Energy and Throughput Efficient Mobile Wireless Sensor Networks: A Deep Reinforcement Learning Approach

Nada Alsalmi (n.alsalmi@lancaster.ac.uk)[a,b], Keivan Navaie (k.navaie@lancaster.ac.uk)[a], Hossein Rahmani (h.rahmani@lancaster.ac.uk)[a]

[a] *Department of Computing and Communications, Lancaster University LA1, 4WA, UK*
[b] *Computer Science and Engineering, University of Jeddah*

Abstract

Efficient development of Mobile Wireless Sensor Networks (MWSNs) relies heavily on optimizing two key parameters: Throughput and Energy Consumption. The proposed work investigates network connectivity issues with MWSN and proposes 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) for MWSNs. The primary objective of the proposed algorithms is to achieve energy-efficient routing while maximizing throughput. These algorithms take into account the interplay among sensor node deployment, communication radius, and detection area, presenting a novel approach to sustaining communication while optimizing energy consumption. Leveraging deep learning techniques, the algorithms facilitate optimal feature extraction, thereby enhancing overall performance. The proposed algorithms are evaluated through simulations by considering various performance metrics, including connection probability, end-to-end delay, overhead, network throughput, and energy consumption. 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 than the case with 'no routing' optimization. A comparison between DRL-OLSR and SOM-OLSR reveals that the former surpasses the latter in terms of low latency and prolonged network lifetime. Specifically, DRL-OLSR demonstrates a 47% increase in throughput, a 67% reduction in energy consumption, and a connection probability three times higher than SOM-OLSR. Furthermore, when contrasted with the 'No optimization' scenario, DRL-OLSR achieves a remarkable 69.7% higher throughput and nearly 89% lower energy consumption. These findings highlight the effectiveness of the DRL-OLSR approach in optimizing both network performance and energy efficiency in wireless sensor networks.

Keywords: Deep reinforcement learning, mobile sensor network, energy consumption, network lifetime, network throughput

1. Introduction

Wireless Sensor Networks (WSNs) are a crucial aspect of modern technology that connects an enormous number of sensors dispersed in the environment to monitor and manage systems. These networks integrate contemporary technologies such as information and communication to combine sensing and processing capabilities. WSNs are widely used due to their low cost, energy efficiency, vast dissemination, and capacity for self-organization. In fact, the next-generation Internet cannot be developed without WSNs as they enable sensing and actuation capabilities for future applications. Wireless Sensor Networks (WSNs) are crucial enablers for evolving applications such as smart homes [1], smart cities [2], healthcare monitoring [3], surveillance [4], and disaster management systems [5]. A WSN consists of many sensor nodes spread in the area of interest for gathering various categories of data as shown in Figure 1 [6], [7]. 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 [8]. The motivation of the present work is discussed below:

1.1. *Motivation*

In many practical applications, sensor nodes often rely on limited energy sources, such as non-rechargeable batteries (see, for example, [9], and [10]. Consequently, these nodes frequently operate under energy-constrained conditions. Therefore, it is crucial to manage energy consumption judiciously to ensure effective utilization and efficient performance of essential operations, including sensing, processing, and communication [11]. The challenges and limitations of MWSN are discussed in the following.

1.2. *Challenges*

For mobile wireless sensor networks (MWSN), achieving high coverage and connectivity is also challenging [12]. In MWSN, the sensor nodes can move within the network. They differ from traditional WSNs due to their mobility feature that improves network coverage, connectivity, scalability, and energy efficiency while prolonging the network's lifetime [13]. MWSNs can be used for various applications with enhanced connectivity and coverage and with limited computational complexity [14, 15, 16]. Nevertheless, MWSNs face several challenges, including limited energy, memory, and processing capabilities, as well as communication and coordination issues.

1.3. *Possible Solutions and Limitations*

To address these challenges, Machine Learning (ML) techniques can be leveraged in Mobile Wireless Sensor Networks (MWSNs) to enhance their performance and capabilities [17]. Deep Learning (DL), a subset of ML, offers a powerful approach by employing multi-layer Artificial Neural Networks (ANNs) to model and solve complex problems. ANNs mimic the structure of the human brain, comprising interconnected layers of nodes that process and transform data.

For MWSN, the DL algorithms can be used for ANN network training using the existing datasets or historical records. The training can be used for the allocating

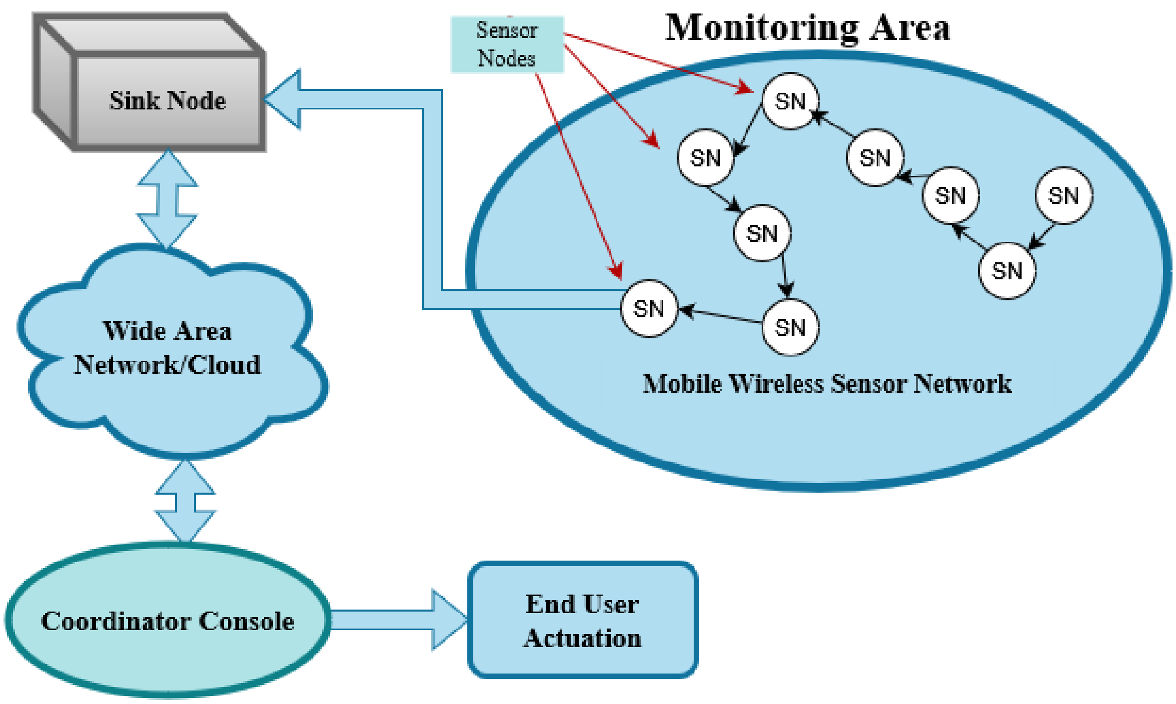

Figure 1: Architecture of MWSN (Mobile Wireless Sensor Network)[7].

medium/channel or routing of the data. There are several ways in which ML and DL 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.

Therefore, the researchers are now working in the field of energy-efficient DL algorithms to further extend the network lifetime [18]. By using DL techniques, it is likely 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 extend the lifetime of the network. In addition, the DL algorithms can also be used for optimization of resource allocation in the network. For example, tasks can be allocated to different nodes based on their available processing power and memory. This ensures that the available resources are used effectively, maximizing the network's overall performance [19].

In some applications, sensor nodes face difficulties in accurately detecting events within their monitoring area.. For such issues also DL algorithms can be used to detect anomalies/events in sensor data, e.g., sudden changes in temperature or humidity. This could cause a potential problem, allowing for appropriate intervention before any major issue arises [20]. For such applications, in some previous works, DL algorithms have been developed which can predict when a sensor node is likely to fail and require maintenance. This enables active maintenance, reduces downtime, and increases the overall network throughput [21].

Such improvements help MWSNs to meet the needs of various applications, making them more useful and reliable. Future research can further improve 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 while maintaining coverage and connectivity without compromising computational complexity [16].

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 [22]. However, to increase the battery life of nodes, it is often necessary to reduce transmit power. In the case of MWSN, probabilistic modelling 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. Mobile nodes are primarily utilized to enhance coverage range and connectivity. The moving agent can collect the data from various nodes and transmit the data to the base station. Nevertheless, the routing process in MWSNs faces several challenges. Firstly, a global addressing process is impractical due to the deployment of many sensor nodes. For MWSN networks, conventional IP-based protocols created for large-scale network infrastructure are often inappropriate. Secondly, most WSNs require a continuous stream of sensed data from multiple sources to a specific sink node or base station, which conflicts with typical communication networks. Thirdly, multiple sensors may generate similar data within the vicinity of a phenomenon, resulting in heavy redundancy traffic across the network [23]. This redundancy consumes more energy and bandwidth, causing various issues such as delay, packet loss, and bandwidth degradation. In the existing OLSR method, the throughput degradation is not addressed [24]. Therefore, the variants of OLSR are proposed, i.e., SOM-OLSR, and DRL-OLSR to deal with above addressed issues.

In this paper, intelligent routing algorithms are developed to address the above challenges. The proposed methods dynamically adjust the data routing paths based on the network congestion status and bandwidth availability. Such algorithms can help to optimize network performance and reduce energy consumption by minimizing data transmission over congested routes. The contributions of this paper are listed in the following.

- The 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) are proposed for energy-efficient data transmission. Both algorithms utilize deep learning techniques to enhance the routing performance in MWSNs. The two methods are compared due to their practical importance in terms of computational complexity, and deployment scenario.

- 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.
- The SOM-OLSR is an unsupervised artificial neural network-based energy-efficient routing protocol that aims to find the optimal path from the sensor node to the sink node. SOM ensures reliable communication by handling noisy and incomplete data, which makes it useful for real-time applications.
- The DRL-OSLR algorithm is a fault-tolerant routing technique designed to maintain robust connectivity in dynamic network topology. By utilizing multiple paths between nodes, the algorithm ensures that data can still be successfully delivered even if one of the paths is disrupted.
- The performance of both algorithms is evaluated and analyzed in terms of various performance metrics through extensive simulations.
- The performance of the proposed methods is also compared with the traditional routing method indicating their significant performance improvement.

The remainder of the paper is organized as follows. Section 2 presents related work on MWSNs, covering recent research in the field. Section 3 outlines the background of the proposed research work. Section 4 outlines the system network architecture and proposes an energy-efficient connectivity technique. In Section 5, the simulation results are presented considering various scenarios and evaluating metrics including connection probability, E2E delay, routing overhead, and network throughput. Section 6 compares the performance of the proposed methods with the existing methods. Finally, Section 7 concludes the paper, summarizing the main findings and discussing future research directions.

2. Related Work

Routing presents a significant challenge in MSNs, given constraints such as limited power supply, low transmission bandwidth, reduced memory capacity, and processing capability. Table 1 summarizes existing works based on DL routing in a nutshell.

Generally, in WSNs, sensor nodes (SNs) that are in close proximity to the base station (BS) tend to consume more energy as they often serve as relay nodes for distant

SNs. This increased energy consumption can lead to premature depletion of energy of the SNs' and ultimately impact the overall network lifetime. Recently, several routing protocols [25, 26, 27] have been developed for WSNs using different approaches. However, very few works were reported on routing protocol using ML techniques. ML techniques have a wide range of applications in WSNs, such as optimal routing, lowering communication overhead, and delay-aware [28]. This section discusses the current research on routing protocols based on ML and DL techniques[29, 30, 31, 32, 33].

Table 1: Literature Review

Ref	**Techniques**	**Outcomes**	**Features**	**Drawback**
[29]	Researchers utilized a DBN, they uncover the correlations between the demand for multi-commodity flow in wireless networks and link usage.	Based on the 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
[31] [30]	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

Table 1: Literature Review

Ref	Techniques	Outcomes	Features	Drawback
[32]	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 centres.	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.
[33]	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 signalling interval of 240 milliseconds.	In this paper, the authors propose a supervised deep learning system that constructs routing tables and demonstrates how it can be seamlessly integrated with programmable routers equipped with CPUs and GPUs.	The paper does not explicitly mention scalability.

Table 1: Literature Review

Ref	Techniques	Outcomes	Features	Drawback
[34]	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 about the routing nodes.	The Viterbi algorithm generates virtual routes based on the classification results and temporary routes.	The technique employs a hybrid wireless ad-hoc network routing solution, leveraging collaboration between wireless ad hoc networks and infrastructure-based wired networks. This approach combines the node degree classifier (NDC) outcomes, generated by deep learning, with the Viterbi algorithm to determine the most efficient route.	Integrating a deep learning-based routing solution with existing network infrastructure and protocols can be challenging.
[35]	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 .

Table 1: Literature Review

Ref	Techniques	Outcomes	Features	Drawback
[36]	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

In [34], 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 packet loss and power management. In [37], a Bayesian learning method-based optimal routing prediction model has been developed for both decentralized and centralized versions. The described approach integrates a scheduling mechanism alongside routing data to achieve balanced energy consumption. This algorithm is particularly well-suited for decentralized systems, offering advantages over centralized counterparts.

Furthermore, [38] used 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 controlling the traffic overhead. In [39], authors have proposed energy efficient clustering protocol using a k-means (EECPK-means) algorithm to find the optimal centre point of the cluster from a random initial centre point. It selects optimal cluster heads (CH)s based on the Euclidean distance

Table 2: Comparative Analysis

Protocol Name	Connectivity	Energy Consumption	Throughput	Delay
LEACH [41]	Moderate	Moderate	High	High
OLSR	Moderate	High	Moderate	High
SOM OLSR	Moderate	Moderate	Moderate	Moderate
D-SOM [24]	Moderate	Optimum	Optimum	Optimum
GHND [42]	Moderate	Moderate	Low	Moderate
IGHND [43]	Moderate	Moderate	Moderate	Moderate

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.

Also in [40], 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.

n previous research studies, researchers have explored the effectiveness of the Self-Organizing Map (SoM) approach for efficient routing.In [24], the Distributed Artificial Intelligence (DAI) and Self Organizing Map (SOM) Hybridized approach is used for energy-efficient routing. The performance is compared with the other protocols [24, 41, 42, 43] as shown in Table 2. While the performance of other D-SoM protocols is generally optimal, they often fall short in terms of network connectivity. Hence, we propose a novel Deep Learning-based OLSR routing method to ensure optimal performance in our research endeavour.

The literature study indicates that the features of DL 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[44]. 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 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. Background

In this section, we briefly present the main concepts used in this work.

3.1. Self-Organizing Maps

Self-organizing maps (SOMs) are a type of unsupervised artificial neural network (ANN). 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. Throughout training, the weights of neurons in a two-dimensional grid are iteratively fine-tuned to align with the input data.

A competitive learning rule is used to compare 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 neighbouring neurons are then updated based on a Gaussian function that decreases with distance from the winner neuron. The SOM process is shown in Figure 2. SOMs can be used for various real-time applications, e.g., network optimization, anomaly detection, fault diagnosis etc.

3.2. Deep Reinforcement Learning

Deep Reinforcement Learning (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 in the form of rewards, and repeatedly 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.

DRL consists of several key components that work together to enable agents to learn and make sequential decisions in complex environments. The components comprise the agent, the environment, the action space, the state space, the reward system, and the learning algorithm. Each component within the DRL framework plays a crucial role and contributes significantly to the overall significance of DRL in tackling complex tasks, as illustrated in Figure 3.

As shown in Figure 3, 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 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. The subsequent component is the environment, which represents the external system with which the agent interacts. It provides the agent with observations or states, accepts the agent's actions, and delivers 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. Similarly, action space and state space define the set of all possible actions and states respectively.

The reward system is used for feedback based on the actions in the environment. For the actions of the learning algorithm, the Q-value will be responsible for updating the agent's decision-making policy based on the feedback received from the environment.

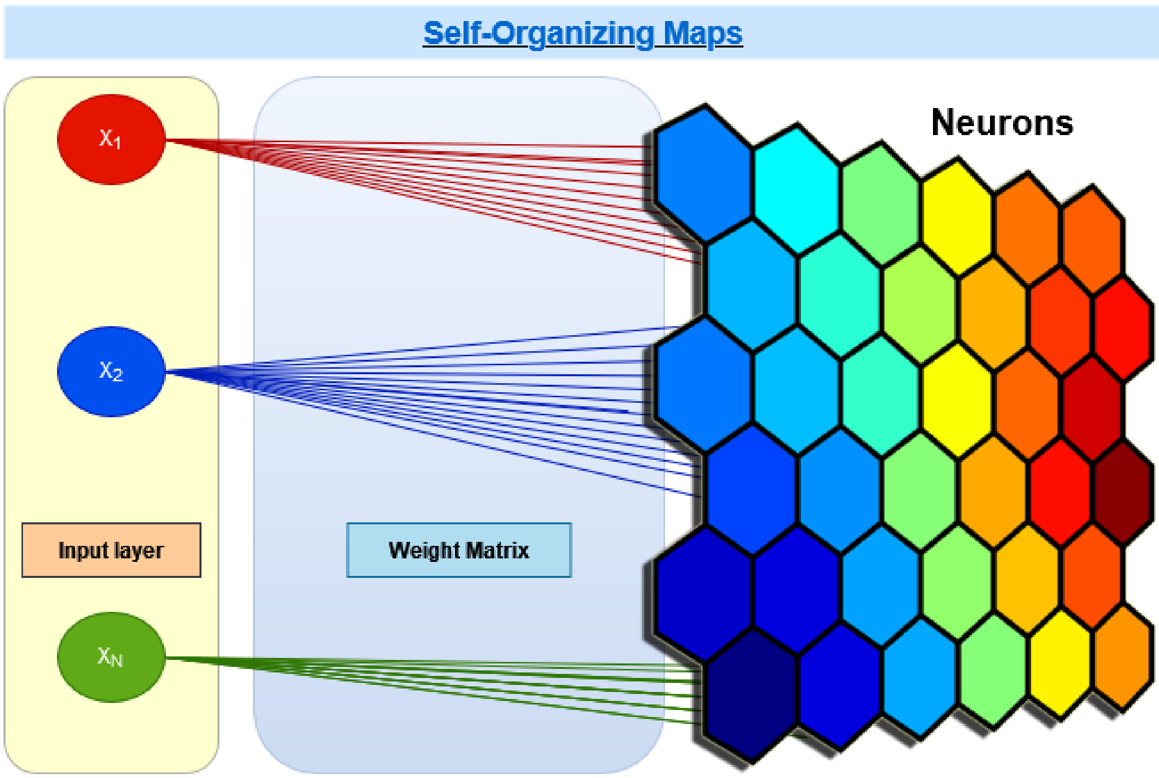

Figure 2: The process of self organizing maps [45].

DRL enables the MWSN to obtain the most energy-efficient routing paths that extend the lifespan of MWSNs that are constantly changing [47]. For such a complex routing problem, the goals can be simplified by breaking them into smaller sub-goals. In each sub-goal, nodes create graph structures by only considering their nearby neighbours, resulting in low-cost, efficient, and real-time routing.

3.3. Routing in MWSN

Routing in MWSNs plays a critical role in ensuring efficient and reliable data transmission in dynamic and resource-constrained environments[48, 49]. Traditional routing protocols often struggle to adapt to the changing network conditions and dynamic node mobility inherent in MWSNs. Various routing protocols for MWSN have been proposed over the last decade, with the most important of them being AODV[50], DSR[51], DSDV[52], and OLSR[53].

The proposed approach outlined in this paper revolves around the Optimized Link State Routing (OLSR) protocol. OLSR is a proactive and table-driven routing protocol designed to enhance routing efficiency. It achieves this by constantly updating routing tables and minimizing the amount of routing overhead required. The primary goal of OLSR is to establish efficient and reliable routes within the network, thereby improving overall network performance. The protocol is specifically designed to handle the routing

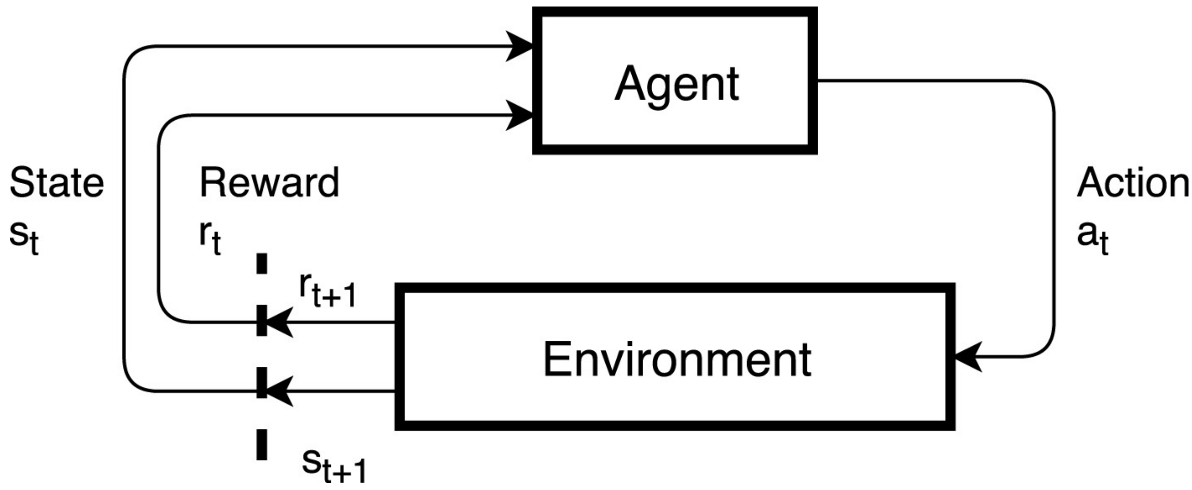

Figure 3: Overview of reinforcement learning [46].

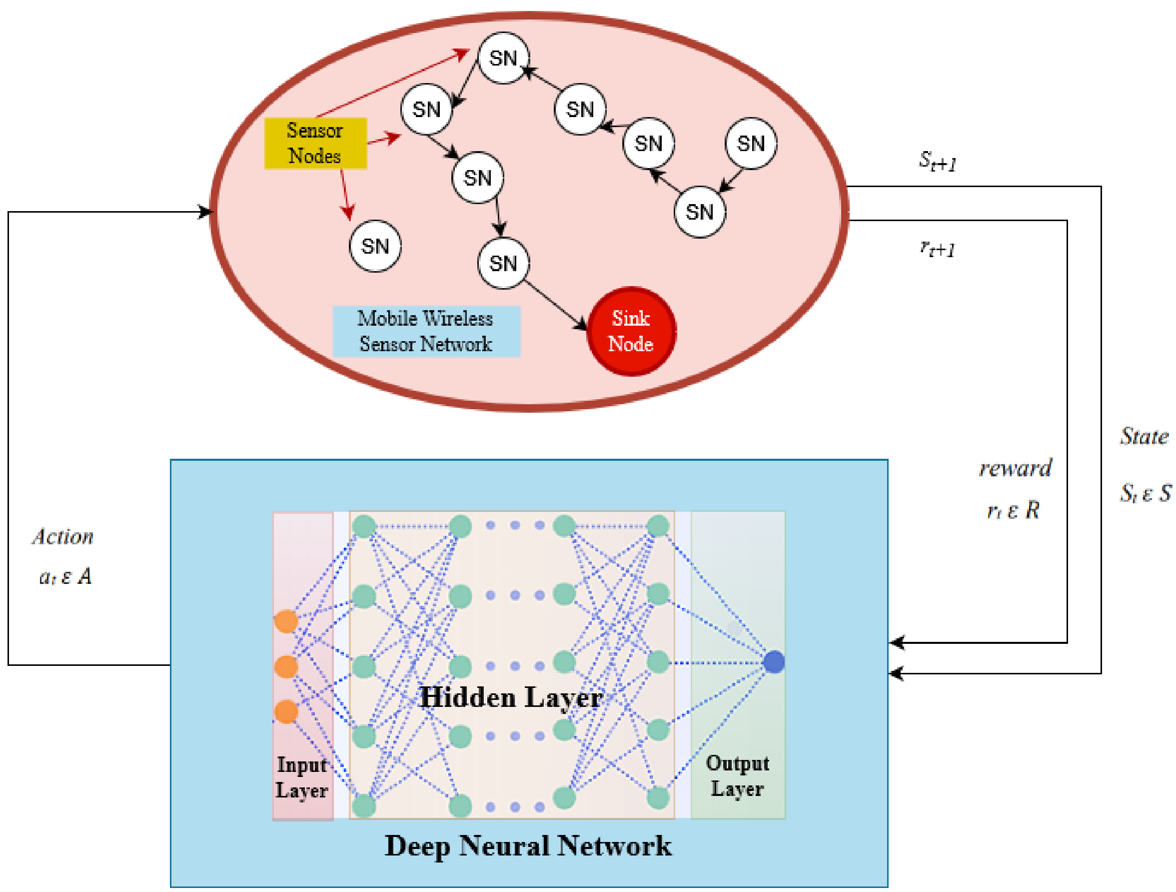

Figure 4: Deep Reinforcement Learning.

of data packets in networks that have a large number of nodes and experience frequent changes in network topology caused by node mobility. Its purpose is to efficiently adapt to these dynamic conditions and ensure reliable data transmission throughout of the network. However, there are still several key challenges associated with OLSR in mobile wireless sensor networks (MWSNs). These challenges include frequent topology changes, routing loop inconsistencies, route stability, limited bandwidth, energy resources, scalability, and control overhead.

To address these challenges, a promising approach is the utilization of deep learning (DL) techniques for routing in MWSNs. One of the key advantages of DL-based routing in MWSNs is its ability to handle node mobility effectively. As nodes move within the network, the topology changes and traditional routing protocols may experience disruptions and suboptimal paths.

Our proposed approaches involve training a deep learning model to optimize routing decisions in OLSR networks by leveraging historical network data and performance metrics [54, 55]. These approaches can 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.

Optimized Link State Routing (OLSR)

The OLSR protocol [56] 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 neighbours and the links to those neighbours. This information is used to update the topology database and to calculate the shortest path between any two nodes in the network. It uses a multipoint relaying (MPR) technique to reduce the number of broadcast messages and to 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 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. The OLSR protocol is crafted to be exceptionally efficient and scalable, rendering it ideal for large-scale MANETs characterized by high mobility and frequent topology alterations. [53].

The performance of OLSR depends upon the lost link. The packets are not forwarded to the lost link but packets are forwarded along the fresh shortest way. One more challenge is scalability as it degrades the performance and efficiency of larger network sizes. This is because OLSR floods topology information throughout the network, which can lead to excessive overhead and congestion. The primary cause of this issue stems from the flooding of topology information in OLSR, resulting in significant overhead and congestion. OLSR's inability to support Quality of Service (QoS) requirements means it cannot differentiate between different types of traffic. Consequently, the network may experience congestion and delays due to this limitation.

4. 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. Optimized Link State Routing (OLSR) is selected for its efficiency in managing link states within mobile wireless sensor networks. In the proposed mechanisms, OLSR serves as the underlying routing protocol that benefits from the insights provided by SOM and the adaptability introduced by DRL. OLSR's ability to dynamically adjust routing decisions based on link-state information makes it relevant to the integration. While the proposed mechanisms might be adaptable to other link-state protocols, the choice of OLSR is grounded in its proven performance in the specific context of wireless networks.

The goal of OLSR is to identify neighbouring nodes within range and utilize OLSR to determine the most efficient route for data transmission. To achieve this, the nodes in the network broadcast periodic 'Hello' messages to discover neighbouring nodes within their communication range. By exchanging Link State Packets (LSPs), nodes gather information about their neighbours 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 [57].

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 the 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 (1).

$$\beta = \frac{D_{packets}}{T_{packets}} \times 100 \tag{1}$$

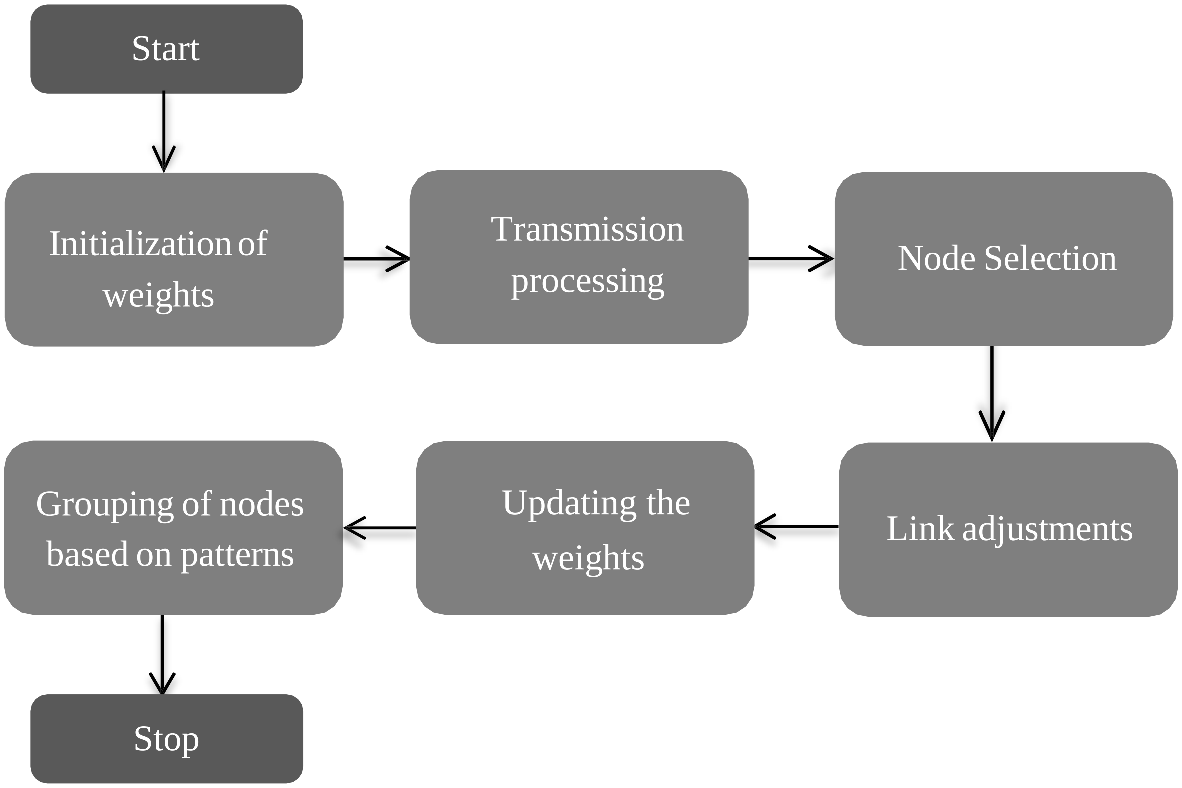

Figure 5: The functional block diagram of SOM.

The routing efficiency depends upon Connection Probability C_p is:

$$C_p(i) = \frac{T(i) \times E(i)}{N \times T_{bps}} \tag{2}$$

where $C_p(i)$ is the Connection Probability of i^{th} node, $E(i)$ is the 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, T_{bps} is the total bit per second taken by the node to transmit the packets.

The energy consumption of the nodes is calculated in terms of the throughput of the network and is given below in (3),

$$E_{cons}(i) = Ov_{cons}(i) \times T_p/N \tag{3}$$

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.

4.1. Routing based on SOM in MWSNs

The SOM-OLSR routing algorithm is proposed, which is based on the SOM algorithm. SOMs, along with the OLSR (Optimized Link State Routing) protocol, are utilized for energy-efficient routing in sensor networks. The combination of SOMs and OLSR aims to find the optimal path for transmitting data from the sensors to the sink node while minimizing energy consumption.

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. 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. The cost function is formulated as

  $$C(d) = k * d \tag{4}$$

  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

  $$C(i, j) = C(d(i, j)) \tag{5}$$

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

We use SOMs and OLSR 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. 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. During training, the SOM adjusts its weight vectors 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

$$W_{Nor}(i,j) = \frac{(W_{i,j} - min_{val})}{(max_{val} - min_{val})} \tag{6}$$

  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:

$$d_i = \sqrt{\sum (x_j - w_i)^2} \tag{7}$$

- 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 neighbouring nodes based on their distance from the selected node.

$$W_{ij}(t+1) = W_{ij}(t) + \eta(t) \times h_{ij}(X_j - W_{ij}(t)) \tag{8}$$

  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 algorithm to ensure that it converges. The neighbourhood function $h_{ij}(t)$ is a Gaussian function that decreases with distance from the winning neuron.

$$h_{ji}(t) = \exp\left(\frac{(c_i(t) - c_j(t))^2}{2\sigma^2(t)}\right) \tag{9}$$

where c_i and c_j are the locations of neurons i and j, and $\sigma(t)$ is the neighborhood radius at iteration t. This process is depicted in Figure 5 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.

Algorithm 1 SOM-OLSR Algorithm

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

- 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. We can associate each region 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

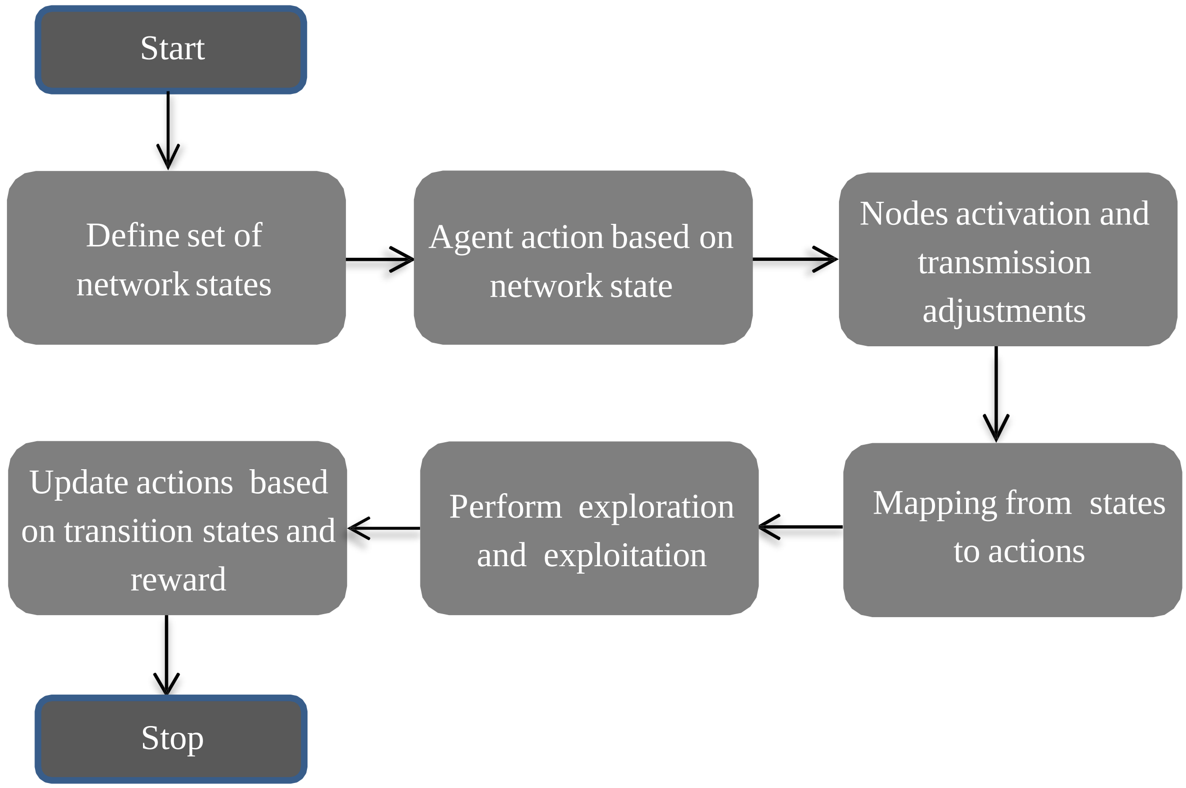

Figure 6: The functional block diagram of Deep RL.

SOM grid represents the optimal path in the sensor network for data transmission from the source to the sink by considering energy efficiency.

The integration of Self-Organizing Maps (SOM) and Optimized Link State Routing (OLSR) involves a two-step process. Firstly, during the offline phase, historical network data is utilized to train the SOM. This training captures the topological structure of the network. Subsequently, during OLSR initialization, the knowledge gained from SOM training is injected into the routing process. OLSR routing tables are optimized using the learned topological map, enabling more informed initial routing decisions. This integration is crucial for enhancing the efficiency and convergence of OLSR, particularly during network initialization and topology changes.

4.2. Routing based on DRL in MWSNs

The DRL-OLSR algorithm utilizes DRL to optimize routing in MWSN. This process is depicted in Figure 6 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, 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 $\mathcal{A}(s_i)$. 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 $P(s_{i+1}|s_i, a_i)$.as depicted in Figure 4. The basic illustration of MDP is depicted in Figure 7. An episode in this process forms a limited sequence of states, actions, and transition functions that return the next state, and rewards in (S, A, δ, R) as given in (10).

$$s_0, a_0, r_1, s_1, a_1, r_2, s_2, ..., s_{n-1}, a_{n-1}, r_n, s_n \tag{10}$$

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 can be represented as,

$$R = r_1 + r_2 + r_3 + ... + r_n \tag{11}$$

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.

$$\pi^* = \arg\max_{\pi(s)} E\ [R_i + \gamma R_{i+1} + \gamma^2 R_{i+2}] \tag{12}$$

where pi^* 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.

$$Q(s_i, a_i) = \mathbb{E}\left[r_i + \gamma \max_{a'} Q(s_{i+1}, a')\right] \tag{13}$$

$Q(s_i, a_i)$ 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 maximise the expected return at each time step in order to maximise overall reward.

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.

$$Q_i^*(s_i, a_i) = (1 - \alpha)Q_{i-1}(s_i, a_i) + \alpha[r_i + \gamma \max_{a_{i+1}} Q_{i-1}(s_{i+1}, a_{i+1})] \tag{14}$$

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

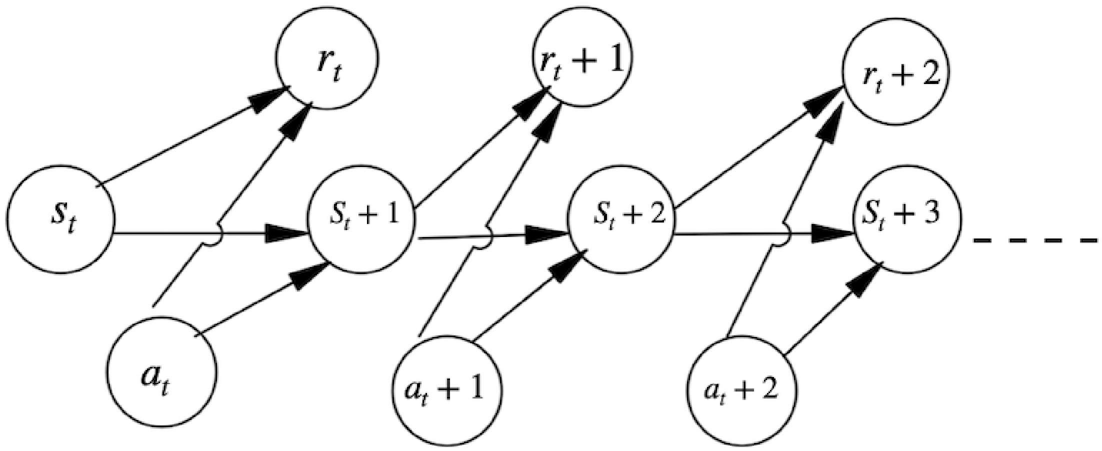

Figure 7: The Markov Decision Processes (MDP).

learn complex relationships between the input and output. The Q-value can be derived as follows:

$$Q_\pi(s_i, a_i) = R(s_i, a_i) + \gamma \sum_{s_{i+1} \in S} P(s_i, a_i, s_{i+1}) Q_i^*(s_{i+1}, a_{i+1}) \tag{15}$$

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}. Then update Q-value by the following formula:

$$\begin{aligned} Q_\pi(s_{i+1}, a_{i+1}) = Q_\pi(s_i, a_i) + \alpha \times \Big[R(s_i, a_i) + \\ \gamma \sum_{s_{i+1} \in S} P(s_i, a_i, s_{i+1}) Q_i^*(s_{i+1}, a_{i+1}) - Q_\pi(s_i, a_i) \Big] \end{aligned} \tag{16}$$

where $\alpha \in [0, 1]$ is the learning rate in(16). The optimal action a_i can be obtained as follows:

$$a_i^* = \arg\max \ Q_\pi(s_i, a_i) \tag{17}$$

Therefore, the optimal policy can be derived from the optimal action as given in (18).

$$L_w = E\left[(Q_\pi(s_i, a_i) - Q_\pi(s_i, a_i^*, w))^2 \right] \tag{18}$$

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.

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 (moving node) 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 behaviour. 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.

DRL-OLSR involves a sophisticated interplay between individual node learning and the collective routing behaviour of the network. Each network node undergoes a Q-learning algorithm modelled as a Markov Decision Process (MDP) within the DRL framework. The learned policies from DRL dynamically influence the parameters and behaviours of the OLSR routing protocol on an individual node basis. Furthermore, there is a continuous feedback loop wherein the real-time dynamics detected by OLSR feed back into the DRL process, ensuring that the collective learning of the network adapts to changes in topology and environmental conditions. This tight integration aims to improve the adaptability, reliability, and efficiency of routing in Mobile Wireless Sensor Network (MWSN) scenarios.

4.3. Training the 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.

Enhancing performance by leveraging function approximation and utilizing samples is crucial for effectively managing large environments in reinforcement learning. These factors play a significant role in the approach's effectiveness. 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(19).

$$Q_i\left(s_i, a_i\right) = (1-\alpha)Q_{i-1}\left(s_i, a_i\right) + \alpha\left[r_i + \gamma \max_{a_{i+1}} Q_{i-1}\left(s_{i+1}, a_{i+1}\right)\right] \tag{19}$$

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

5. Simulation Results

The MWSN network is simulated using the MATLAB environment, with the simulation parameters listed in TABLE 2 and TABLE 3. The network comprises multiple nodes that are randomly distributed as depicted in Figure 8. Using the node-to-node infrastructure, packets are broadcasted to surrounding nodes based on the network's

Algorithm 2 DRL-OLSR 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 Node 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 5A: Generate the simulation of the transmission and integration of the nodes with the neighbour nodes in the network.
Step 5B: Broadcast Link State Packet (LSP) to neighboring nodes LSP=$X_{Loc}, Y_{Loc}, E(x), E_{cons}(x)$
Step 6: Evaluate the node-to-node distance to apply OLSR
$Dist(n(x) : n(y))$
$Dist = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
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

Table 3: 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 s
Maximum Radius	60 s
Coverage Distance	150 m
Total Iterations	50
RNN States	8
Maximum Steps per Episode(RNN)	50
Epsilon	0.9
Epsilon Decay	0.9

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 utilized artificial neural networks and deep learning-based reinforcement learning techniques. The whole proposed methodology used in simulation analysis is illustrated in Figure 9.

The simulation parameters for the mobile wireless sensor network encompass critical factors governing the movement dynamics of sensor nodes. The node speed, initial direction, and pause time establish fundamental characteristics, with a default speed of 1 m/s and random initial directions. The details are given in TABLE 3.

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

5.1. OSLR Performance evaluation without training

Figure 10 to Figure 14 shows the performance analysis when no training is considered. The performance is evaluated in terms of connection probability with different network coverage as shown in Figure 10. The connection probability is less in this case as it can be seen that there is a huge effect of dynamic topology change which needs optimization. As there is less connection probability, the E2E delay also increases as shown in Figure 11 which is not the desired output. The E2E delay must be as low as possible to have a high packet delivery rate. The E2E delay can be minimized using optimization. As shown in Figure 12 routing overhead also increases which should not be the case. Routing overhead signifies that a lot of packets are at the maintenance level which can increase the chances of failures and causes high packet drops. In ad-

Table 4: 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

dition, high packet drops directly affect the throughput of the network. Throughput and energy consumption are shown in Figure 13 and Figure 14, respectively. Both throughput and energy consumption are not appropriate and it needs enhancement and improvement which is done using SOM and DRL implementation.

5.2. Performance evaluation using Self-organizing Map

The performance analysis using the self-organizing maps is shown in Figure 15 to Figure 19. As illustrated the performance is measured in terms of connection probability with various network coverage. In this instance, the connection probability is low as compared to the analysis implemented without optimization which can be seen in Figure 15. Although the probability is low, the E2E delay is improved as shown in Figure 16, which shows that overall packet transmission is taking fewer executions to transmit packets at high data rates.

The E2E delay must be as minimal as possible to have a high packet receiving rate. There is a significant impact on E2E delay when the chance of connection declines. In the case of SOM, it can be observed that the E2E delay is decreasing and can be further

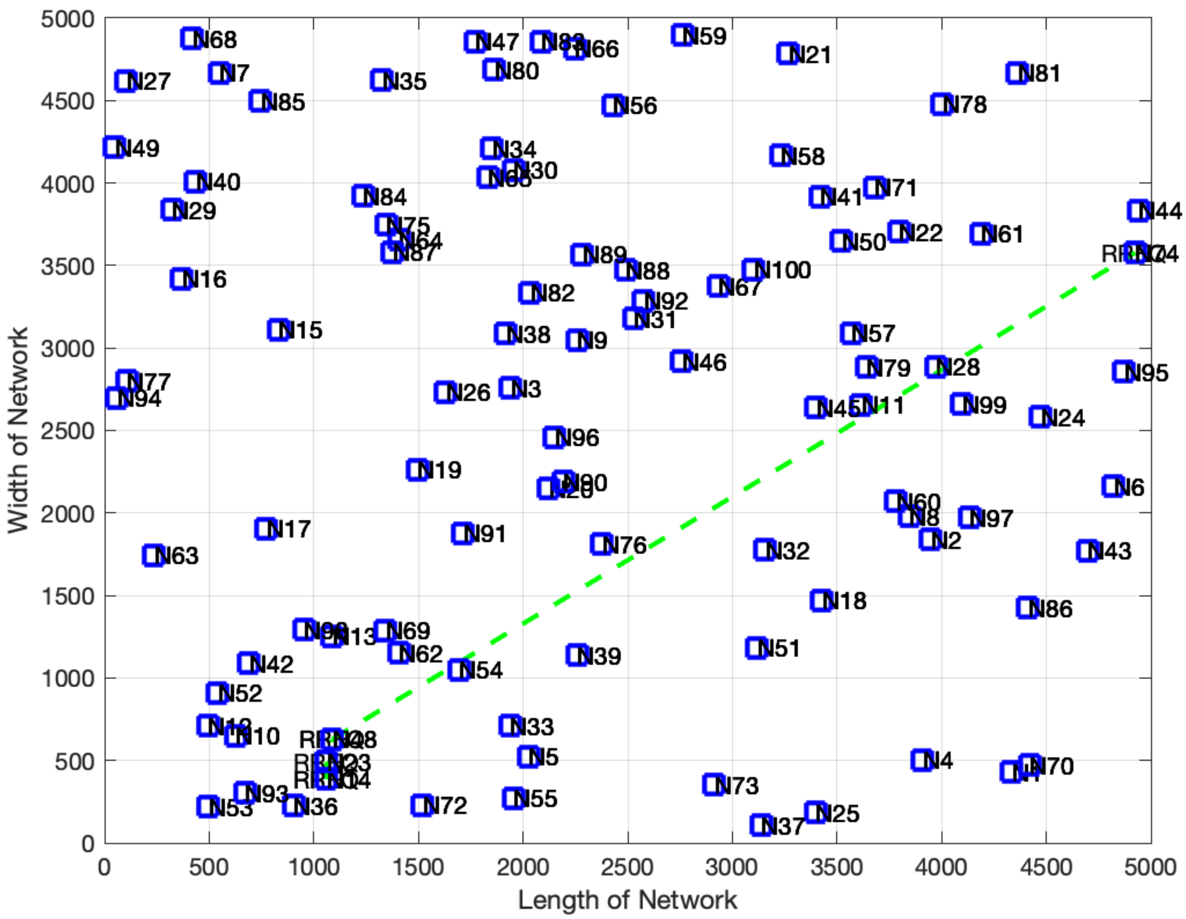

Figure 8: MWSN network simulation setup.

reduced by training the network with more iterations with dense layers. In Figure 17, the routing overhead also improved, which should be controlled to achieve high control of packets with high data rates and high mobility.

Routing overhead shows that many packets are at the maintenance level, which raises the possibility of failures and massive packet dropouts. Additionally, the network throughput also improved in the case of neural networks and indicates high successful packet deliveries at the receiver side as shown in Figure 18. The results also show that the energy consumption is also lower as shown in Figure 19. The higher energy consumption causes more failure of nodes which causes network failure.

5.3. Training using Reinforcement Learning

Figure 20 shows the training process using deep reinforcement learning (DRL). DRL executed for the number of episodes to achieve low losses for evaluation of rewards. It can be seen from Figure 20, the commutative reward is increasing which shows the quality of learning and efficient decision-making process. It enhances the frequency and

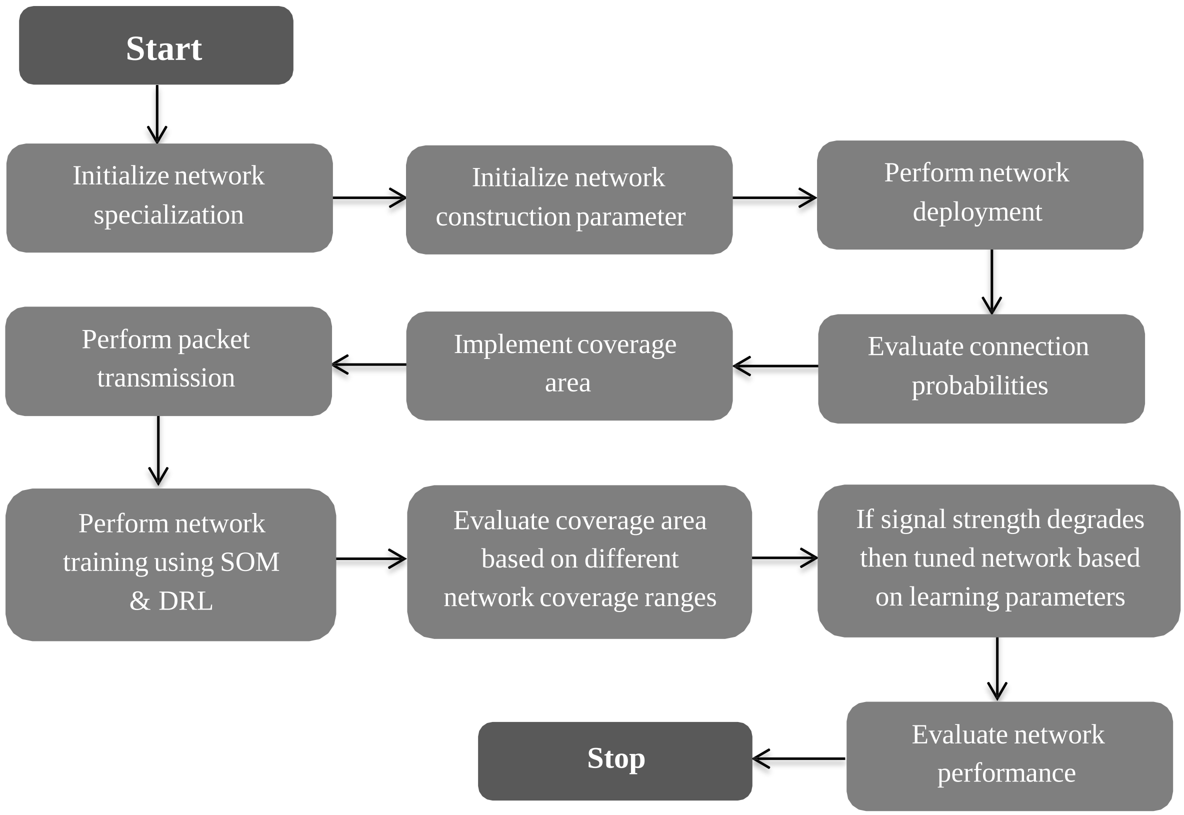

Figure 9: The functional block diagram of the proposed method.

strength of the actions that lead to reduced energy consumption, fewer path losses, and low network error rates.

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

The orange dark line signifies 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.

The light yellow line signifies the quality of the current episode (Episode Q_o) 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.

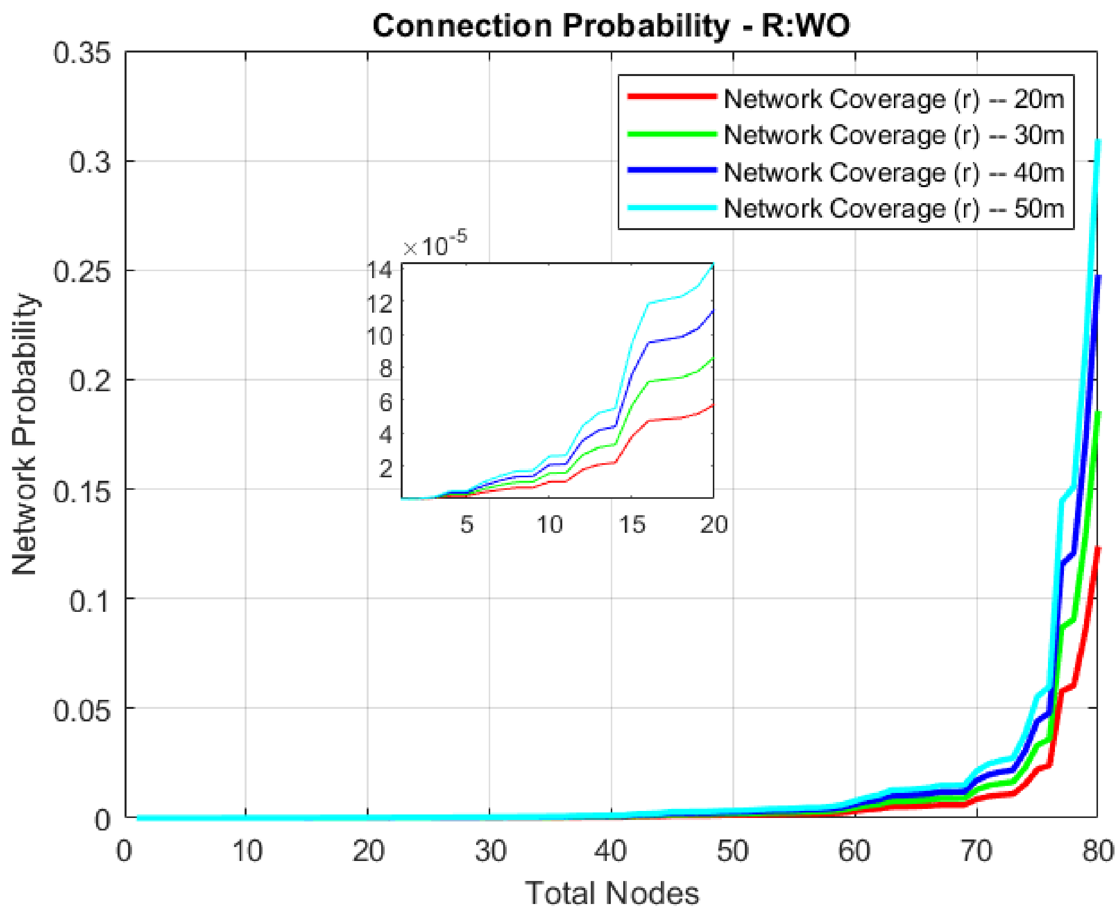

Figure 10: Performance Evaluation without optimization Connection Probability vs. total number of nodes

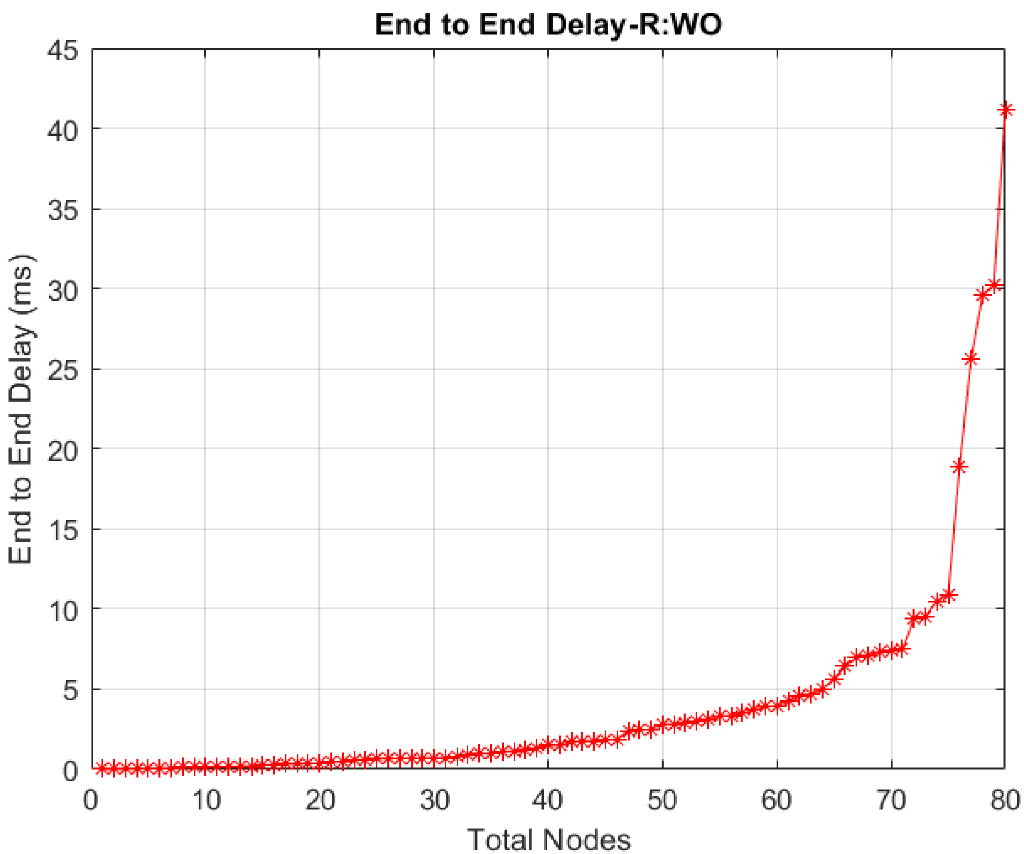

Figure 11: Performance Evaluation without optimization E2E delay vs. total number of nodes

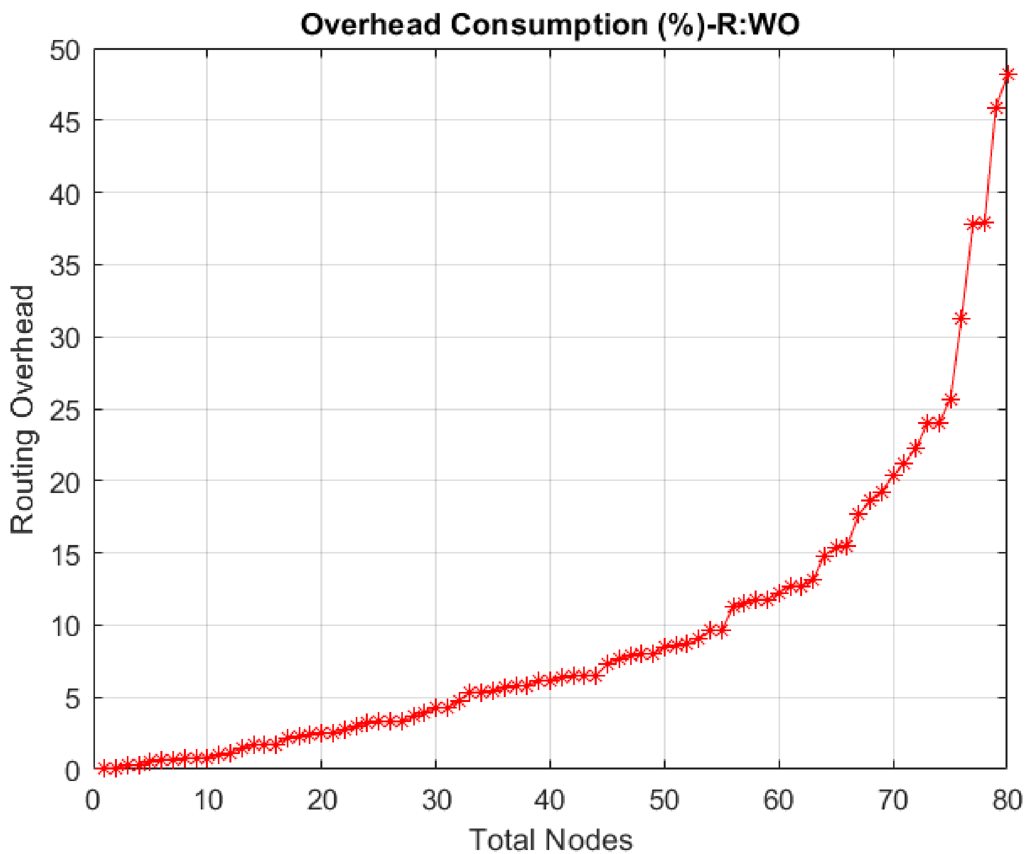

Figure 12: Performance Evaluation without optimization Routing Overhead vs. total number of nodes

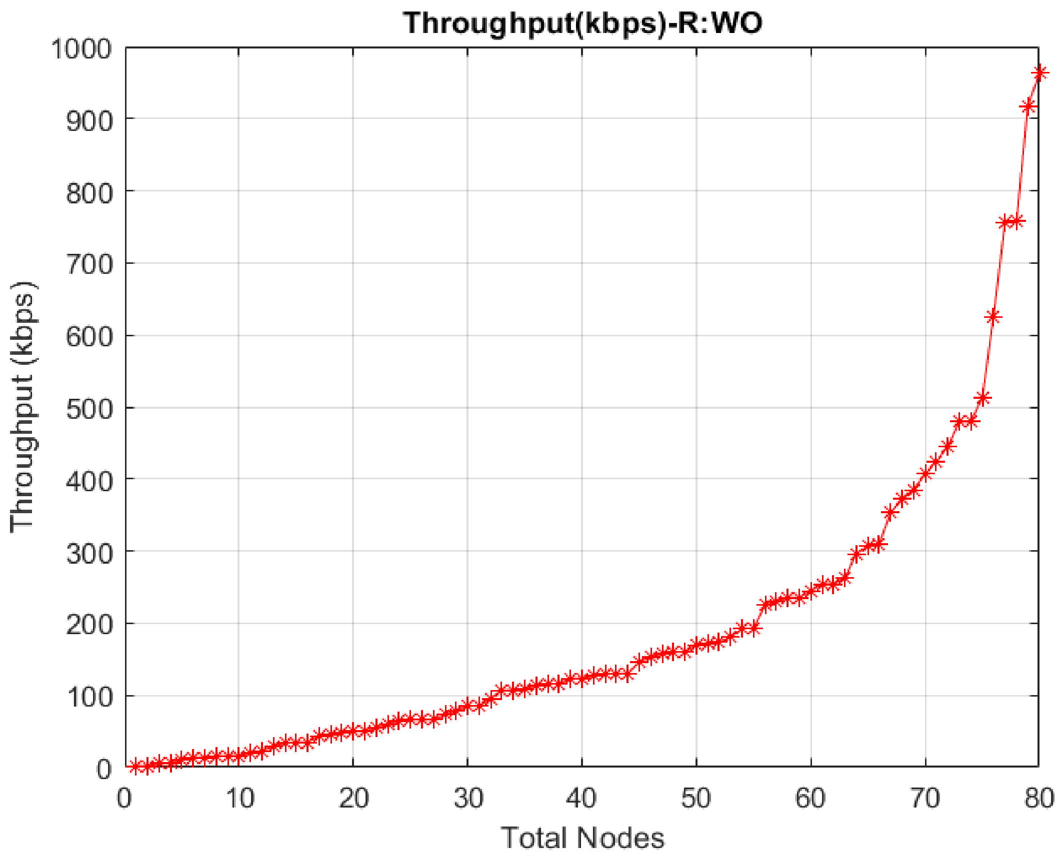

Figure 13: Performance Evaluation without optimization Network Throughput vs. total number of nodes.
.

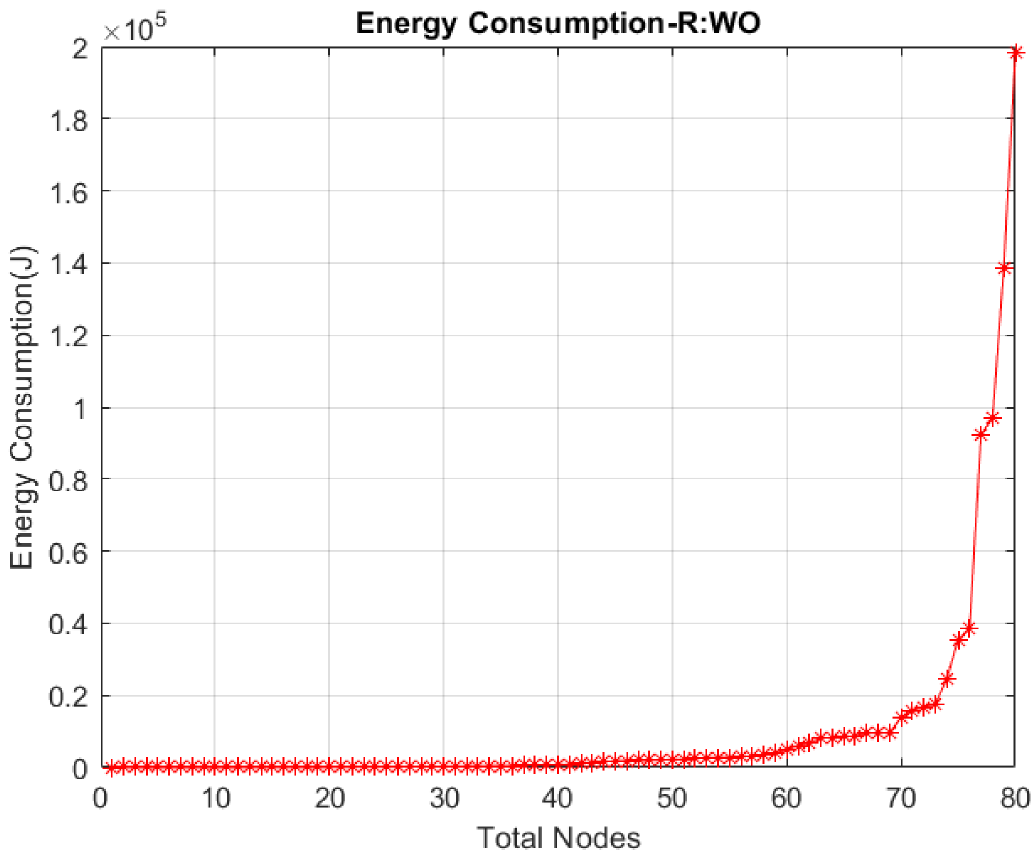

Figure 14: Performance Evaluation without optimization Energy consumption vs. total number of nodes.

.

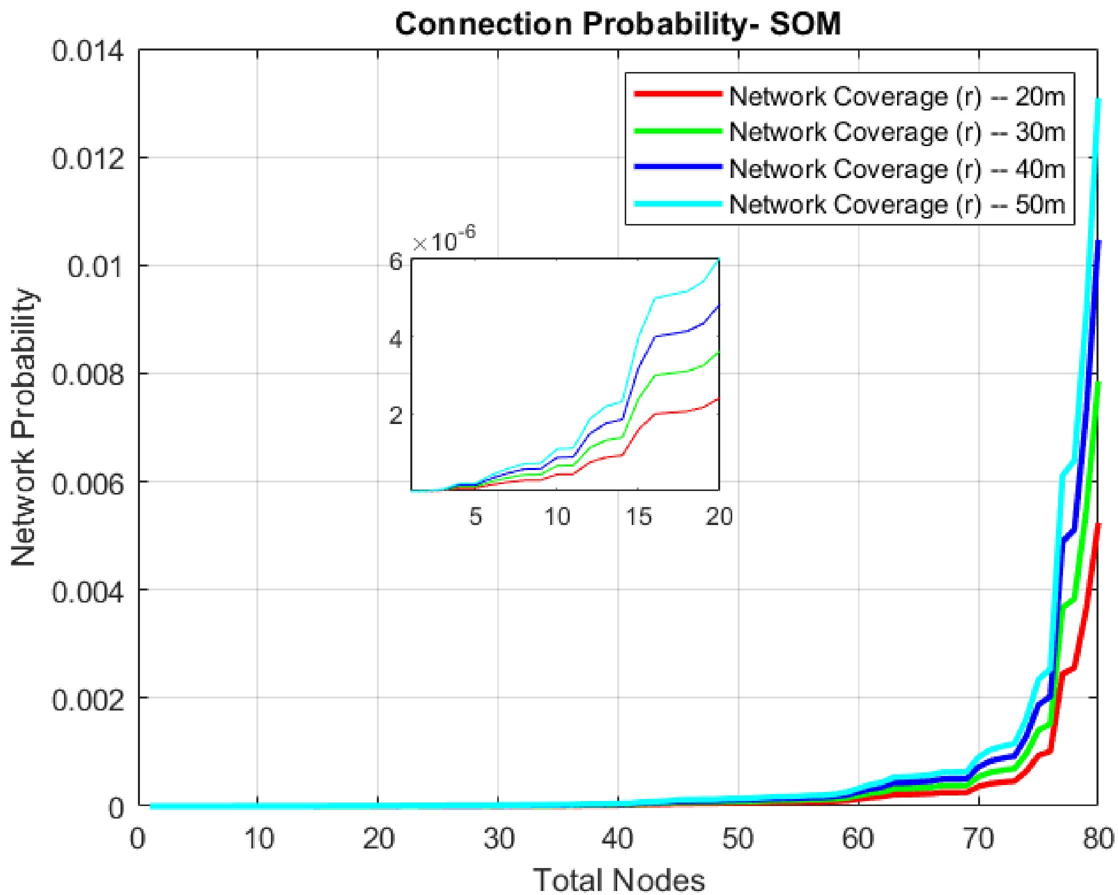

Figure 15: Performance Evaluation using Self-organizing Map Connection Probability vs. total number of nodes.

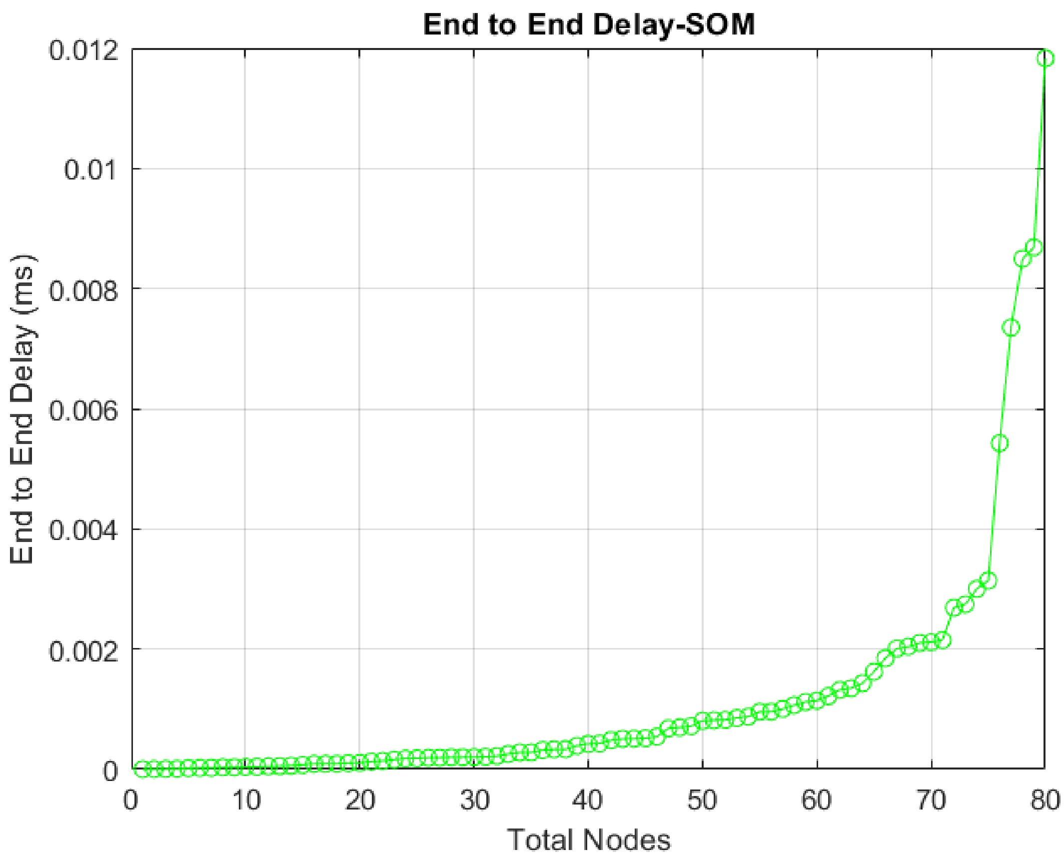

Figure 16: Performance Evaluation using Self-organizing Map E2E delay vs. total number of nodes.

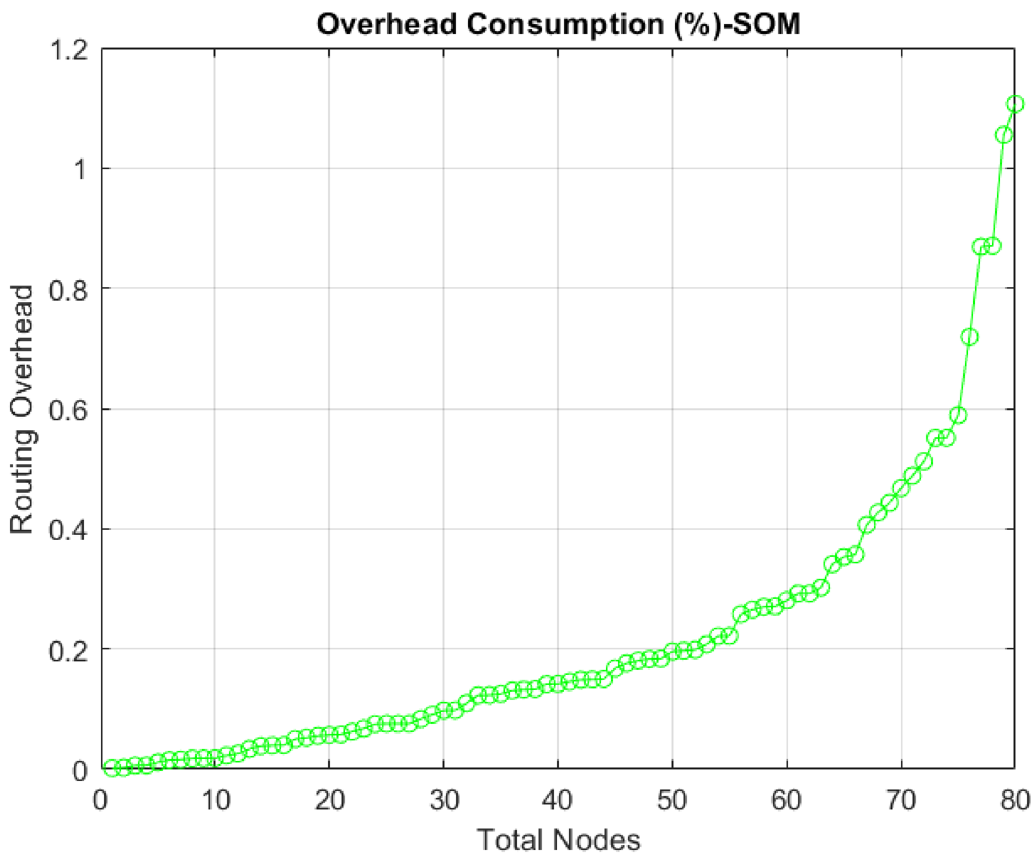

Figure 17: Performance Evaluation using Self-organizing Map Routing overhead vs. total number of nodes.

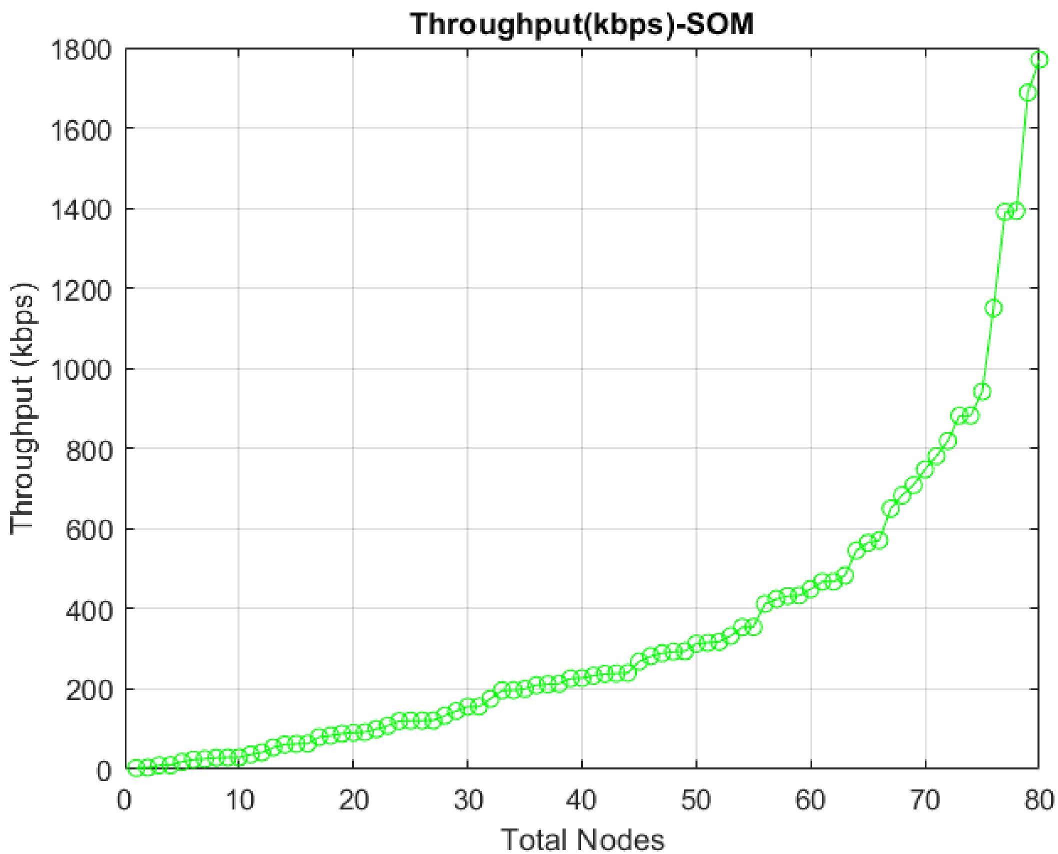

Figure 18: Performance Evaluation using Self-organizing Map Network Throughput vs. total number of nodes.

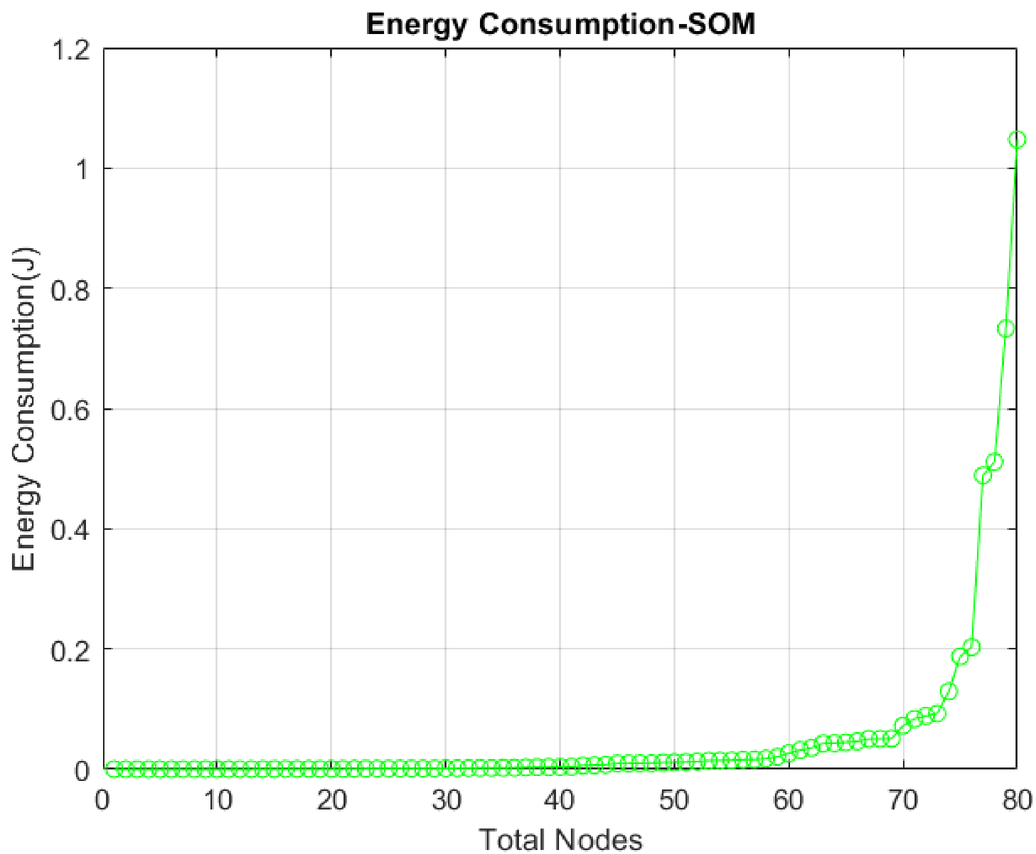

Figure 19: Performance Evaluation using Self-organizing Map Energy consumption vs. total number of nodes.

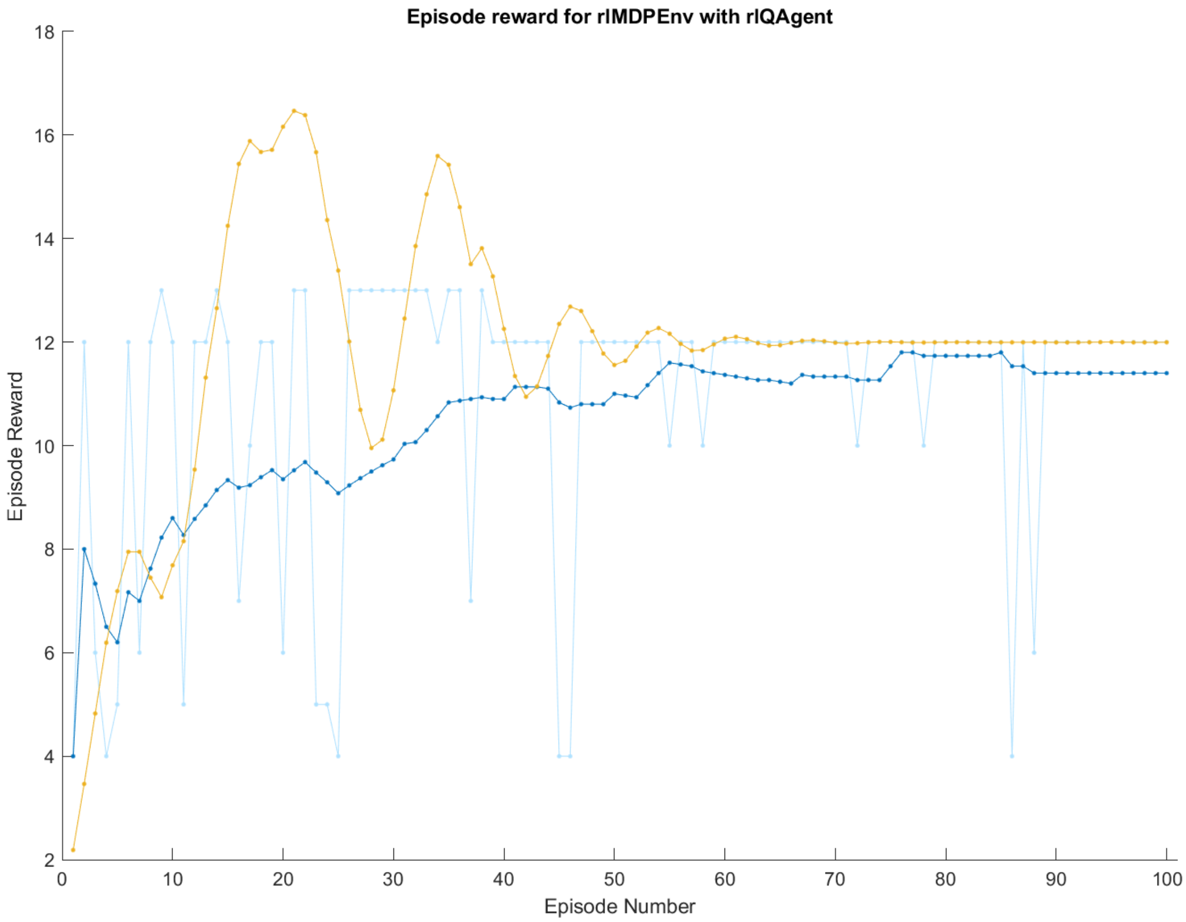

Figure 20: Deep Reinforcement Learning Training Process.

5.4. *Performance evaluation using Deep reinforcement learning*

Figure 21 to Figure 25 shows the performance analysis by considering reinforcement learning. The performance is measured in terms of connection probability with different network coverage, as shown in Figure 21. The connection probability in the case of reinforcement learning is high as compared earlier two cases. The E2E delay is also decreased as shown in Figure 22 which signifies that it requires fewer executions to transmit packets at high data rates. The E2E delay must be as low as possible to have a high packet receiving rate.

By decreasing the likelihood of connection, as it does in the case of reinforcement learning with achieving high rewards for the optimized route, there is a considerable influence on E2E delay. The routing overhead, which should be reduced to reach optimal monitoring of packets with high bandwidth rates and high portability, is also improved and can be seen in Figure 23. There are a lot of packets at the predetermined levels, according to routing overhead, which increases the risk of breakdowns and major packet interruptions which is improved in the proposed work. Moreover, in the case of

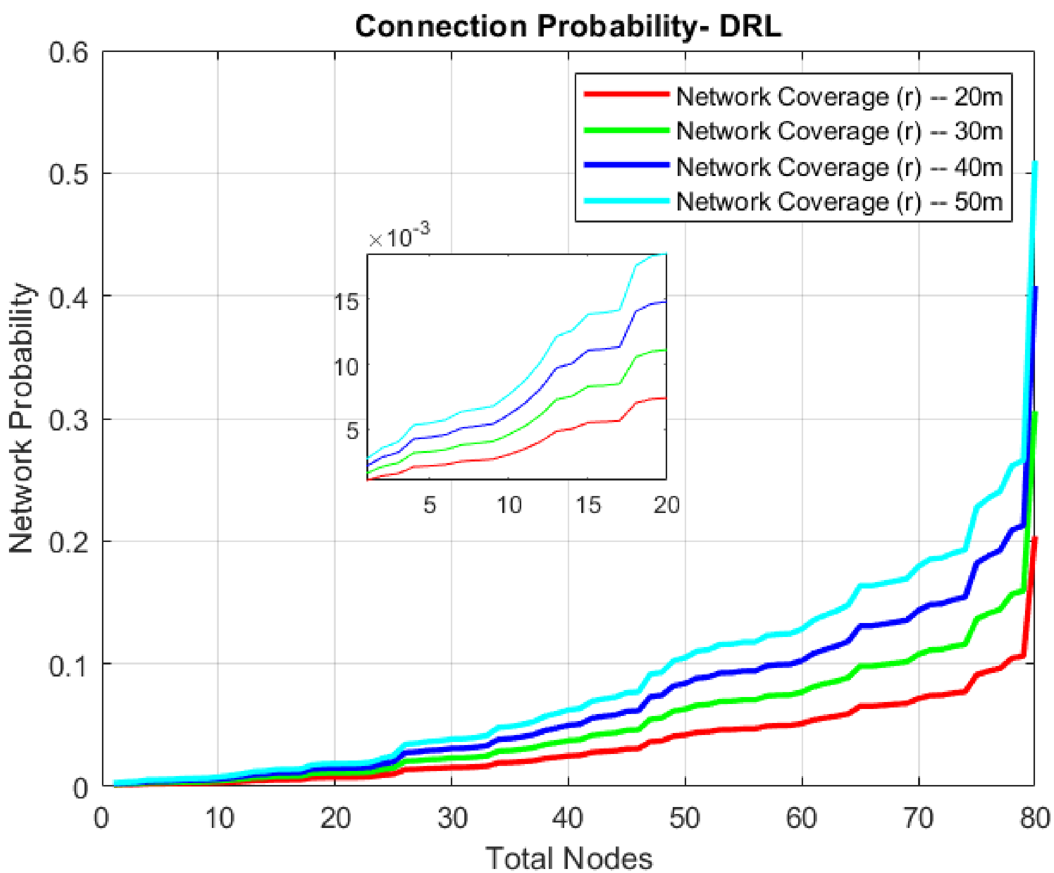

Figure 21: Performance Evaluation using Deep reinforcement learning Connection Probability vs. total number of nodes.

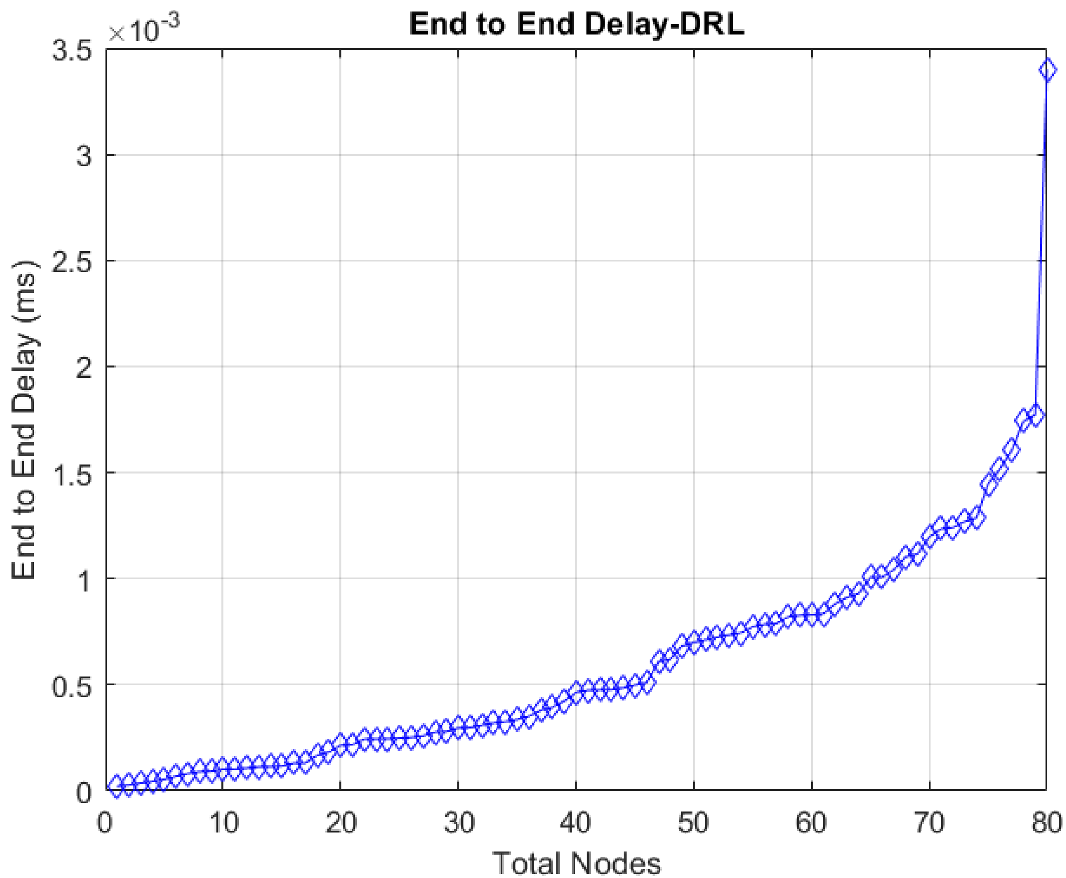

Figure 22: Performance Evaluation using Deep reinforcement learning E2E delay vs. the total number of nodes.

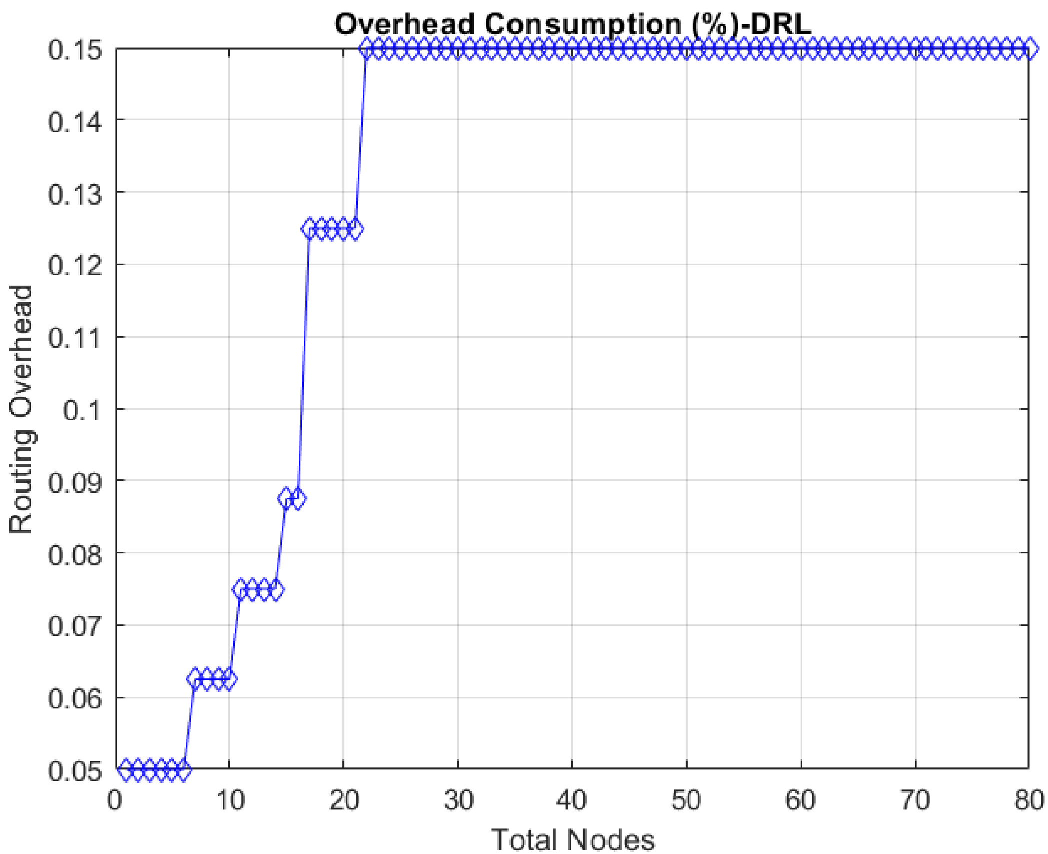

Figure 23: Performance Evaluation using Deep reinforcement learning Routing Overhead vs. total number of nodes.

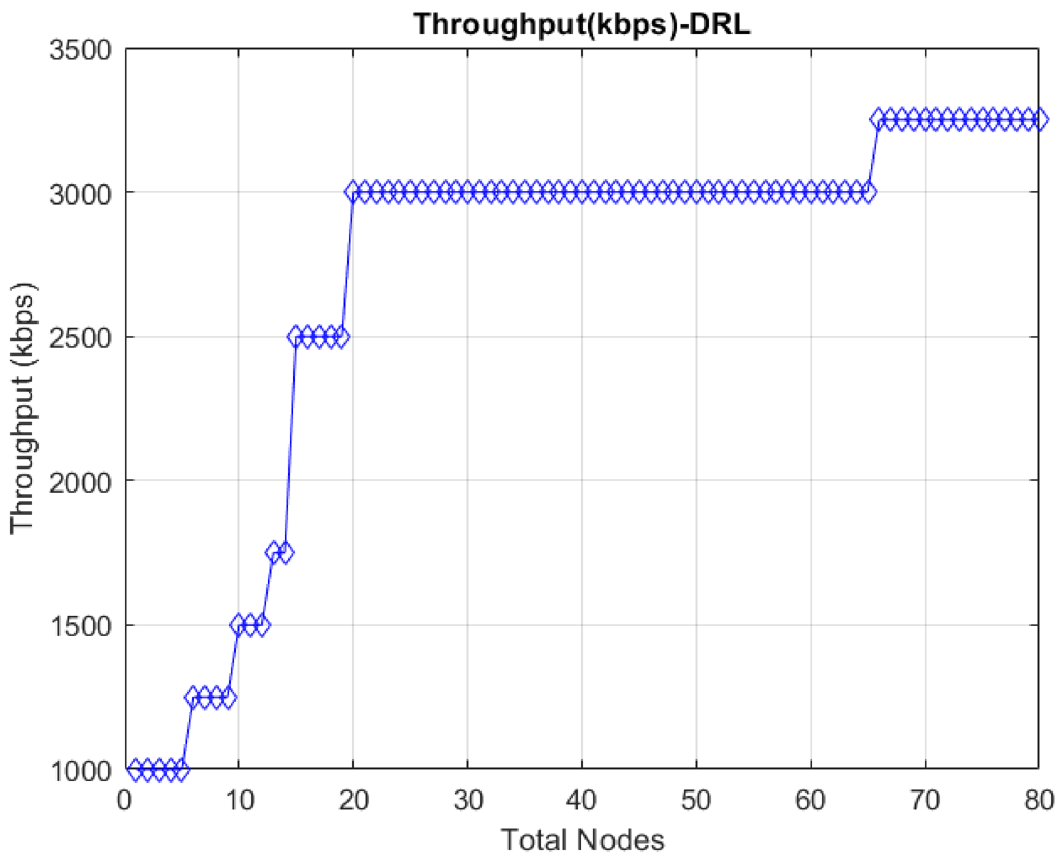

Figure 24: Performance Evaluation using Deep reinforcement learning Network Throughput vs. total number of nodes.

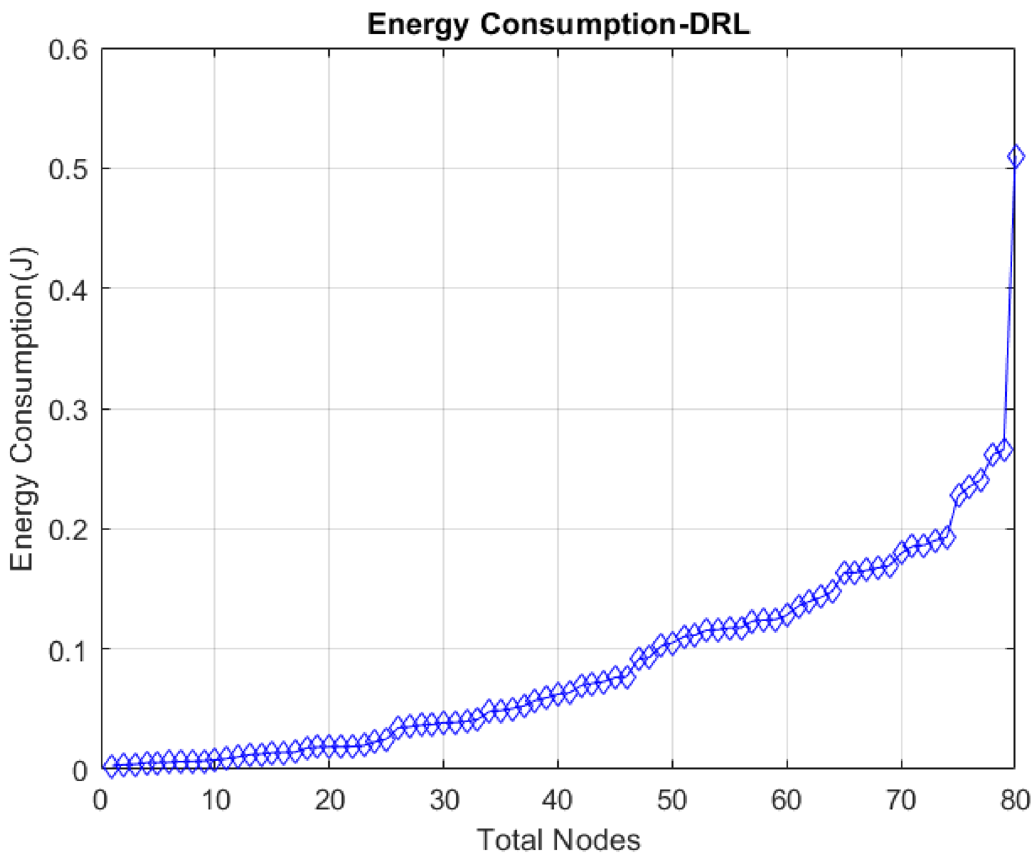

Figure 25: Performance Evaluation using Deep reinforcement learning Energy Consumption vs. Total Number of nodes.

reinforcement learning the network's capacity is high as shown in Figure 24. It indicates a high percentage of successful packet deliveries at the receiver side.

It is also noticed that the energy consumption of the proposed DRL method is also low, as shown in Figure 25. Energy consumption is a very critical part of the sensor networks. The low energy consumption indicates that the resources are efficiently utilized and there is proper load balancing among nodes in the network as a result failures of packets are also reduced.

6. Performance Comparison

Figure 26 to Figure 30 shows the performance comparison of our proposed work in a graphical view. It can be clearly seen that the DRL outperforms with respect to other routing methods and shows high network throughput. In addition, SOM is performing well but slightly less than the DRL as DRL is a deep quality-based learning approach which includes a dense arrangement of the neurons in the network.

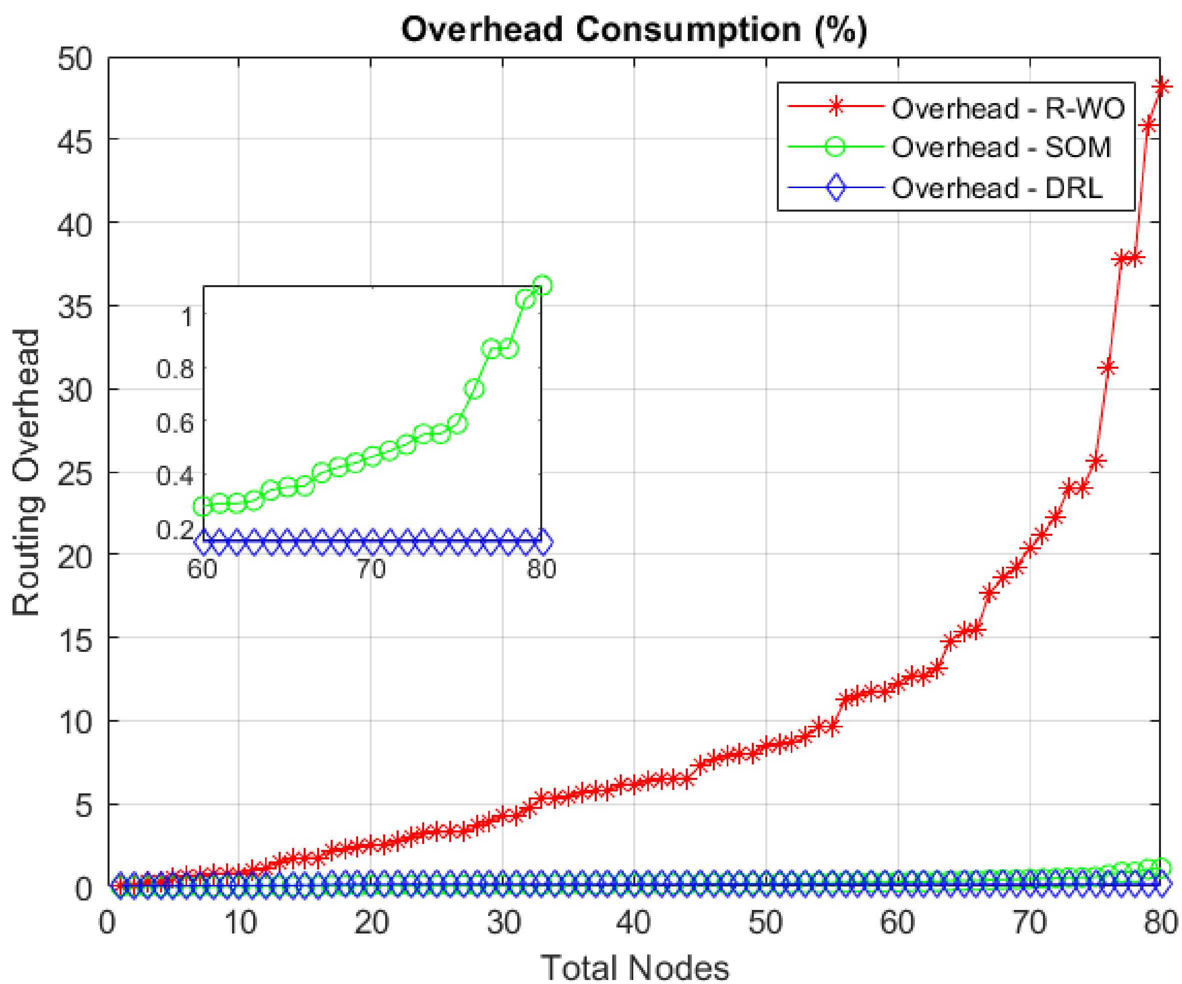

Figure 26: Performance comparison of Routing overhead vs. total number of nodes

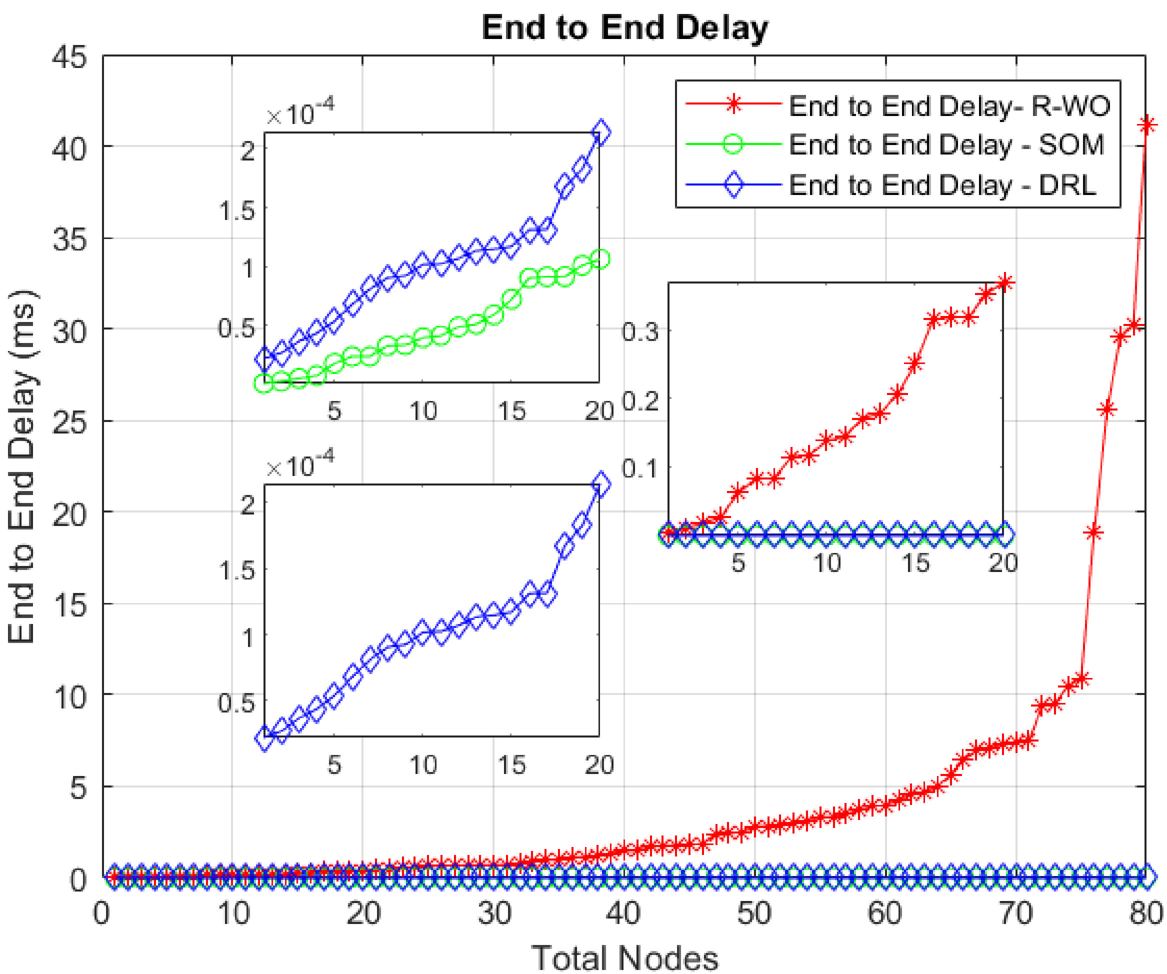

Figure 27: Performance comparison of E2E delay vs. total number of nodes

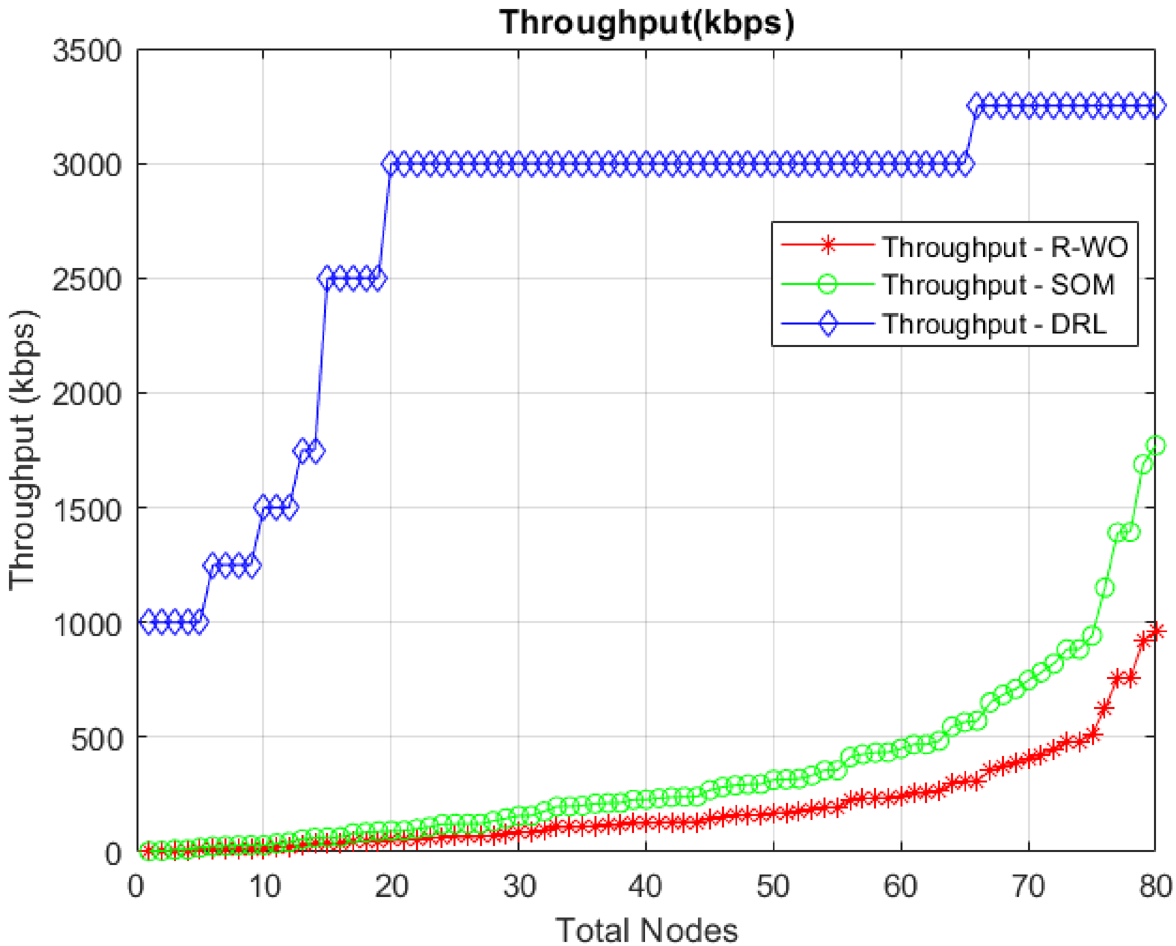

Figure 28: Performance comparison of Network Throughput vs. total number of nodes.

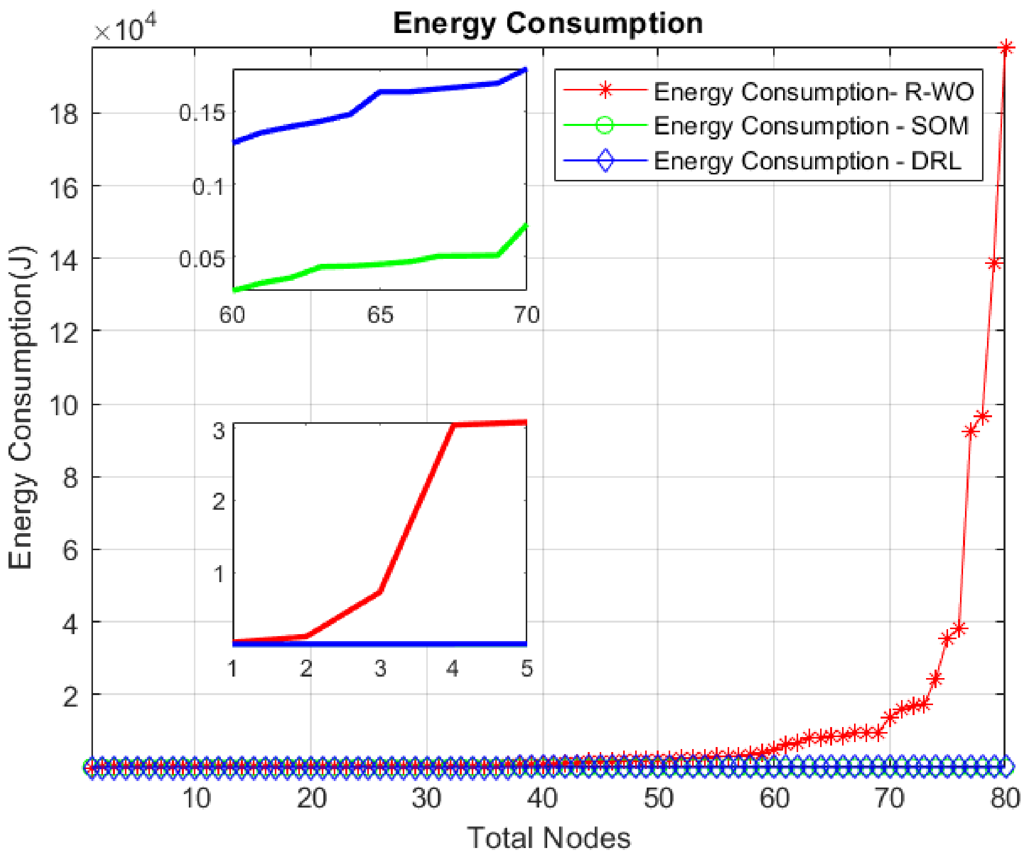

Figure 29: Performance comparison of Energy Consumption vs. total number of nodes.

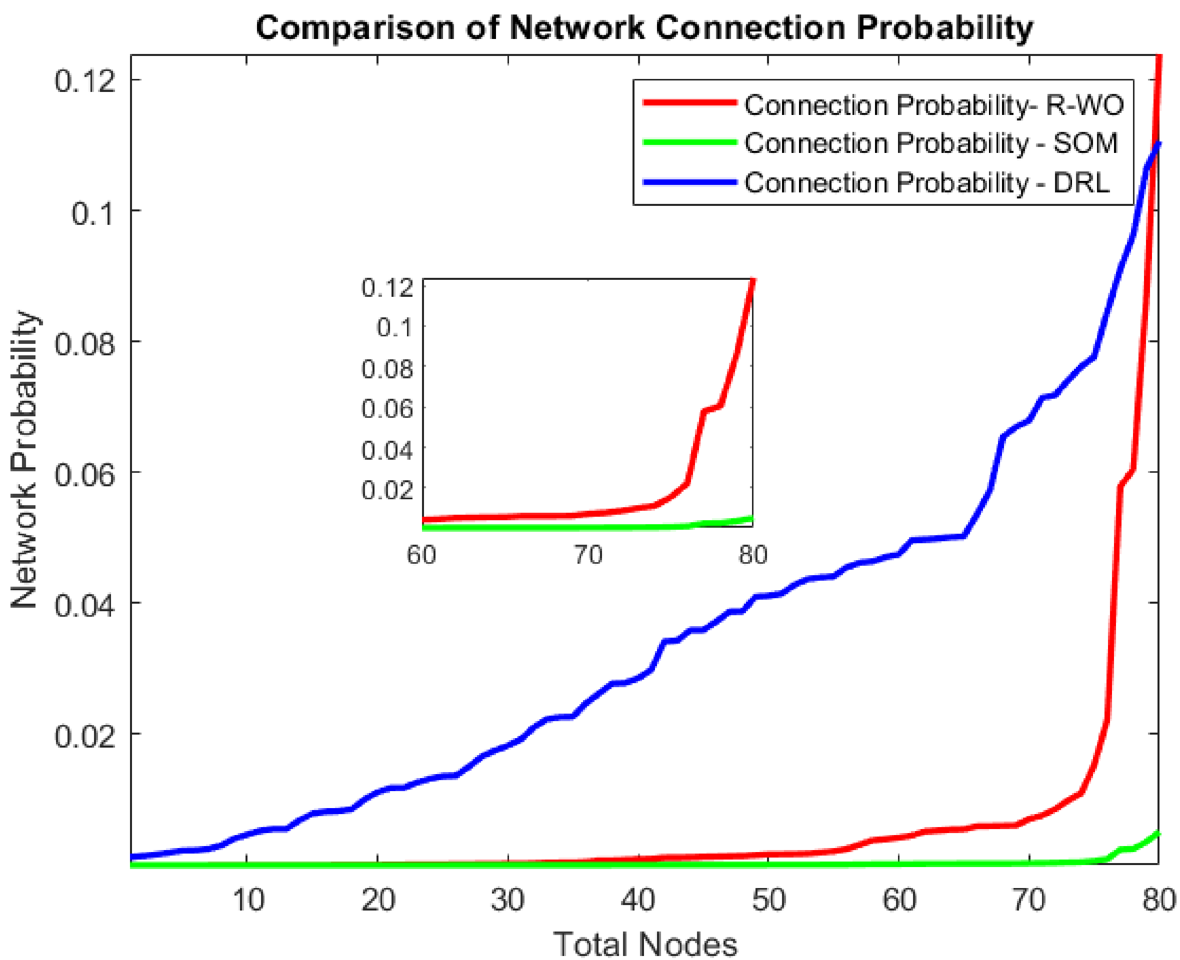

Figure 30: Performance comparison of Connection Probability vs. total number of nodes.

Also the overhead and E2E delay is less in case of DRL which shows that out proposed approach is able to achieve high QoS. Routing overhead which can be seen in Figure 26 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 is not 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 E2E delay as shown in Figure 27, high network throughput as shown in Figure 28, and low energy consumption as shown in Figure 29. Also, the connection probability is also compared for all three conditions as shown in Figure 30. The results confirm that the connection probability of DRL is significantly high as compared to the SOM and 'No optimization' methods. Therefore, it can be concluded that the DRL method outperforms in terms of all QoS parameters.

7. Conclusion

Energy optimization has always been a major challenge in the formation of wireless sensor networks. The presence of mobile nodes leads to irregular changes in nearby nodes' distance and positions, which further complicates 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 results clearly illustrate that the Deep Reinforcement Learning (DRL) method outperforms other routing approaches in terms of network throughput. This proves the superiority of DRL in achieving high-quality service performance. While the Self-Organizing Map (SOM) method performs well it falls slightly behind DRL, mainly due to DRL's dense arrangement of neurons in the network. Besides, the overhead and end-to-end (E2E) delay are lower in the case of DRL, indicating that the proposed approach achieves high Quality of Service (QoS).

In the proposed work, we've focused on free space propagation. Future endeavours could broaden the scope by integrating the impacts of multipath fading and interference among neighbouring nodes. This expansion would enrich the study's comprehensiveness. 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. As we have already concluded, DRL-OLSR performs better than the SOM-OLSR. The only drawback of DRL-OLSR is the high computational cost due to the large dataset used for training. In addition, future energy requirements can also be determined in future work by analyzing mobility profiles with swarm intelligent optimization tools.

References

[1] S. P. Yadav, A. Kumbhare, and R. Parab, "Smart home application using internet of things," *perception*, vol. 6, no. 02, 2019.

[2] T.-h. Kim, C. Ramos, and S. Mohammed, “Smart city and IoT,” pp. 159–162, 2017.

[3] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, “Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare,” *Future Generation Computer Systems*, vol. 78, pp. 659–676, 2018.

[4] A. Rego, A. Canovas, J. M. Jiménez, and J. Lloret, “An intelligent system for video surveillance in IoT environments,” *IEEE Access*, vol. 6, pp. 31 580–31 598, 2018.

[5] Farzinvash, Leili and Najjar-Ghabel, Samad and Javadzadeh, Tahmineh, “A distributed and energy-efficient approach for collecting emergency data in wireless sensor networks with mobile sinks,” *AEU-International Journal of Electronics and Communications*, vol. 108, pp. 79–86, 2019.

[6] “An effective metaheuristic based node localization technique for wireless sensor networks enabled indoor communication,” *Physical Communication*, vol. 48, p. 101411, 2021.

[7] D. K. Sah and T. Amgoth, “Parametric survey on cross-layer designs for wireless sensor networks,” *Computer Science Review*, vol. 27, pp. 112–134, 2018.

[8] “Adaptive cooperative routing transmission for energy heterogeneous wireless sensor networks,” *Physical Communication*, vol. 49, p. 101460, 2021.

[9] D. Praveen Kumar, A. Tarachand, and A. C. S. Rao, “Machine learning algorithms for wireless sensor networks: A survey,” *Information Fusion*, vol. 49, pp. 1–25, 2019.

[10] S. P. R. Banoth, P. K. Donta, and T. Amgoth, “Dynamic mobile charger scheduling with partial charging strategy for wsns using deep-q-networks,” *Neural Computing and Applications*, pp. 1–13, 2021.

[11] Y. Chang, H. Tang, Y. Cheng, Q. Zhao, and B. Yuan, “Dynamic hierarchical energy-efficient method based on combinatorial optimization for wireless sensor networks,” *Sensors (Basel)*, vol. 17, no. 7, p. 1665, Jul. 2017.

[12] S. P. R. Banoth, P. K. Donta, and T. Amgoth, “Target-aware distributed coverage and connectivity algorithm for wireless sensor networks,” *Wirel. Netw.*, vol. 29, no. 4, pp. 1815–1830, May 2023.

[13] L. Cao, Y. Cai, and Y. Yue, “Swarm intelligence-based performance optimization for mobile wireless sensor networks: survey, challenges, and future directions,” *IEEE Access*, vol. 7, pp. 161 524–161 553, 2019.

[14] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, "A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 1, pp. 550–586, 2017.

[15] P. K. Donta, T. Amgoth, and C. S. R. Annavarapu, "Delay-aware data fusion in duty-cycled wireless sensor networks: A q-learning approach," *Sustain. Comput. Inform. Syst.*, vol. 33, no. 100642, p. 100642, Jan. 2022.

[16] Y. Chang, W. Chen, J. Li, J. Liu, H. Wei, Z. Wang, and N. Al-Dhahir, "Collaborative multi-bs power management for dense radio access network using deep reinforcement learning," 2023.

[17] Y. Chang, X. Yuan, B. Li, D. Niyato, and N. Al-Dhahir, "A joint unsupervised learning and genetic algorithm approach for topology control in energy-efficient ultra-dense wireless sensor networks," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2370–2373, Nov. 2018.

[18] M. Masikos, K. Demestichas, E. Adamopoulou, and M. Theologou, "Machine-learning methodology for energy efficient routing," *IET Intelligent Transport Systems*, vol. 8, no. 3, pp. 255–265, 2014.

[19] G. S. Kori and M. S. Kakkasageri, "Resource allocation scheme for wireless sensor networks: Machine learning approach," in *2022 International Conference on Signal and Information Processing (IConSIP)*. IEEE, 2022, pp. 1–5.

[20] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, H. Xiong, and L. Akoglu, "A comprehensive survey on graph anomaly detection with deep learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[21] M. Paolanti, L. Romeo, A. Felicetti, A. Mancini, E. Frontoni, and J. Loncarski, "Machine learning approach for predictive maintenance in industry 4.0," in *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. IEEE, 2018, pp. 1–6.

[22] S. Sachan, R. Sharma, and A. Sehgal, "Energy efficient scheme for better connectivity in sustainable mobile wireless sensor networks," *Sustainable Computing: Informatics and Systems*, vol. 30, p. 100504, 2021.

[23] G. S. Sara and D. Sridharan, "Routing in mobile wireless sensor network: A survey," *Telecommunication Systems*, vol. 57, pp. 51–79, 2014.

[24] S. Banik, M. M. Mowla, and I. Ahmad, "A strategic routing analysis for agro sensor communications in mobile ad hoc networks," in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, 2019, pp. 1–6.

[25] M. Hammoudeh and R. Newman, "Adaptive routing in wireless sensor networks: QoS optimisation for enhanced application performance," *Information Fusion*, vol. 22, pp. 3–15, 2015.

[26] X. Liu, "Routing protocols based on Ant Colony Optimization in wireless sensor networks: A survey," *IEEE Access*, vol. 5, pp. 26 303–26 317, 2017.

[27] M. Asif, S. Khan, R. Ahmad, M. Sohail, and D. Singh, "Quality of service of routing protocols in wireless sensor networks: A review," *IEEE Access*, vol. 5, pp. 1846–1871, 2017.

[28] K.-L. A. Yau, H. G. Goh, D. Chieng, and K. H. Kwong, "Application of reinforcement learning to wireless sensor networks: models and algorithms," *Computing*, vol. 97, no. 11, pp. 1045–1075, 2015.

[29] L. Liu, Y. Cheng, L. Cai, S. Zhou, and Z. Niu, "Deep learning based optimization in wireless network," in *2017 IEEE international conference on communications (ICC)*. IEEE, 2017, pp. 1–6.

[30] Y. He, Z. Zhang, F. R. Yu, N. Zhao, H. Yin, V. C. Leung, and Y. Zhang, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 433–10 445, 2017.

[31] Y. He, C. Liang, F. R. Yu, N. Zhao, and H. Yin, "Optimization of cache-enabled opportunistic interference alignment wireless networks: A big data deep reinforcement learning approach," *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2017.

[32] L. Chen, J. Lingys, K. Chen, and F. Liu, "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*, 2018, pp. 191–205.

[33] B. Mao, Z. M. Fadlullah, F. Tang, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "Routing or computing? the paradigm shift towards intelligent computer network packet transmission based on deep learning," *IEEE Transactions on Computers*, vol. 66, no. 11, pp. 1946–1960, 2017.

[34] Y. Lee, "Classification of node degree based on deep learning and routing method applied for virtual route assignment," *Ad Hoc Networks*, vol. 58, pp. 70–85, 2017.

[35] B. Mao, Z. M. Fadlullah, F. Tang, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "A tensor based deep learning technique for intelligent packet routing," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[36] B. Zhao and X. Zhao, "Deep reinforcement learning resource allocation in wireless sensor networks with energy harvesting and relay," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 2330–2345, 2022.

[37] F. Kazemeyni, O. Owe, E. B. Johnsen, and I. Balasingham, "Formal modeling and analysis of learning-based routing in mobile wireless sensor networks," in *Integration of Reusable Systems*. Springer, 2014, pp. 127–150.

[38] M. E. E. D. A. El, A. A. Youssif, A. Z. Ghalwash *et al.*, "Energy aware and adaptive cross-layer scheme for video transmission over wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 21, pp. 7792–7802, 2016.

[39] A. Ray and D. De, "Energy efficient clustering protocol based on k-means (EECPK-means)-midpoint algorithm for enhanced network lifetime in wireless sensor network," *IET Wireless Sensor Systems*, vol. 6, no. 6, pp. 181–191, 2016.

[40] Shashikala and C. Kavitha, *Secure cluster-based data integrity routing for wireless sensor networks*. Singapore: Springer Singapore, 2018.

[41] J.-S. Lee and T.-Y. Kao, "An improved three-layer low-energy adaptive clustering hierarchy for wireless sensor networks," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 951–958, 2016.

[42] H. Farman, H. Javed, J. Ahmad, B. Jan, and M. Zeeshan, "Grid-based hybrid network deployment approach for energy efficient wireless sensor networks," *Journal of Sensors*, vol. 2016, p. 2326917, Oct 2016.

[43] H. J. N. A. J. I. M. A. S. A. Haleem Farman, Bilal Jan, "Multi-criteria based zone head selection in internet of things based wireless sensor networks," *Future Generation Computer Systems*, vol. 87, pp. 364–371, Oct 2018.

[44] Y. Chang, X. Yuan, B. Li, D. Niyato, and N. Al-Dhahir, "Machine-learning-based parallel genetic algorithms for multi-objective optimization in ultra-reliable low-latency WSNs," *IEEE Access*, vol. 7, pp. 4913–4926, 2019.

[45] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[46] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[47] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[48] M. E. Al-Sadoon, A. Jedidi, and H. Al-Raweshidy, "Dual-tier cluster-based routing in mobile wireless sensor network for iot application," *IEEE Access*, vol. 11, pp. 4079–4094, 2023.

[49] B. P. Kumar, J. Umamageswaran, A. Kalpana, and R. Dhanalakshmi, "Mobility and behavior based trustable routing in mobile wireless sensor network," in *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, 2023, pp. 1004–1008.

[50] I. Chakeres and E. Belding-Royer, "Aodv routing protocol implementation design," in *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.*, 2004, pp. 698–703.

[51] A. R. Zarzoor, "Enhancing dynamic source routing (dsr) protocol performance based on link quality metrics," in *2021 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2021, pp. 17–21.

[52] S. Nithya, G. A. Kumar, and P. Adhavan, "Destination-sequenced distance vector routing (dsdv) using clustering approach in mobile adhoc network," in *2012 International Conference on Radar, Communication and Computing (ICRCC)*, 2012, pp. 319–323.

[53] K. Guleria and A. K. Verma, "Comprehensive review for energy efficient hierarchical routing protocols on wireless sensor networks," *Wireless Networks*, vol. 25, pp. 1159–1183, 2019.

[54] P. K. Donta, S. N. Srirama, T. Amgoth, and C. S. R. Annavarapu, "iCoCoA: intelligent congestion control algorithm for CoAP using deep reinforcement learning," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 3, pp. 2951–2966, Mar. 2023.

[55] D. K. Sah, K. Cengiz, P. K. Donta, V. N. Inukollu, and T. Amgoth, "EDGF: Empirical dataset generation framework for wireless sensor networks," *Comput. Commun.*, vol. 180, pp. 48–56, Dec. 2021.

[56] S. Gadekar and S. Kadam, "Secure optimized link state routing (olsr) protocol against node isolation attack," in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, 2017, pp. 684–687.

[57] Z. Fuhao and L. Jiping, "An algorithm of shortest path based on dijkstra for huge data," in *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 4, 2009, pp. 244–247.