols in wireless sensor networks: A survey”. In: *IEEE Communications surveys & tutorials* 15.2, pp. 551–591.

Philip Paul Arunodhayam, T et al. (2023). “Sustainable Energy Efficient Routing Protocol using Chimp algorithm and SOM for Underwater Wireless Sensor Networks”. In: *2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET)*, pp. 1–4. DOI: `10.1109/ICRASET59632.2023.10420186`.

Phoemphon, Songyut, Chakchai So-In, and Dusit Tao Niyato (2018). “A Hybrid Model using Fuzzy Logic and an Extreme Learning Machine with Vector Particle Swarm Optimization for Wireless Sensor Network Localization”. In: *Applied Soft Computing* 65, pp. 101–120.

Poole, David L and Alan K Mackworth (2010). *Artificial Intelligence: Foundations of computational agents.* Cambridge University Press.

Praveen Kumar, D., Amgoth Tarachand, and Annavarapu Chandra Sekhara Rao (2019). “Machine learning algorithms for wireless sensor networks: A survey”. In: *Information Fusion* 49, pp. 1–25. ISSN: 1566-2535.

Qin, Jiahu et al. (2017). “Distributed k-Means Algorithm and Fuzzy c-Means Algorithm for Sensor Networks Based on Multiagent Consensus Theory”. In: *IEEE transactions on cybernetics* 47.3, pp. 772–783.

Radi, Marjan et al. (2012). “Multipath routing in wireless sensor networks: survey and research challenges”. In: *sensors* 12.1, pp. 650–685.

Ramasamy, Velmani (2017). “Mobile wireless sensor networks: An overview”. In: *Wireless Sensor Networks—Insights and Innovations*.

Rawat, Priyanka et al. (Apr. 2014). “Wireless sensor networks: a survey on recent developments and potential synergies”. In: *The Journal of Supercomputing* 68.1, pp. 1–48. ISSN: 1573-0484.

Ray, Anindita and Debashis De (2016). “Energy efficient clustering protocol based on K-means (EECPK-means)-midpoint algorithm for enhanced network lifetime in wireless sensor network”. In: *IET Wireless Sensor Systems* 6.6, pp. 181–191.

Rezazadeh, Javad, Marjan Moradi, and Abdul Samad Ismail (2012). “Mobile wireless sensor networks overview”. In: *International Journal of Computer Communications and Networks* 2.1, pp. 17–22.

Sabor, Nabil et al. (2017). “A comprehensive survey on hierarchical-based routing protocols for mobile wireless sensor networks: Review, taxonomy, and future directions”. In: *Wireless Communications and Mobile Computing* 2017.

Sah, Dinesh Kumar and Tarachand Amgoth (2018). “Parametric survey on cross-layer designs for wireless sensor networks”. In: *Computer Science Review* 27, pp. 112–134.

Sah, Dinesh Kumar, Korhan Cengiz, et al. (Dec. 2021). “EDGF: Empirical dataset generation framework for wireless sensor networks”. en. In: *Comput. Commun.* 180, pp. 48–56.

Salarian, Hamidreza, Kwan-Wu Chin, and Fazel Naghdy (2014). “An energy-efficient mobile-sink path selection strategy for wireless sensor networks”. In: *IEEE Transactions on vehicular technology* 63.5, pp. 2407–2419.

Sara, Getsy S and D Sridharan (2014). “Routing in mobile wireless sensor network: A survey”. In: *Telecommunication Systems* 57, pp. 51–79.

Sekhar, Pudi et al. (2021). “An effective metaheuristic based node localization technique for wireless sensor networks enabled indoor communication”. In: *Physical Communication* 48, p. 101411. ISSN: 1874-4907. DOI: `https://doi.org/10.1016/j.phycom.2021.101411`.

Sharma, Amandeep and Ajay Kakkar (2018). “Forecasting daily global solar irradiance generation using machine learning”. In: *Renewable and Sustainable Energy Reviews* 82, pp. 2254–2269.

Shashikala and C. Kavitha (2018). *Secure cluster-based data integrity routing for wireless sensor networks*. Singapore: Springer Singapore, pp. 157–167.

Shi, Lei et al. (2015). “An efficient distributed routing protocol for wireless sensor networks with mobile sinks”. In: *International Journal of Communication Systems* 28.11, pp. 1789–1804.

Silva, Ricardo et al. (2011). "Mobility in WSNs for critical applications". In: *2011 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, pp. 451–456.

Singh, Shio Kumar (2010). "Routing protocols in wireless sensor networks-A survey". In: *Int. J. Computer Science & Engineering Survey (IJC-SES)* 1, pp. 63–83.

Sodhro, Ali Hassan et al. (2018). "A joint transmission power control and duty-cycle approach for smart healthcare system". In: *IEEE Sensors Journal* 19.19, pp. 8479–8486.

Song, Xin et al. (2013). "DLRDG: distributed linear regression-based hierarchical data gathering framework in wireless sensor network". In: *Neural Computing and Applications* 23.7-8, pp. 1999–2013.

Srivastava, Nitish, Ruslan R Salakhutdinov, and Geoffrey E Hinton (2013). "Modeling documents with deep boltzmann machines". In: *arXiv preprint arXiv:1309.6865*.

Sudha, C., D. Suresh, and A. Nagesh (2021). "An Enhanced Machine learning data aggregation model for Efficient data processing in wireless sensor networks". In: *2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatre, India* 10. Pp. 1748–1753.

Sun, Wei et al. (2018). "End-to-End Data Delivery Reliability Model for Estimating and Optimizing the Link Quality of Industrial WSNs". In: *IEEE Transactions on Automation Science and Engineering* 15.3, pp. 1127–1137.

Sun, Yongjun, Wenxin Dong, and Yahuan Chen (2017). "An improved routing algorithm based on ant colony optimization in wireless sensor networks". In: *IEEE Communications Letters* 21.6, pp. 1317–1320.

Sutton, Richard S and Andrew G Barto (2018). *Reinforcement learning: An introduction*. MIT press.

Tan, Wilson M et al. (2017). "An indoor test methodology for solar-powered wireless sensor networks". In: *ACM Transactions on Embedded Computing Systems* 16.3, pp. 82.1–82.25.

Tang, Jiaqi, Andrej Mihailovic, and Hamid Aghvami (2022). "Constructing a DRL Decision Making Scheme for Multi-Path Routing in All-IP Access Network". In: *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 3623–3628. DOI: `10.1109/GLOBECOM48099.2022.10000597`.

Velmani, R (n.d.). "Novel topology management schemes for effective energy utilization in large scale mobile wireless sensor networks". In: ().

Wang, F. and J. Liu (2011). "Networked Wireless Sensor Data Collection: Issues, Challenges, and Approaches". In: *IEEE Communications Surveys Tutorials* 13.4, pp. 673–687. DOI: `10.1109/SURV.2011.060710.00066`.

Wang, Tian et al. (2017). "Data collection from WSNs to the cloud based on mobile Fog elements". In: *Future Generation Computer Systems*. DOI: `https://doi.org/10.1016/j.future.2017.07.031`.

Wang, Yuqin et al. (2020). "XgBoosted Neighbor Referring in Low-duty-cycle Wireless Sensor Networks". In: *IEEE Internet of Things Journal.* DOI: `https://doi.org/10.1109/JIOT.2020.3022650`.

Wang, Yuzhi et al. (2017). "A deep learning approach for blind drift calibration of sensor networks". In: *IEEE Sensors Journal* 17.13, pp. 4158–4171.

Wen, Weimin et al. (2018). "EAPC: Energy-Aware Path Construction for Data Collection Using Mobile Sink in Wireless Sensor Networks". In: *IEEE Sensors Journal* 18.2, pp. 890–901.

Wichmann, Andrew and Turgay Korkmaz (2015). "Smooth path construction and adjustment for multiple mobile sinks in wireless sensor networks". In: *Computer Communications* 72, pp. 93–106.

Xiao, Kefan, Shiwen Mao, and Jitendra K Tugnait (2019). "TCP-Drinc: Smart Congestion Control Based on Deep Reinforcement Learning". In: *IEEE Access* 7, pp. 11892–11904.

xie, k. et al. (2017). "Low Cost and High Accuracy Data Gathering in WSNs with Matrix Completion". In: *IEEE Transactions on Mobile Computing* 17.7, pp. 1595–1608.

Xu, Chuan et al. (2019). "An Energy-Efficient Region Source Routing Protocol for Lifetime Maximization in WSN". In: *IEEE Access* 7, pp. 135277–135289. DOI: `10.1109/ACCESS.2019.2942321`.

Yang, Shusen et al. (2016). "Practical opportunistic data collection in wireless sensor networks with mobile sinks". In: *IEEE Transactions on Mobile Computing* 16.5, pp. 1420–1433.

Yetgin, H. et al. (2017). "A Survey of Network Lifetime Maximization Techniques in Wireless Sensor Networks". In: *IEEE Communications Surveys Tutorials* 19.2, pp. 828–854. DOI: `10.1109/COMST.2017.2650979`.

Yick, Jennifer, Biswanath Mukherjee, and Dipak Ghosal (Aug. 2008). "Wireless Sensor Network Survey". In: *Computer Networks* 52.12, pp. 2292–2330. ISSN: 1389-1286.

Yoo, Jaehyun, Woojin Kim, and Hyoun Jin Kim (2015). "Distributed estimation using online semi-supervised particle filter for mobile sensor networks". In: *IET Control Theory & Applications* 9.3, pp. 418–427.

Zhang, F. and Q. Yang (2020). "Energy Trading in Smart Grid: A Deep Reinforcement Learning-based Approach". In: *2020 Chinese Control And Decision Conference (CCDC), Hefei, China* 10. Pp. 3677–3682.

Zhao, Bin and Xiaohui Zhao (2022). "Deep Reinforcement Learning Resource Allocation in Wireless Sensor Networks With Energy Harvesting and Relay". In: *IEEE Internet of Things Journal* 9.3, pp. 2330–2345. DOI: `10.1109/JIOT.2021.3094465`.

Zhao, Huan et al. (2015). "Energy-efficient topology control algorithm for maximizing network lifetime in wireless sensor networks with mobile sink". In: *Applied Soft Computing* 34, pp. 539–550.

Zhu, Xiaojin and Andrew B Goldberg (2009). "Introduction to semi-supervised learning". In: *Synthesis lectures on artificial intelligence and machine learning* 3.1, pp. 1–130.

Zhu, Z., X. Ye, and L. Fu (2020). "Deep Reinforcement Learning Based Energy Efficient Underwater Acoustic Communications". In: *Global Oceans*, pp. 1–5.