

## Research Article

# Enhanced Route Discovery Mechanism Using Improved CH Selection Using Q-Learning to Minimize Delay

Navpreet Kaur <sup>1</sup>, Inderdeep Kaur Aulakh <sup>1</sup>, Sumegh Tharewal <sup>2</sup>, Ismail Keshta <sup>3</sup>,  
Abdul Wahab Rahmani <sup>4</sup> and Tri Duc Ta <sup>5</sup>

<sup>1</sup>IT Department, UIET, Chandigarh, Punjab, India

<sup>2</sup>School of Computer Science, Dr. Vishwanath Karad MIT World Peace University, S. No. 124, Paud Road, Kothrud, Pune 411038, Maharashtra, India

<sup>3</sup>Computer Science and Information Systems Department, College of Applied Sciences, AlMaarefa University, Riyadh, Saudi Arabia

<sup>4</sup>Isteqlal Institute of Higher Education, Kabul, Afghanistan

<sup>5</sup>Faculty of Electronics and Telecommunications, University of Science (HCMUS), Vietnam National University, Ho Chi Minh City 700000, Vietnam

Correspondence should be addressed to Abdul Wahab Rahmani; [ab.wahab.professor@isteqlal.edu.af](mailto:ab.wahab.professor@isteqlal.edu.af)

Received 16 July 2022; Revised 6 August 2022; Accepted 12 August 2022; Published 20 September 2022

Academic Editor: Punit Gupta

Copyright © 2022 Navpreet Kaur et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the technological advancements, practical challenges of establishing long-distance communication should be addressed using hop-oriented routing networks. However, long-distance data transmissions usually deteriorate the quality of service (QoS) especially in terms of considerable communication delay. Therefore, in the presented work, a reward-based routing mechanism is proposed that aims at minimizing the overall delay which is evaluated under various scenarios. The routing process involved a refined CH selection mechanism based on a mathematical model until a threshold simulation is not attained. The illustrations for the coverage calculations of CH in the route discovery are also provided for possible routes between the source and the destination to deliver quality service. Based on this information, the data gathered from the past simulations is passed to the learning mechanism using the Q-learning model. The work is evaluated in terms of throughput, PDR, and first dead node in order to achieve minimal transmission delay. Furthermore, area variation is also involved to investigate the effect of an increase in the deployment area and number of nodes on a Q-learning-based mechanism aimed to minimize the delay. The comparative analysis against four existing studies justifies the success of the proposed mechanism in terms of throughput, first dead node, and delay analysis.

## 1. Introduction

Wireless sensor networks (WSN) are made of tiny sensor nodes that are powered by small batteries that provide power to the sensor node. The sensor nodes are designed to transfer data from one end to another end, or in other words, the sensor nodes transfer the data from the source node to the terminal node. The transfer of the data takes place in collaboration with other nodes in the network and a multihop path is used as a route for the data transfer. Every data transfer follows a process of broadcasting the signals

(R-REQ) in order to get the responses against the transferred signals and are termed as route replies (R-REP). This process is categorized under hierarchical routing [1]. When the sensor node is overoccupied with data to be transferred or if the sensor node is not able to transfer the data due to any physical or software-oriented barriers, a time-lapse is generated, and it is termed as delay [2]. One of the major reasons for the delay is node overhead, and in order to reduce the overhead of the nodes, the concept of clustering was introduced in the early stage of the development of the WSN [3]. When a sensor node is unable to transfer or receive the

data, it is indicated that the node has run out of stored energy which is also called residual energy (RE). A node with 0 residual energy is termed as a dead node. The researchers have focused on maximizing the time interval when the first node is dead in order to increase the lifetime and reduce the overall delay of the network. More number of dead nodes in the network will produce packet overhead and delay. The CH selection method plays a crucial role in order to reduce the overall computation complexity and delay in the network. Since WSN is scalable, it can always make room for more nodes or gadgets. Since it is adaptable, physical divisions are possible. The centralized monitoring system provides access to all WSN nodes. It does not need cords or wires because it is wireless. WSNs may be used extensively and across a wide range of industries, including mining, healthcare, surveillance, and agriculture. It employs various security algorithms in accordance with the underlying wireless technology and thereby offers customers or users a dependable network.

The use of procedures or technologies on a network to manage traffic and assure the functioning of key applications that require limited network capacity is known as quality of service (QoS). It enables enterprises to prioritize certain high-performance apps while adjusting total network traffic. QoS is commonly used in networks that transport traffic for resource-intensive systems. It is commonly required for services such as Internet protocol television (IPTV), online gaming, streaming media, videoconferencing, video on demand (VOD), and voice over IP (VoIP). Organizations may use QoS in networking to improve the performance of various applications running on their network and obtain insight into its bit rate, latency, jitter, and packet rate. By doing so, companies may control network traffic and alter how packets are sent to the Internet or other networks to prevent transmission delays. In addition, this makes sure that the company provides apps with the desired service quality and user experiences. According to the definition of QoS, the main objective is to give networks and organizations the ability to prioritize traffic. This is accomplished by providing dedicated bandwidth, managed jitter, and decreased latency. The technologies that are employed to make this possible are crucial for improving the functionality of corporate applications and wide-area networks.

There are certain methods that may be applied to raise the service quality. Scheduling, traffic shaping, admission control, and resource reservations are the four most often used techniques. A switch or router receives packets from various flows for processing. A good scheduling method addresses the various flows equally and correctly. Several scheduling strategies are intended to raise the level of service. Here, we will talk about three of them: weighted fair queuing, priority queuing, and FIFO queuing. A method of regulating the volume and speed of traffic transmitted to the network is called traffic shaping. A token bucket and a leaky bucket are two methods that can influence traffic. Resources such as buffer, bandwidth, CPU time, and others are required for a data flow. Admission control is the technique through which a router or switch accepts or rejects a flow based on established parameters known as flow

requirements. Before accepting a flow for processing, a router examines the flow requirements to see whether its capacity (in terms of bandwidth, buffer size, CPU speed, and so on) and past commitments to other flows can handle the incoming flow.

Reward-based routing encourages adopting the pathways that result in the least amount of interaction between intermediate nodes while simultaneously taking into account the fact that each freshly forwarded packet results in some more exposure. The reward (metric) connected with each routing choice is carried out through incentive-based packets, which enables routers to assess their rules in light of past usage. In order to avoid the most vulnerable locations, reward-based routing is intended as a technology-neutral routing technique that develops a link between route cost and population exposure. Reward-based routing uses virtual currency (reward) to forward packets; therefore, the more exposure a node causes, the more reward it needs to do so. Reward-based routing, which is a legitimate stand-in for the aforementioned EI, takes into consideration exposure in terms of the number of packets. There are two underlying ideas in reward-based routing. The first is to adjust the hop cost in a way that considers exposure, and the second is to create and keep the least-exposed routes in place. The reward-based routing behavior may have the fascinating side effect of reducing the frequency of retransmissions, which would assist in further limiting the exposure. Reward-based routing attempts to avoid routes with greater loads, which results in fewer crashes and retransmission occurrences. A longer (average) battery lifespan, which encourages energy-aware behaviors, is another benefit of load balancing promoted by the RBR strategy.

Wireless sensor networks are gaining popularity in contemporary IoT-enabled industrial and home applications that use either heterogeneous or homogeneous sensors to gather intent data. Because their application is geographically essential, WSNs are intended to function utilizing self-powered sensor nodes. Such nodes must enable energy efficiency to ensure network lifespan. Cluster head selection is an important element in a WSN design that focuses on minimizing network energy usage. It organizes sensor nodes in such a manner that a complex network cluster is produced, resulting in an increased lifetime and reduced power consumption. The stability of the cluster head (CH) has a significant impact on the network's robustness and scalability. The overhead is decreased since the stable CH assures little intracluster and intercluster communication. This article proposes a novel CH selection method based on a reward mechanism generated by the application of the proposed Q-learning method to enhance the network performance and to reduce the overall delay. Routing decisions are made using the shortest path algorithms in typical routing algorithms. As a result, nodes typically selected along the shortest pathways deplete faster than other nodes, resulting in a shorter network lifespan. Numerous routing techniques have been developed to increase network longevity. The energy-efficient algorithm takes coverage into account. It can maintain  $k$ -coverage while achieving maximum coverage for a given region while

using the least amount of energy. Low energy usage does not, however, guarantee a long network lifetime. The network lifespan is also impacted by the residual energy distribution. Q-learning follows a reward generation mechanism which is defined based on the agents carrying state variables against a specified set of actions defined in the methodology section of this paper. The proposed network model is scalable, and Q-learning algorithms have been found to be the most suitable for scalable problems [4]. The Q-learning algorithm produces a Q-value against each defined action. The proposed overall algorithm architecture is compared with other state-of-the-art techniques being discussed in the related work section. The predominant objective of this article is to reduce the overall delay of the network by opting for the most suitable CH based on the generated reward by Q-learning. The main contributions of this article are as follows.

- (1) The article considers a quality of service (QoS)-oriented network considering minimization of the delay as the primary objective. The defined network focuses to illustrate the delay minimization by considering packet delivery ratio and packet flow ratio alias throughput.
- (2) The article proposes a reward-based routing mechanism in order to reduce the overall delay in the network with all the encapsulated scenarios in the article. The routing mechanism is performed utilizing the formed CHs in the network. The CH formation is done by utilizing the mathematical model until a threshold simulation is not attained. The gathered information for the “ $t$ ” number of threshold simulations is passed on to the learning mechanism of the proposed Q-learning model. Once a historical diagram of the participant CHs is created, a hybrid mechanism is applied which combines the current selection through the mathematical model, and the reward generated by the proposed Q-learning mechanism and the CH selection is performed on the base of a hybrid score of the node. Table 1 illustrates the notations used in the paper.

The rest of the article is organized in the following manner. Section 2 illustrates the related work which encapsulates route selection methods using machine learning and QoS. The third section presents the detailed work architecture of the proposed hybrid protocol route selection, path optimization, and delay minimization.

## 2. Related Works

The passing of time has witnessed a revolution in the design of sensors used in wireless networks. The sensors that have been advanced are characterized by lightweight, miniature size, and low power feasting. However, the delay observed in the data transmission process remained a critical challenge constantly drawing high attention of the scientific community. The rising impact of WSN in real-world applications has added pace to the research pertaining to various strategies to guarantee network reliability and minimize

TABLE 1: Notations used.

|                   |                                 |
|-------------------|---------------------------------|
| $DE_r$            | Delay efficacy                  |
| $\emptyset$       | Policies                        |
| $A(r)$            | Action                          |
| RREQ              | Route request                   |
| $S(t)$            | State function                  |
| $T\{A(r), S(r)\}$ | Transition function             |
| $W_D$             | Act as a destination node       |
| $W_N$             | Number of nodes                 |
| $W_S$             | Act as a source node            |
| WSN-FR            | Final route from $W_S$ to $W_D$ |
| $\psi$            | Transition <sub>fitness</sub>   |

energy consumption and end-to-end delay during data transfer. However, such issues still penetrate at some step during communication. Förster and Murphy conducted an investigational study of machine learning at all layers in the WSN network stack. A detailed outline of various machine learning algorithms including decision trees, neural networks, and reinforcement learning was provided. The study was mainly referred to for the discussion and implication of the RL algorithm. The existing work illustrates the efforts and applications at each level of the stack [5]. Predictive analytics, as we are all aware, uses methods such as predictive modeling and machine learning to examine historical data and forecast future patterns contrary to conventional forecasting techniques; neural networks are unique. In contrast to a neural network, the most popular model, linear regression, is actually a pretty straightforward approach to problem-solving. Because of their hidden layers, neural networks perform predictive analytics more effectively. Only input and output nodes are used in linear regression models to generate predictions. The hidden layer is also used by the neural network to improve prediction accuracy. That is because it “learns” similarly to how people do. They are prohibitively expensive due to their high computing power requirements. In addition, massive data sets are required to train neural networks, which a company might not have. But as IT technology becomes more affordable, the first obstacle could soon vanish. Soon, there will not be “unpleasant shocks” any more because of technologies such as ANNs. In this respect, Velmani and Kaarthick proposed a simple scheme to mitigate node mobility and delay issues. The proposed cluster-independent data collection tree scheme addressed the challenges of cluster formation and selection of CH based on CH location. The simulation analysis depicted that the scheme provided a better quality of service in terms of throughput, energy consumption, network lifetime, and end-to-end delay [6]. In the same year, Alsheikh et al. presented a review of machine learning algorithms in WSN in which they evaluated the advantages and disadvantages of existing machine learning algorithms. They also outlined the existing challenges while summarizing the adopted machine learning-based methods to address these challenges pertaining to distinct research areas [7]. Han et al. had evaluated CH selection strategies in different routing protocols. The nodes in the existing heterogeneous WSN network were analyzed to investigate when

the first node died and its impact on the number of packets transferred from source to sink. The most important aspect of the research was the evaluation of the monitoring ability of each protocol under study for the selection of the number of CHs in the defined network [8]. Levendovszky and Thai had developed a quality-aware routing algorithm that could perform routing in an energy-efficient manner. This energy efficiency was achieved by selecting some particular nodes in the multihop network for packet transfer. The algorithm achieved energy balancing under various reliability constraints and was also named as the high quality-of-service routing algorithm by the authors due to the selection of near-optimal paths within the network [9]. Chen et al. had addressed various constraints such as high energy consumption, more transmission time, and constraints of path movement that are important to be dealt with for network optimization. Furthermore, the work also led to the reduction in the energy consumption by the node during latency [10]. Wang et al. had adopted a cross-layered routing protocol that was capable of power control and could adapt and coordinate with network dynamics. In the process, researchers had implemented a multiagent Q-learning mechanism which was analyzed using the delay-based nonselfish cost function. The simulation analysis performed using reward functioning improved the throughput and end-to-end latency [11]. Fei et al. made development efforts to address various conflicting optimization criteria. The work was mainly referred to for its metrics curves and concept of multiobjective optimizations. This study also summarized the energy consumption and latency trade-offs along with the lifetime vs. performance trade-offs of existing approaches [12]. Kumar and Kumar used hierarchical clustering in which two swarm intelligence algorithms were integrated one after the other. The artificial bee colony (ABC) was implemented for the selection of CH, and ant colony optimization (ACO) was used for efficient data transmission. The CHs were selected, and subclusters were formed based on the threshold value resulting in an enhanced stability of the designed network architecture [13]. Sun and Park had provided route choice modeling using SVM and neural network architecture and was mainly referred to for distinguishing route prediction accuracy. The predicted route prediction accuracy using NN was 0.6833 and using SVM was 0.7086 with a computing time of 7640.20 s with NN and 602.27 s with SVM [14]. SVM is one of the most effective machine learning algorithms since it is a highly complex and mathematically sound method. It is a dynamic method that can resolve a variety of issues, including regression issues, binary, binomial, and multiclass classification issues, as well as linear and nonlinear issues. By utilizing the idea of the margins and attempting to maximize the distinction between two classes, SVM lowers the likelihood of model overfitting and increases the model's stability. Because kernels are readily available and because SVM is based on fundamental principles, it can operate with ease whenever the data are in a high-dimensional state and are precise in high-dimensional datasets to the point where it can compete with other algorithms such as Naive Bayes, which is adept at handling classification problems with

extremely high dimensions. SVM is renowned for its memory management and processing speed. In particular, when compared to machine learning and deep learning algorithms, with which SVM frequently competes and occasionally even exceeds to this day, it requires less memory. Jin et al. found that in some circumstances, the replacement of batteries becomes too difficult. Therefore, in addition to low network latency and a high transmission rate, an extended network lifetime is a necessity. In this context, the authors had postulated a Q-learning-based delay-aware routing algorithm that could easily adapt to the changing environment in the vicinity of the sensor nodes. Furthermore, an action utility function was defined in which decisions were based on observed delay and residual energy to assure uniform routing. Simulation analysis showed that the mechanism achieved 20% to 25% minimization of end-to-end delay [15]. Jan et al. had addressed the energy utilization issues in WSN. However, balanced energy consumption was achieved at the cost of a slight increase in the end-to-end delay observed during transmission in WSN [16]. Li et al. adopted optimized CH selection to minimize network delay and reduce the energy consumed in the process of the CH selection process. A variable step size firefly algorithm was adopted by the researchers for the identification of the head node to assign CH within each cluster. The objective function was based on a number of parameters such as intracluster distance, probable cluster heads, and residual energy. The delay and the latency observed during data transmission were observed due to the controlled size of data packet transfer. It was observed that lower duty cycle operation followed by efficient cluster formation significantly reduced the network delay and latency of the sensor network [17]. Li et al. had proposed a routing algorithm that could minimize the interference delay during data transmission. Furthermore, it was also associated with efficient management of the energy resources resulting in quality routing [18]. Elappila et al. had presented survival path routing that was based on the survivability factor, the observed interference, and the noise within the traced path from one hop to another. The evaluation studies showed that the designed protocol improved the throughput and packet delivery at the destination with comparatively lesser end-to-end delay. However, the number of packets delivered decreased due to congestion observed during data transmission [19]. Ilamathi and Rangarajan had focused on determining the shortest distant path that was determined using optimization techniques and neural network architecture. The simulation analysis had shown that integration of harmony search as an optimization technique resulted in reduced time complexity and energy consumption [20]. Alghamdi had addressed the challenges of CH selection based on delay, energy, distance, and security. In WSN, optimal CH selection is the key element of data transmission which was addressed using hybrid optimization in which the firefly optimized and replaced the positions and updated them by using the dragonfly. Thus, more refined CH selection was performed resulting in the least computational time of 9102 ms which was consumed using the proposed CH selection work based on hybrid optimization and was analyzed using 1000

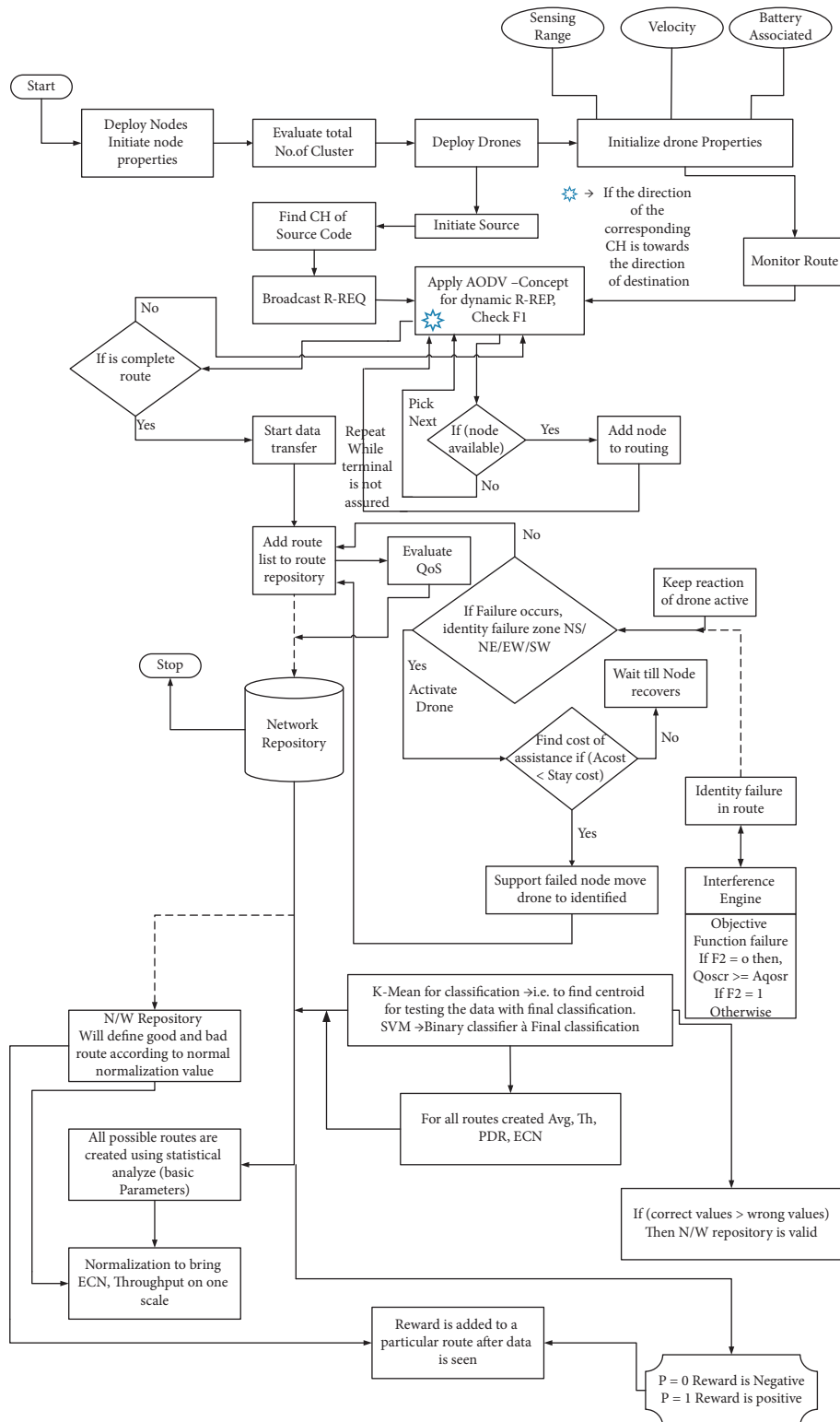


FIGURE 1: Proposed system architecture.

simulation rounds. It was observed that the risk probability, energy, and the number of live node analysis was performed for 2000 simulations, but delay analysis was restricted to only 1000 simulations. It was concluded that further analysis may affect the presented observations if performed for more simulation rounds [21]. Adil et al. adopted a hybrid routing scheme that combines three routing protocols. The CH formation was performed dynamically for a particular time duration using the dynamic cluster-based static routing protocol (DCBSRP) associated with the AODV routing protocol and low-energy adaptive clustering hierarchy (LEACH) protocol. It was observed that static routing limited the number of nodes selected for a fixed interval for the CH selection process. Once that particular interval was completed, the nodes associated with the CH were released, and reassignment of the CH node was performed. The simulation analysis showed that by using the proposed scheme, 95.9% of the total nodes participated in routing that significantly balanced the network load and minimized the transmission delay. The latency of the network using the designed scheme was found to be quite consistent due to the unicast communication of the clusters [22]. On large networks, dynamic routing is seen to be simple to set up. It is also believed to be more user-friendly than static routing in terms of choosing the optimum path, detecting route modifications, and discovering faraway networks. However, because routers constantly exchange updates, they always use more bandwidth than static routing does. The extra burdens caused by routing protocols are also felt by the router's CPUs and RAM. Finally, it is believed that dynamic routing is less secure than static routing. There are several IGP's or inner gateway routing protocols. The protocols listed here are those that might be applied within the network. Every router and server operating system, including Windows 2003 Server and Linux, supports these protocols. So, it follows that the routing protocols are really a collection of languages that the router employs to share routing data with other routers. The capacity of the routing protocol to adapt to changing network topology is its key advantage.

Routes are only created when they are required in the reactive protocol known as ad-hoc on-demand distance vector (AODV). The mobile network is where AODV is most commonly used. One entry is used in the routing table for each destination. In order to ensure that routing information is current and to avoid routing loops, sequence numbers are employed. Reactive protocols, such as AODV, have a tendency to reduce the overhead of control traffic messages. Only the nodes affected by network topological changes receive updates from AODV, which responds to these changes reasonably quickly. The AODV routing protocol conserves energy and storage space. They only reply once to the initial request made by the destination node and disregard the subsequent ones. Each destination has a maximum of one entry in the routing table. AODV grows to a high number of mobile nodes and is loop-free. The processing of routing is handled by AODV without the need for any central management.

Leach is a clustering hierarchy protocol that uses less energy. The leach procedure has various benefits. First, the

cluster heads integrate the whole data, which reduces network traffic overall. Second, since there is only one hop in routing from nodes to cluster heads, there is a significant reduction in energy usage. Thirdly, this protocol enhances and lengthens the network's life. Fourthly, it is not required to know the node's location to construct a cluster. Leach is entirely dispersed and self-organized since it does not require any control of the flow of information from the ground station. The first hierarchical routing protocol that enabled data fusion was the leach protocol. In the clustering routing protocol, this protocol is crucial. In addition to the discussed literature, numerous researchers had combined the multi-path techniques with the machine learning approaches. A handful of the referred machine learning-inspired research publications are further summarized in Table 2.

It has been observed that the selection of CH in each cluster is one of the most challenging tasks to minimize transmission delays and energy consumption. The aforementioned literature shows that the efficacy of CH selection was addressed by numerous researchers to minimize the transmission delay, increase network lifetime, and reduce energy consumption.

### 3. Proposed Algorithm Architecture

The proposed algorithm architecture is a three-phase model. The first phase performs data aggregation based on the statistical approaches of routing. The second phase generates a reward mechanism that utilizes the data aggregated in the first phase using Q-learning architecture, and the third phase incorporates the reward generated in the second phase into the first phase of routing architecture. The overall system architecture can be explained using Figure 1. Simulated on the elastic idea of area and node variation is the suggested network. The energy associated at the beginning of the simulation,  $RE = 100$  mJ, is distributed across the " $n$ " number of nodes in the specified network  $N$ . Each of the four zones that make up the full simulated region is patrolled by a different drone. There might be several clusters in a zone. To save communication overhead, the drones only communicate with the cluster's cluster head (CH). The sensor nodes' deployment locations are chosen at random. In order to adjust for " $n$ " in the network, a total number of CH must be determined before the simulation can begin. For the initial simulation, the maximum RE is used to determine each region's CH. To make the most of the available resources, each node in the proposed simulation environment may receive and send data from other nodes. Source and terminal nodes are created that identify their CH and pass the data packet to the CH carrying packet characteristics in order to comprehend routing.

*3.1. Phase 1: The Routing Architecture and the Data Aggregation with Action Label Generation.* The proposed network is simulated under the elastic concept of area and node variation. The designed network  $N$  has " $n$ " number of nodes with  $RE = 100$  mJ, the amount of energy associated at the start of the simulation, and the Econ model for energy

TABLE 2: Machine learning-inspired work.

| Citation                       | Key elements/<br>techniques                                   | Purpose of<br>article reference              | Routing  | Cluster<br>size                                   | Topological<br>area | Total<br>number<br>of nodes | Deployment<br>type | Packet size      | Clustering | Additional<br>techniques | Alive<br>node | Dead<br>nodes | Energy | Lifetime | Packet-<br>based<br>analysis | Delay/<br>latency | Results and findings   |
|--------------------------------|---|--|--|---|---------------------|-----------------------------|--------------------|------------------|------------|--------------------------|---------------|---------------|--------|----------|------------------------------|-------------------|--|
| Arjunan<br>and Sujatha<br>[25] | ACO and fuzzy<br>logic  | Network<br>division into<br>unequal clusters | Hybrid routing<br>protocol   | Variable<br>(from very<br>small to<br>very large) | 200 m × 200 m       | 300                         | Random             | Not<br>mentioned | Yes        | Yes                      | Yes           | Yes           | Yes    | Yes      | —                            | —                 | Performed first node die<br>and half node die<br>analysis resulting in the<br>following outcomes<br>(i) Improved lifetime<br>(ii) Eliminated hot spot<br>problem<br>(iii) Balanced energy<br>consumption   |
| Gowda and<br>Jayasree<br>[26]  | Hybrid neural<br>network-based<br>energy-efficient<br>routing | Improved PDR<br>and throughput               | HNN_GTOA   | Variable  | 100 m × 100 m       | 500                         | Random             | 512 bytes        | Yes        | Yes                      | —             | —             | Yes    | —        | Yes                          | Yes               | The proposed approach<br>achieved the following:<br>(i) Highest PDR of 96%<br>(ii) Minimum delay of<br>3.7 s<br>(iii) High throughput of<br>1 Mbps   |
| Guo et al.<br>[27]             | Reinforcement<br>learning                                     | RL-based<br>routing                          | RL-based routing   | Variable  | 100 m × 100 m       | 100                         | Random             | 512 bits         | Yes        | —                        | Yes           | —             | Yes    | Yes      | Yes                          | —                 | Energy minimization<br>was achieved using a<br>reward function based<br>defined based on the<br>following elements:<br>(i) Residual energy<br>(ii) Link distance<br>(iii) Hop count<br>Further, Q-learning<br>achieved minimal<br>energy consumption but<br>a relatively lower<br>network lifetime |
| John and<br>Rodrigues<br>[29]  | Taylor series and<br>crow optimization                        | Energy and<br>throughput<br>routing          | Multiojective<br>Taylor Crow<br>optimization<br>(MOTCO)<br>algorithm | Variable  | 50, 100             | 50, 100                     | Random             | Not<br>mentioned | Yes        | Yes                      | Yes           | —             | Yes    | —        | Yes                          | —                 | Despite of decrease in<br>the energy at higher<br>simulation rounds,<br>MOTCO exemplified<br>the following:<br>Throughput at a rate of<br>0.6514 (highest among<br>other algorithms)<br>Energy left at a rate of<br>0.1007 (highest among<br>other algorithms)                                     |
| Huynh<br>et al. [30]           | Delay constraint<br>energy multihop                           | CH selection<br>strategy                     | Energy-efficient<br>cluster-based<br>multihop routing                | Variable  | 100 m × 100 m       | 100                         | Random             | 30 bytes         | Yes        | —                        | —             | Yes           | Yes    | —        | Yes                          | Yes               | Designed energy cost<br>function resulted in<br>lowest cost route<br>selection<br>Intercluster multihop<br>routing resulted in<br>energy-efficient data<br>transfer with minimum<br>end-to-end delay   |

TABLE 2: Continued.

| Citation                | Key elements/<br>techniques  | Purpose of<br>article reference | Routing   | Cluster<br>size | Topological<br>area | Total<br>number<br>of nodes       | Deployment<br>type | Packet size      | Clustering<br>techniques | Alive<br>node | Dead<br>nodes | Energy | Lifetime | Packet-<br>based<br>analysis | Delay/<br>latency | Results and findings   |
|-------------------------|--|---------------------------------|---|-----------------|---------------------|-----------------------------------|--------------------|------------------|--------------------------|---------------|---------------|--------|----------|------------------------------|-------------------|--|
| Elhabyan<br>et al. [28] | Nondominated<br>sorting genetic<br>algorithm II<br>and speed-<br>constrained<br>multiobjective<br>particle swarm<br>optimization | Multiobjective<br>minimization  | SMPSO for<br>clustering and<br>routing protocol<br>(SMPSO-CR)                                   | Variable        | 100 m × 100 m       | 100, 200,<br>300, 400,<br>500     | Random             | Not<br>mentioned | Yes                      | —             | —             | —      | —        | Yes                          | —                 | Simulation analysis<br>showed that SMPSO-CR<br>exhibited higher average<br>throughput for more<br>than 60% of cases<br>SMPSO-CR minimized<br>the number of active<br>CHs per round that<br>reduced the overall<br>energy consumption |
| Chang et al.<br>[29]    | Machine-<br>learning-based<br>parallel genetic<br>algorithm<br>(MLPGA),<br>PCA, K-means  | Multiobjective<br>optimization  | Ultra-reliable and<br>low-latency<br>Internet of Things<br>(uRLLIoT)-based<br>efficient routing | Variable        | 100 m × 100 m       | 50, 100,<br>150, 200,<br>250, 300 | Random             | Not<br>mentioned | Yes                      | Yes           | —             | Yes    | Yes      | —                            | —                 | Simulation analysis<br>showed that<br>involvement of PCA had<br>significantly improved<br>the network<br>performance using<br>MLPGA in terms of<br>network lifetime, energy<br>consumption, reliability,<br>and complexity           |



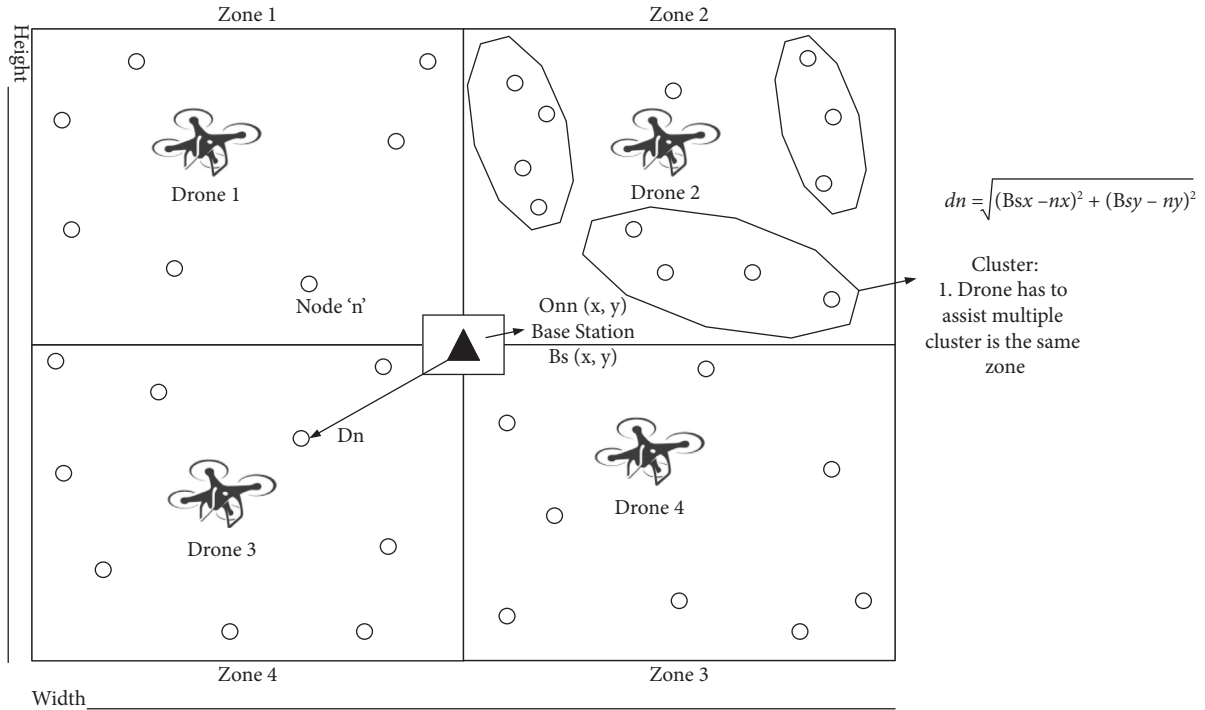


FIGURE 2: Placement model.

TABLE 3: Parameters and their reference articles.

| Parameter under study                       | Referred citation   |
|---|---|
| Balanced energy consumption                 | $tm * ((fij * (1/1e12)) + (fij * (1/1e12 + (100/1e15) * d)))$ ; |
| Data packets, delay, and reward functioning | Wang et al. [11]  |
| Delay                                       | Li et al. [18]  |
| Energy and time complexity                  | Ilamathi and Rangarajan [20]                                    |
| Energy consumption constraints              | Chen et al. [10]  |
| Energy value computation                    | Levendovszky and Thai [9]                                       |
| Number of sensor nodes in a single cluster  | Velmani and Kaarthick [6]                                       |
| Packet transfer rate                        | Elappilla et al. [19]   |
| $D$   | $d = 10^{-((Rssi + A)/(10 * Loss))}$ ;                          |
| $Fij$                                       | $(packet\_count - drop\_count) / packet\_count$                 |

TABLE 4: Node vector message structure.

| Device type | Signal strength | Hop number | Cov list | Neighbour node number | RE left | Distance |
|-------------|-----------------|------------|----------|-----------------------|---------|----------|
|-------------|-----------------|------------|----------|-----------------------|---------|----------|

consumption which is defined in Table 1, and “Ar” is the area of simulation. 4 mobile sinks based on the geographic location of the sensor nodes are shown in Figure 2. The mobile sinks are drones and are there to assist the faulty wireless nodes in the network. CH formation is different from zone formation for the drone in the proposed work. The total number of CHs in the network is decided by the following ordinal measures.

- (a) The entire simulation area is divided into 4 zones, and each zone is covered by one drone
- (b) One zone may have more than one cluster
- (c) The drones communicate only with the cluster head (CH) of the cluster to avoid communication overhead
- (d) The deployment position of the sensor nodes is random

The existing literature referred for various parameters is listed in Table 3.

The simulation starts with the identification of the total number of CH required in order to compensate for “n” in the network. As the RE is similar for each node at the start of the simulation, distance to the base station becomes the only way to select the CH. Drones follow a random walk model to move from one location to another location [23]. Where Econ is the total energy consumption, delay is the propagated delay, r is the total number of simulation rounds, and “n” is the total number of nodes in the network. The objective function aims to minimize the delay produced in each simulation round for every participating node in the list.

The CH selection process utilizes the following equation in order to determine the total number of CHs required for the supplied number of nodes and area [24]:

$$T = \sqrt{\sum_{i=1}^n \frac{\text{distance}_{\text{tobase}} \times \ln(n)}{\text{Network Width/Attraction}_{\text{index}}}} \quad (1)$$

The attraction index is calculated as follows:

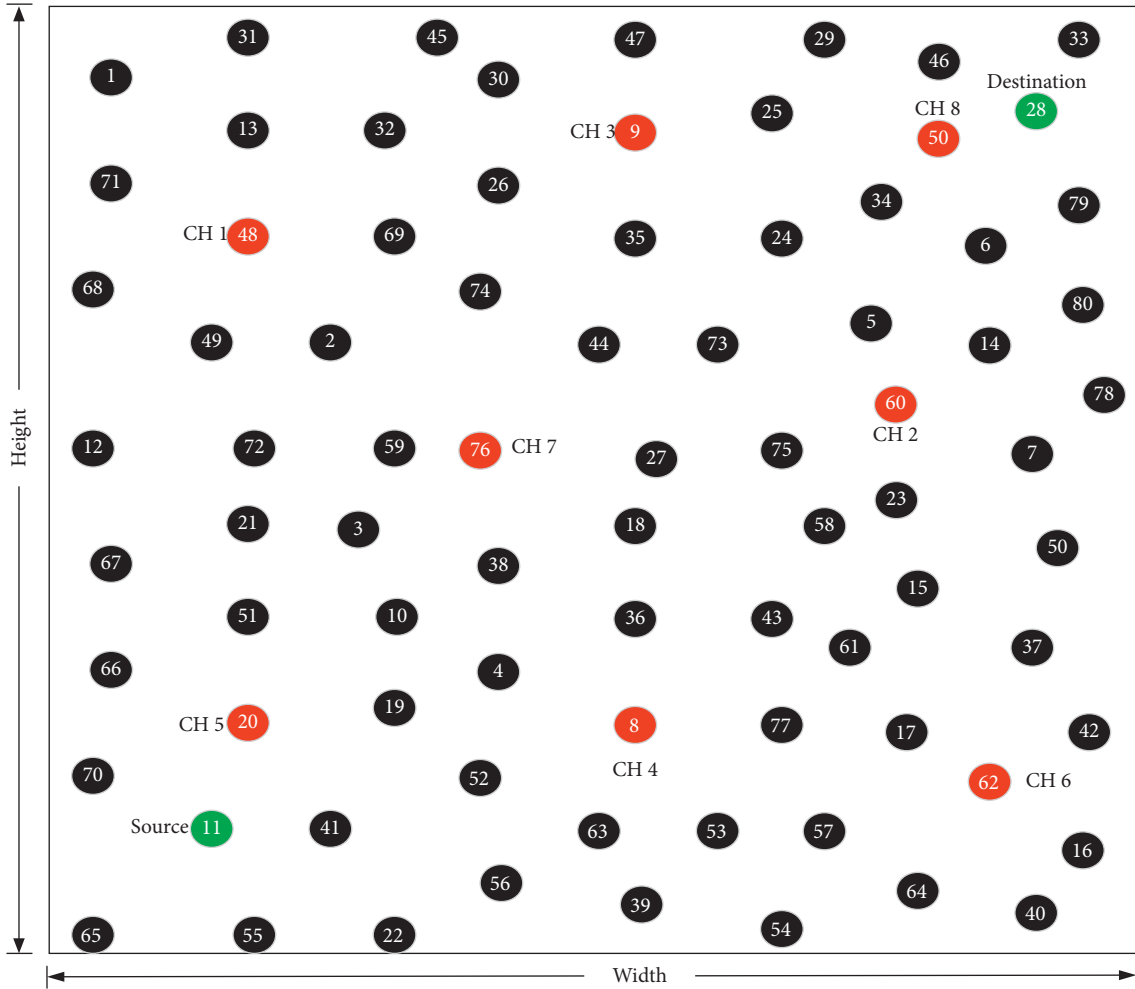


FIGURE 3: Network deployment with the cluster head (CH) selection mechanism.

TABLE 5: Routing loop.

| Source                  | Destination             | CHs                                    |
|-------------------------|-------------------------|--|
| Node 11 (member of CH5) | Node 28 (member of CH8) | Nodes 48, 60, 9, 8, 20, 62, 76, and 50 |

$$\begin{aligned}
 & 2 \quad \text{if } \frac{n}{\text{network width}} < .10, \text{ and } n > 50. \quad (2) \\
 & 1 \quad \text{otherwise}
 \end{aligned}$$

For the first simulation, the CH of every region is selected based on the maximum available RE.

In the proposed simulation environment, each node can receive and transfer data from multiple nodes in order to utilize the resources as much as possible. In order to comprehend routing, source and terminal nodes are generated that identify their CH and forward the data packet to the CH containing packet attributes. This node vector used in the present work is inspired by Lee's work [31] and comprises 7 features as shown in Table 4. In further processing, each feature becomes a part of the

learning architecture. The device type considers the type of device that needs to forward the data (sensor node or drone). RSSI is an abbreviation for the received signal strength indicator. It is an approximated measure of the power level received by an RF client device from an access point or router. At greater distances, the signal weakens, and wireless communication speeds decline, resulting in reduced total data throughput. The receive signal strength indicator (RSSI) measures signal strength and, in most circumstances, displays how effectively a specific radio can hear the remote-linked client radios. RSSI levels are often lower in wider channels. Smaller channel widths are thus advised in all but except for a few exceptional cases when installing EnGenius APs. Signal strength is passed on in the form of RSSI value computed in the algorithm. Hop number is given to count the hop number in that particular route. Cov list is the neighbouring node of the sender node. Neighbour node number gives the number of the next node to which data are being currently forwarded. RE left will measure the left residual energy of the sender node. Distance will depict the current distance between the sender and the current intermediate node to which the data are being forwarded.

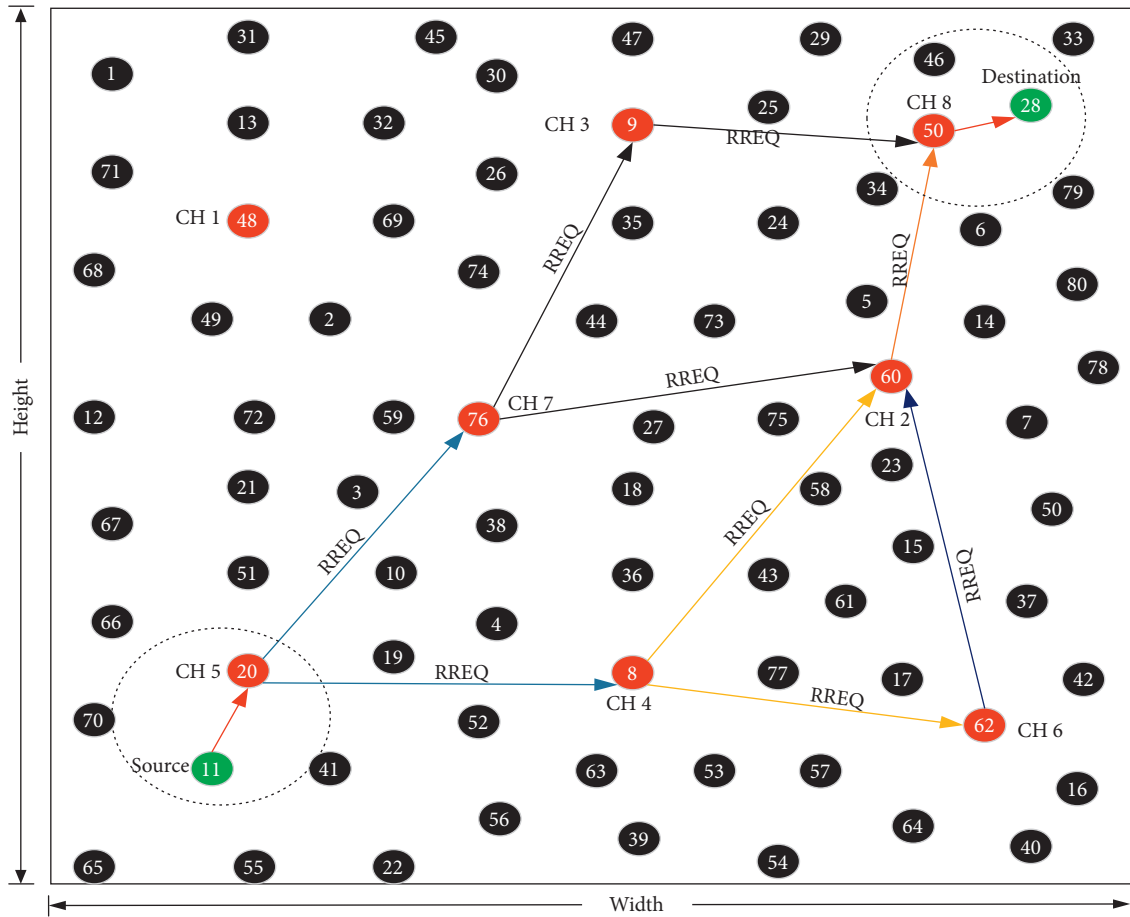


FIGURE 4: Coverage calculation of CH for route discovery.

TABLE 6: CH and their coverage.

| Selected CH                   | Coverage CHs  |
|-------------------------------|---------------|
| CH 1                          | CH 3 and CH 7 |
| CH 2                          | CH 4 and CH 6 |
| CH 3                          | CH 8          |
| CH 4                          | CH 2 and CH 6 |
| CH 5 (containing source)      | CH 4 and CH 7 |
| CH 6                          | CH 2 and CH 4 |
| CH 7                          | CH 2 and CH 3 |
| CH 8 (containing destination) | CH 2 and CH 3 |

The CH uses a broadcast mechanism and follows an ad-hoc on-demand vector routing (AODV) protocol for the dynamic response messages as route replies (R-REP). The process of broadcasting the route request (R-REQ) keeps on rolling in between the CHs until the CH which is associated with the terminal alias destination is not found.

**3.1.1. The Routing Protocol.** The AODV protocol was first developed without any thought to security. Therefore, no protective system was created to recognize the presence of a malicious assault. Due to the fast change in network structure, maintaining a new route to the target is one of

AODV’s essential jobs. The route discovery procedure is used to carry it out. In AODV, the destination node number and the number of hops are crucial characteristics to identifying the route’s freshness, and these characteristics are simple for attackers to influence. Because of this, the security of ad hoc routing protocols such as the AODV is essential, and researchers in MANET from all over the world are always looking for ways to create a routing protocol that is both safe and effective for wireless ad hoc networks. A dynamic reactive routing protocol is what AODV is. A route will be constructed via a reactive routing protocol based on the requirement (upon request by the source node). Route request (RREQ) and route reply are two crucial control messages in AODV route finding (RREP). Both control messages contain a crucial property called the destination sequence number, which has the incremental value to assess the route’s freshness. The proposed routing protocol is inspired by the AODV routing protocol which uses the broadcast mechanism. Cluster-oriented routing has been incorporated to reduce the computation delay of data communication. In order to demonstrate the routing strategy, a working example of 80 nodes is illustrated as follows.

Figure 3 represents the deployment of a network having an area of 1000 m<sup>2</sup> where network height and width are

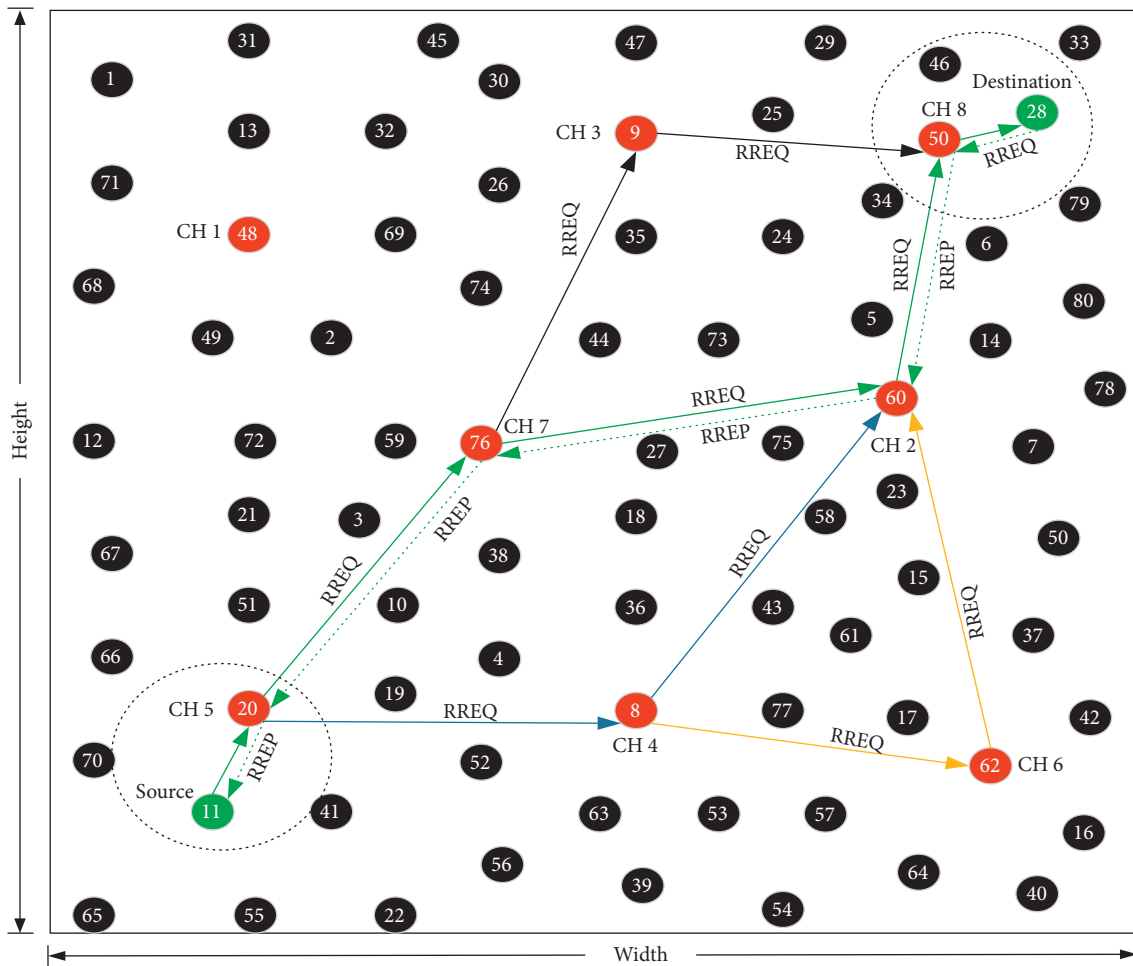


FIGURE 5: Coverage calculation of CH for route discovery.

TABLE 7: Routing table.

| Possible routes   | Member nodes                                |
|-------------------|---|
| Route 1           | [Node 11, CH5, CH7, CH3, CH8, Node 28]      |
| Route 2 (optimal) | [Node 11, CH5, CH7, CH2, CH8, Node 28]      |
| Route 3           | [Node 11, CH5, CH4, CH2, CH8, Node 28]      |
| Route 4           | [Node 11, CH5, CH4, CH6, CH2, CH8, node 28] |

considered as 1000 m. Here, 80 nodes are deployed, and the total network is clustered into 8 segments based on its coverage capability. Afterwards, a cluster head (CH) is selected in each region that is represented by a red color in Figure 3, and their node numbers are 48, 60, 9, 8, 20, 62, 76, and 50. In this scenario, node 11 is considered the source, and node 28 is considered the destination. Here, the source node belongs to CH5, and the destination node

belongs to CH8, and a route formation mechanism is needed to transmit data packets from the source to the destination node. The most suitable and fast routing protocol is used to discover a route from the source to the destination node via CHs based on demand. The used “Request (RREQ)–Reply (RREP)” routing mechanism helps to minimize the transmission delay by avoiding the routing loops according to Table 5.

Here, the source node is the member node of the CH and transfers data packets to CH5. Afterwards, CH5 starts broadcasting a RREQ message packet to neighbour CHs based on the coverage limit. According to Figure 4, the calculated coverage of CHs is given in Table 6.

Here, CH5 starts broadcasting a RREQ message packet to neighbours CH4 and CH7. The structure of the broadcast contains the information about the source CH and the destination CH, and it could be illustrated as (source CH, hops, and destination CH). In the case of the current

example, it would be (CH5, hops, and CH8) CH4 and CH7 that receives this packet and reverts back to CH5 with a RREP = [CH5, Hop-Count, CH4/CH7]. R-REQ indicates the “hello message” having information about source CH and destination CH with their hop count, and RREP is generated after the response of neighbour nodes that indicates the identification of nodes. This process continues until destination CH is not found, and the entire route discovery mechanism is shown in Figure 5.

The CH of Node 11 starts to send routing request packets RREQ to all neighboring CHS to find an optimal path through the calculation of total power consumption and transmission delay. After receiving the RREQ, the intermediate CHs reply to the sender CH by creating a reverse path to the CH of Node 11 with a RREP packet. This process continues and sends the RREQ message to neighbour CH until the message packet is not received by the CH of destination Node 28. At the same time, the destination CH of Node 28 generates a RREP message and transmits this message through the reverse path to the CH of source Node 11. After the CH of source Node 11 receives RREP message packets through various possible routes, a forward route is selected based on the minimum power consumption and transmission delay. There are a total of four possible routes from source to destination via CHs using the routing mechanism. Based on Figure 5, a constructed routing table is given in Table 7.

Based on the concept of minimum power consumption and transmission delay, route 2 is selected as an optimal route for data transmission from the source to the destination node. The algorithmic structure of the protocol is given as follows.

### 3.1.2. AODV Routing Protocol

**Input:**  $W_N \rightarrow$  Number of Node

$W_S \rightarrow$  Act as a Source Node

$W_D \rightarrow$  Act as a Destination Node

**Output:** WSN-FR  $\rightarrow$  from  $W_S$  to  $W_D$  developed a final route

- (1) **Start**
- (2)  $W_N$  start broadcasting an RREQ (Route Request) message to neighbor nodes and CH, Where CH  $\rightarrow$  Cluster Head
- (3) Create an RREQ message, RREQ = [ $W_N$ , Hop Count,  $W_D$ ]
- (4) Initially consider, Hop Count = 0 // because, at starting of route formation, there is no node that can be considered as an intermediate node.
- (5) Consider an array for routing, W-Route = [] // take an empty variable to store route information
- (6) W-Route (1<sup>st</sup> Node) =  $W_S$  // because routing always starts from the source node

- (7) Set a Variable,  $W_D$ -Found = 0 // initially, destination is not founded
- (8) **While** ( $W_D$  -Found  $\neq$  1)
  - (a) 1<sup>st</sup> Node Broadcast RREQ to neighbors is own coverage and record Hop Count
  - (b) Neighbor node receives and analyzes the requirements
  - (c) **If** W-Route (Node [ $W_N$ , Hop Count,  $W_D$ ]) =  $N_N$ [ $W_N$ , Hop Count,  $W_D$ ] // Where Neighbor Node  $\rightarrow N_N$ 
    - (i) W-Route += Neighbor  $N_N$  is a  $W_D$
    - (ii) Each node sends RREP (Route Reply) to  $W_S$  // Feedback sent by a node in coverage of  $N_N$
    - (iii) Hop Count = 1
  - (d) **Else**
    - (i) W-Route = Neighbor  $W_N$
    - (ii) Send RREP to  $W_S$
    - (iii) Hop Count = +1
  - (e) **End-If**
    - (i) Update the Route array and repeat the above steps until  $W_D$  not founded
  - (f) **If** ( $W_D$  = 1)
    - (i)  $W_D$ -Found = 1
    - (ii) **Break**
  - (g) **End-If**
    - (i) Possible Route,  $R = R_1, R_2, R_3, \dots, R_N$
  - (h) **For**  $r = 1 \rightarrow R$ 
    - (i) Current W-Route,  $R = R(r)$
    - (ii) Calculate to distance ( $D$ ) from  $W_S$  to  $W_D$
    - (iii) **If**  $D$  is minimum then
      - (1) WSN-FR =  $R(r)$  // Current W-Route becomes the final route
    - (iv) **Else**
      - (1) Check the next route condition
    - (v) **End-If**
      - (i) **End-For**
  - (9) **End-While**
  - (10) **Return:** WSN-FR as a final route from  $W_S$  to  $W_D$  in WSN
  - (11) **End-Function**

3.1.3. *The Ground Truth.* A formed route is categorized as a good and effective route if the evaluated QoS parameters fit perfectly as per the network demand. The proposed solution evaluates throughput, energy consumption, network delay, and packet deliver ratio (PDR) by using

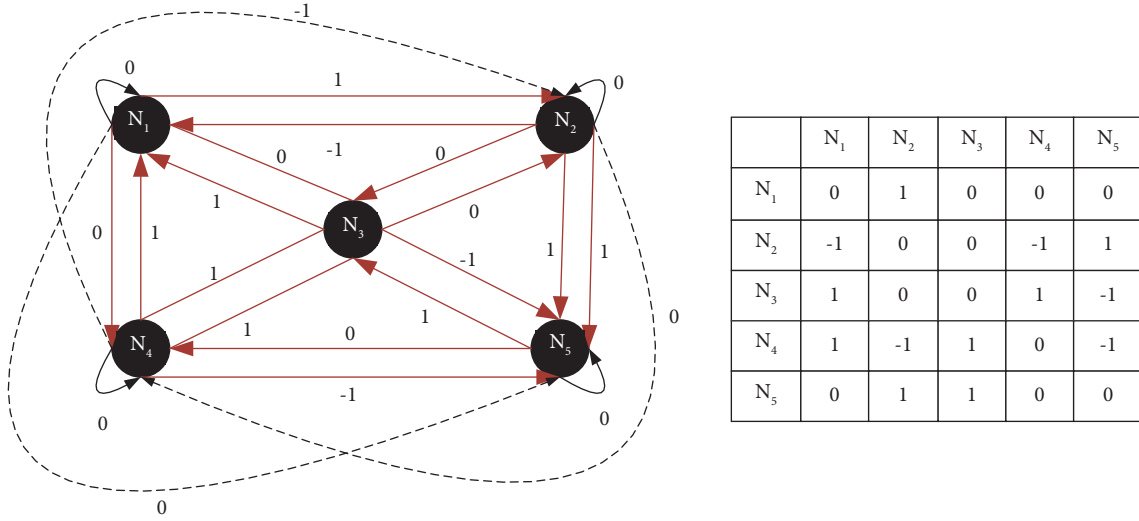


FIGURE 6: Construction of Q-table.

$$\text{Throughput} = \frac{\text{Total}_{\text{Received}}_{\text{packets}}}{\text{Per unit Time}}, \quad (3)$$

$$\begin{aligned} \text{Energy Consumption} &= \sum E_{\text{transmitting}}, E_{\text{receiving}}, E_{\text{amplification}}, E_{\text{aggregation}}, \\ \text{Network Delay} &= \int_{i=1}^r \bigoplus_{i=1}^{P_n} \text{Route}_{ij} \cdot \text{timelaps}(). \end{aligned} \quad (4)$$

where  $r$  is the total number of simulation rounds and  $P_n$  is the total number of participating nodes under one route, and  $\text{timelapse}()$  computes the time interval between the expected arrival time “at” and the actual time of arrival “act.”

$$\text{PDR} = \frac{\text{Received}_{\text{packets}}}{\text{Transferred}_{\text{packets}}}. \quad (5)$$

The aggregated parameters are passed to the  $k$ -means algorithm keeping the attribute list centric centroid in order to reduce the convergence issue of  $k$ -means. Two action labels “Stable” and “NonStable” are considered to be applied to the aggregated data. Suitable and unsuitable action labelling has been performed by calculating the root mean squared value (R-MSE) between the route that is classified under its class and the centroid of the class. The less R-MSE value will indicate the record that belongs to the good class due to its correlation between the data elements.

$$\text{R-MSE} = \int_{l=1}^2 \frac{1}{t} \times \sqrt{\sum_{j=1}^t (x_{ij} - C_{ij})^2}, \quad (6)$$

where  $t$  is the total number of attributes getting used for centroid calculation.  $C$  is the centroid for the respective cluster.

**3.2. Phase 2: Proposed Learning Algorithm for Reward Generation for Optimal Route Selection.** The proposed learning mechanism for reward generation follows the Q-learning

mechanism, and it integrates the support vector machine (SVM) for the classification of the action labels. The action labels are defined as stable and nonstable routes based on the R-MSE calculated in phase 1. The description of naming is deliberated in algorithm description. The learning mechanism is illustrated as follows.

**3.2.1. State.** In the case of the proposed solution, the state variables are collected over “ $r$ ” number of simulations in which “ $rt$ ” is the number of routes that were formed. The proposed solution aggregated the data from the discovered routes defined in the multiroute description in the later section of this phase. In the case of the proposed solution,  $tr$  is equal to 10,000. Hence, in an interval of  $1-r$ , the state is defined as  $S(t) = \int_{i=1}^r Th, pdr, econ, delay dr$ . Two binary indicators alias the ground truth  $SI \in 0, 1$ . When the R-MSE value is higher, the binary indicator is set to be 1 and remains 0 in the other case. The binary indicators are also referred to as action labels in this article.

**3.2.2. Action.** The action is the selection of the CH in the route with the maximum value of the objective function defined as  $A_t(r) = (A_{t,r-1}, R_r)$  where  $R$  is the estimation function that is defined as follows:

$$R_r = \min \left\{ \int_{i=0}^r \int_{j=1}^n d_r, dn, R_{r,d} \max \right\}. \quad (7)$$

Inputs:  $Data_{structure}$  Output: Updated Q – table with reward

- (1) Extract the states  $S(t)$  for  $t$  interval of time // extract the states by collecting the route values as illustrated in the state definition earlier in the article//
- (2) Create Q – repository = [] //create an empty Q-repository which will contain the Q-table and transition results to update the Q-table
- (3) Initiate  $k$  – means, set  $k = 2$  // set the value of  $k = 2$  as there are two action labels namely optimal and nonoptimal
- (4)  $[Route_{index}, Route_{centroid}] = kmeans(Data_{structure} \cdot \int_{t=1}^{itr} data dt, 2)$ . //Apply k-means on the data gathered in  $itr$  time interval with  $k = 2$ , k-means will return route index as the action label is defined as 1 for optimal and 2 for nonoptimal//
- (5) Calculate R – MSE by using equation (7)
- (6) Apply  $\min$  –  $\max$  rule to label classes as optimal and nonoptimal {1, 2} //if the R-MSE is higher, the class is labelled as nonoptimal and vice-versa.//
- (7) Initiate Q – table =  $\phi_{itr=1}^r \int_{j=1}^n CH - CH\{-1, 0, 1\}dr$  //initialize the Q-table as illustrated in Table 4.
- (8) Define T as the transition function dignified under  $\emptyset \in \prod Svm_{rating}$  //SVM rating follows polynomial kernel for hyperplane separation//
- (9)  $(T = Successful | \text{if } \emptyset_{plane\_policy} \cdot \text{maps } Route_{index})$  //the transition is said to be successful if the data gathered as state variables, maps the hyperplane while getting selected for the gradient satisfaction of the policies defined under  $\emptyset$ . //
- (10)  $(T' = Semi - Successful | \text{if } \emptyset_{plane\_policy} \cdot \text{maps } Route_{index} \&\& \emptyset_{plane\_mapping\_policy} \cdot \text{maps } Route_{index})$  where  $T' \in T$ .
- (11) If transition.state == successfull
- (12) Up date Q – repository, Add transition state.value(1) //represents that the route will get maximum reward for this action//
- (13) Else if transition.state == Semi – successfull
- (14) Up date Q – repository, Add transition state.value(0) //represents that the route will get discounted reward for this action.//
- (15) Else
- (16) Up date Q – repository, Add transition state.value(-1) //represents that the route will not get any reward for this action and might also be penalized depending upon the distance of the result to its ground truth.//
- (17) Create  $T\pi\epsilon\pi$  //create a policy T-policy according to the transition actions and update the learning policy//
- (18) //The proposed transition function introduces a semi-successful transition where if the state was selected during the plane policy of the transition function but was unsuccessful under the mapping policy, then the transition is called semi-successful and will get a partial reward as well.
- (19)  $Transition_{sigmoid} = Map.SVM(T, \mu', K)$  //where  $T$  is the transition function, is the selection strategy declared in kernel  $K$  and dignified under  $\emptyset$ . //
- (20) Calculate  $\psi_1$  to  $\psi_4$  //calculate the defined parameters stated under equations (14) to (16).
- (21) Define transition<sub>fitness</sub>  $\psi = (\psi_1 + \psi_4)\psi_2 + \psi_3$  (20)//both neutralized throughput and pdr will surely be high if the delay is low so as the energy consumption.//
- (22) Foreach  $rt$  in  $route_{list}$
- (23) Calculate  $\psi_{current}$  //calculate the transition value of the route in the list by applying equation (20) to the state variable of the current route//
- (24) If  $Transition_{sigmoid} == 1 \&\& \psi_{current} > \psi$
- (25) Reward =  $Transition_{sigmoid} \cdot learning_{rate} + dcb$  //dcb is the direct connection benefit in which if the transition function is satisfied completely, the value of dcb is .1//
- (26) Else if  $Transition_{sigmoid} == 0 \&\& \psi_{current} > \psi$  ||  $Transition_{sigmoid} == 1 \&\& \psi_{current} < \psi$
- (27) Reward =  $Transition_{sigmoid} \cdot learning_{rate} + dcb_2$
- (28) Else
- (29) Reward = 0
- (30) End If
- (31) Update Q-table
- (32) End For
- (33) Find Max.reward(Q – table)
- (34) Choose Route
- (35) Add to List()
- (36) Return List if transition completed.

ALGORITHM 1: Algorithm reward mechanism.

3.2.3. *Reward.* The reward function is defined as the delay efficacy by executing the action  $A(r)$  in the state  $S(t)$ .

$$H(r) = DE_r(A(r), S(r)). \quad (8)$$

Delay efficacy is measured in terms of the difference between the minimum delay and the average delay of the discovered routes up to now. The reward is generated if the

transition function does not cross the limit of delay efficacy (DE).

3.2.4. *Transition Function.* As a result of the data transfer in the network as a given simulation round  $r$ , a transition function  $T(A(r), S(r))$  is evaluated under the architecture of the policies  $\emptyset \in \prod Svm_{rating}$  generated through the training

mechanism of the support vector machine (SVM). Due to the absence of information on the strategies followed by other CHs to choose the next CH in the route list, each CH demonstrates a selfish pattern to select a strategy  $\mu(r)$  to attain maximum reward function. The selection procedure might distort the selection strategy  $\mu'(r)$  which may increase the delay further. This leads us to the formation of a collaborative inference engine to minimize the overall delay. A drone-assisted strategy inference engine is  $\mu_{-r} = (\mu_{r1}, \mu_{r2}, \dots, \mu_{r-t}, \dots, \mu_{r-1}, \mu_r)$ ,  $\forall r \in Y$  where  $Y$  is a finite number.

The process results in knowledge inference of other CHs to select the next CH in the list for the anticipated discount return as follows:

$$\max_{\mu} \left\{ \int_{i=1}^{\text{ch\_count}} \int_{i=1}^r EE_r(A(r), S(r), \emptyset, \mu(r)) \right\}, \quad (9)$$

where CH count is the total number of CHs in the list. The transition function uses the training mechanism borrowed from the utilization architecture of SVM as follows: the kernel has been utilized to form the architecture of the transition function. SVM utilizes kernel function to attain hyperplane separation. The most common kernel is a polynomial kernel expressed in the equation as follows, but it is not recommended for complex and high-volume data.

$$K(x, y) = (\alpha x^T y + c)^d. \quad (10)$$

Slope  $\alpha$ , constant term classification, and the polynomial degree  $d$  are adjustable parameters. The proposed solution utilizes radial basis function (RBF) as the kernel for SVM.

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right). \quad (11)$$

The performance of the RBF kernel depends on the adjustable parameter sigma and should be carefully tuned according to the dataset. If overestimated, the exponential will behave almost linearly, and the higher-dimensional projection will start to lose its nonlinear power. On the other hand, if underestimated, the function will lack regularization, and the decision boundary will be highly sensitive to noise in training data. The learning of the engine is based on a learning rate, which indicates how often the learning agent changes its Q-value.  $\delta = 0$  denotes no learning, but  $\delta = 1$  indicates that new information has totally replaced the old one. The learning rate of the proposed model is stored in a structure containing all the information. As the Bellman equation states that the long-term benefit for a particular activity is equal to the sum of the rewards from the present action and the anticipated rewards from the subsequent acts, the Bellman equation may be used to determine if we have succeeded in maximizing the long-term reward, which is the purpose of reinforcement learning, as it is obvious. Bellman's optimality equation can be written as maximizing

$$q^*(s, a) = E[R_{t+1} + \text{WF} \max_{a'} q^*(s', a')]. \quad (12)$$

This equation states that, at time  $t$ , for any state-action pair  $(s, a)$ , we can get the expected reward  $R_{t+1}$  by choosing

TABLE 8: Parameters used in the simulation.

| Parameters                                    | Values                     |
|---|----------------------------|
| Network dimensions                            | 1000 × 1000 m <sup>2</sup> |
| The initial energy of nodes (residual energy) | 100 mJ                     |
| Total number of simulations                   | 8000                       |
| Nodes   | 50 to 300                  |
| Amplification energy                          | 0.0000001 mJ               |
| $E_{\text{elec}}$                             | 50 nJ/bit                  |
| Packet injection rate                         | 2800 P/sec                 |
| Channel                                       | AWGN                       |
| Packet size                                   | 2000 Bytes/P               |

an action in state  $s$ , in addition to the maximum of “expected discounted return” that is achievable by any  $(s', a')$ , where  $(s', a')$  is a potential next state-action pair, and will be equal to the expected return from starting state  $s$ , taking action  $a$ , and then with the optimal policy afterwards. The key point is that a reinforcement learning system can determine the action that maximizes  $q$  using this equation to get optimum  $q$  and optimal policy  $(s, a)$ . This equation is crucial for this reason. The Bellman optimality equation and the optimal value function are connected iteratively. The Q-table is updated by using Bellman's optimality equation for the reward generation and is defined as follows. The proposed algorithm uses the discount factor by evaluating the record with SVM and is illustrated as follows. The proposed algorithm views every CH as a node and every root as the link edge between the CHs. If there ever exists a route between the node and the CH, the Q-table represents it as 1, and if there does not exist a direct link between a node to another node, it would be represented as  $-1$ .

If there is no direct communication link between node A and node B but communication exists through any other node, then the connection value is 0. In order to illustrate the architecture, a 5-node sample architecture is presented in Figure 6.

As the Q-learning algorithm initially selects random action from the list of policies that are defined by the broadcast mechanism, the transition function provides a discount factor  $\tau$  based on the semisupervised policy imposed by the RBF kernel. If the selected CH is classified as true against its own action label, it would get a discount and will be not penalized. If the selected CH is not classified as true against its own action label, the transition function computes the following equations:

$$\psi 1 = \frac{\sum_{i=1}^{rt d} \text{throughput}_{\text{direct\_connections}^i}}{\sum_{j=1}^{rt} \text{throughput}_j}, \quad (13)$$

$$\psi 1' = 1 - \psi 1, \quad (14)$$

$$\psi 2 = \frac{\sum_{i=1}^{rt d} \text{delay}_{\text{direct\_connections}^i}}{\sum_{j=1}^{rt} \text{delay}_j}, \quad (15)$$

$$\psi 2' = 1 - \psi 2,$$



TABLE 9: The discovery result with proposed Q-learning.

| “Route”             | “Average throughput” | “Average PDR” | “Average consumed energy in mJ” | “Delay in seconds” |
|---------------------|----------------------|---------------|---------------------------------|--------------------|
| [32, 6, 4, 5, 19]   | 4810.2412            | 0.48102412    | 2.07785734                      | 2.07681893         |
| [32, 6, 1, 5, 19]   | 3550.70427           | 0.35507043    | 1.75527254                      | 2.81352634         |
| [32, 6, 2, 5, 19]   | 3281.28783           | 0.32812878    | 1.98639803                      | 3.04453633         |
| [32, 6, 3, 5, 19]   | 3612.52665           | 0.36125267    | 1.70709964                      | 2.76537752         |
| [32, 6, 5, 5, 19]   | 3291.70546           | 0.32917055    | 1.97675782                      | 3.03490094         |
| [32, 6, 6, 5, 19]   | 3383.42715           | 0.33834271    | 1.89444323                      | 2.95262749         |
| [32, 6, 7, 5, 19]   | 3714.24222           | 0.37142422    | 1.63133114                      | 2.68964688         |
| [10, 2, 1, 3, 9]    | 4922.81658           | 0.49228166    | 2.03034073                      | 2.02932607         |
| [10, 2, 1, 3, 7, 9] | 1679.81251           | 0.16798125    | 1.82933359                      | 5.94709228         |
| [10, 2, 2, 3, 9]    | 1617.0014            | 0.16170014    | 2.06045908                      | 6.17810227         |
| [10, 2, 3, 3, 9]    | 1693.52361           | 0.16935236    | 1.7811607                       | 5.89894346         |
| [10, 2, 4, 3, 9]    | 1593.42447           | 0.15934245    | 2.1519184                       | 6.26951588         |
| [10, 2, 5, 3, 9]    | 1619.52722           | 0.16195272    | 2.05081888                      | 6.16846688         |
| [10, 2, 6, 3, 9]    | 1641.42006           | 0.16414201    | 1.96850429                      | 6.08619343         |
| [10, 2, 7, 3, 9]    | 1715.54781           | 0.17155478    | 1.70539219                      | 5.82321282         |
| [8, 2, 1, 3, 43]    | 5782.65255           | 0.57826526    | 1.72844467                      | 1.72758088         |
| [8, 2, 2, 3, 43]    | 1238.94573           | 0.12389457    | 1.75856303                      | 8.06330719         |
| [8, 2, 3, 3, 43]    | 1283.37739           | 0.12833774    | 1.47926464                      | 7.78414838         |
| [8, 2, 4, 3, 43]    | 1225.05727           | 0.12250573    | 1.85002234                      | 8.15472079         |
| [8, 2, 5, 3, 43]    | 1240.428             | 0.1240428     | 1.74892282                      | 8.0536718          |
| [8, 2, 6, 3, 43]    | 1253.23056           | 0.12532306    | 1.66660823                      | 7.97139834         |
| [8, 2, 7, 3, 43]    | 1295.98581           | 0.12959858    | 1.40349613                      | 7.70841774         |
| [28, 5, 3, 6, 42]   | 5973.0656            | 0.59730656    | 1.67334425                      | 1.672508           |
| [28, 5, 1, 6, 42]   | 990.256549           | 0.09902565    | 1.65889784                      | 10.0882948         |
| [28, 5, 2, 6, 42]   | 968.088472           | 0.09680885    | 1.89002333                      | 10.3193048         |
| [28, 5, 3, 6, 42]   | 995.005453           | 0.09950055    | 1.61072494                      | 10.040146          |
| [28, 5, 4, 6, 42]   | 959.587958           | 0.0959588     | 1.98148264                      | 10.4107184         |
| [28, 5, 5, 6, 42]   | 968.993244           | 0.09689932    | 1.88038312                      | 10.3096694         |
| [28, 5, 6, 6, 42]   | 976.788232           | 0.09767882    | 1.79806853                      | 10.2273959         |
| [28, 5, 7, 6, 42]   | 1002.5676            | 0.10025676    | 1.53495643                      | 9.96441534         |
| [11, 6, 4, 5, 48]   | 4757.46686           | 0.47574669    | 2.10090691                      | 2.09985698         |
| [11, 6, 1, 5, 48]   | 752.03812            | 0.07520381    | 1.7783221                       | 13.2839011         |
| [11, 6, 2, 5, 48]   | 739.183554           | 0.07391836    | 2.0094476                       | 13.5149111         |
| [11, 6, 3, 5, 48]   | 754.773873           | 0.07547739    | 1.73014921                      | 13.2357523         |
| [11, 6, 4, 5, 48]   | 734.217376           | 0.07342174    | 2.10090691                      | 13.6063247         |
| [11, 6, 5, 5, 48]   | 739.710927           | 0.07397109    | 1.99980739                      | 13.5052757         |
| [11, 6, 6, 5, 48]   | 744.244829           | 0.07442448    | 1.9174928                       | 13.4230022         |
| [11, 6, 7, 5, 48]   | 759.117294           | 0.07591173    | 1.6543807                       | 13.1600216         |
| [31, 3, 7, 4, 33]   | 6286.90312           | 0.62869031    | 1.58981216                      | 1.58901765         |
| [31, 3, 1, 4, 33]   | 612.749427           | 0.06127494    | 1.71375356                      | 16.3035648         |
| [31, 3, 2, 4, 33]   | 604.188503           | 0.06041885    | 1.94487905                      | 16.5345748         |
| [31, 3, 3, 4, 33]   | 614.564402           | 0.06145644    | 1.66558066                      | 16.255416          |
| [31, 3, 4, 4, 33]   | 600.866532           | 0.06008665    | 2.03633836                      | 16.6259884         |
| [31, 3, 5, 4, 33]   | 604.540795           | 0.06045408    | 1.93523884                      | 16.5249394         |
| [31, 3, 6, 4, 33]   | 607.565709           | 0.06075657    | 1.85292425                      | 16.442666          |
| [31, 3, 7, 4, 33]   | 617.440932           | 0.06174409    | 1.58981216                      | 16.1796853         |

$$\psi_3 = \frac{\sum_{i=1}^{rtd} Econ_{direct\_connections_i}}{\sum_{j=1}^{rt} Econ_j}, \quad (16)$$

$$\psi_3' = 1 - \psi_3, \quad (17)$$

$$\psi_4 = \frac{\sum_{i=1}^{rtd} pdr_{direct\_connections_i}}{\sum_{j=1}^{rt} pdr_j}, \quad (18)$$

$$\psi_4' = 1 - \psi_4, \quad (19)$$

where rtd is the total number of discovered routes with all the nodes having a direct connection to each other as per the Q-table explained in Table 4 and rt is all the discovered routes. The proposed transition function aims to optimize delay as reward utilizing throughput, PDR, and energy consumption considering that raise in delay will necessarily provide a raise in the energy consumption. Looking at the other scenario, a sustainable PDR can be attained via a sustainable route itself, and hence, energy consumption is

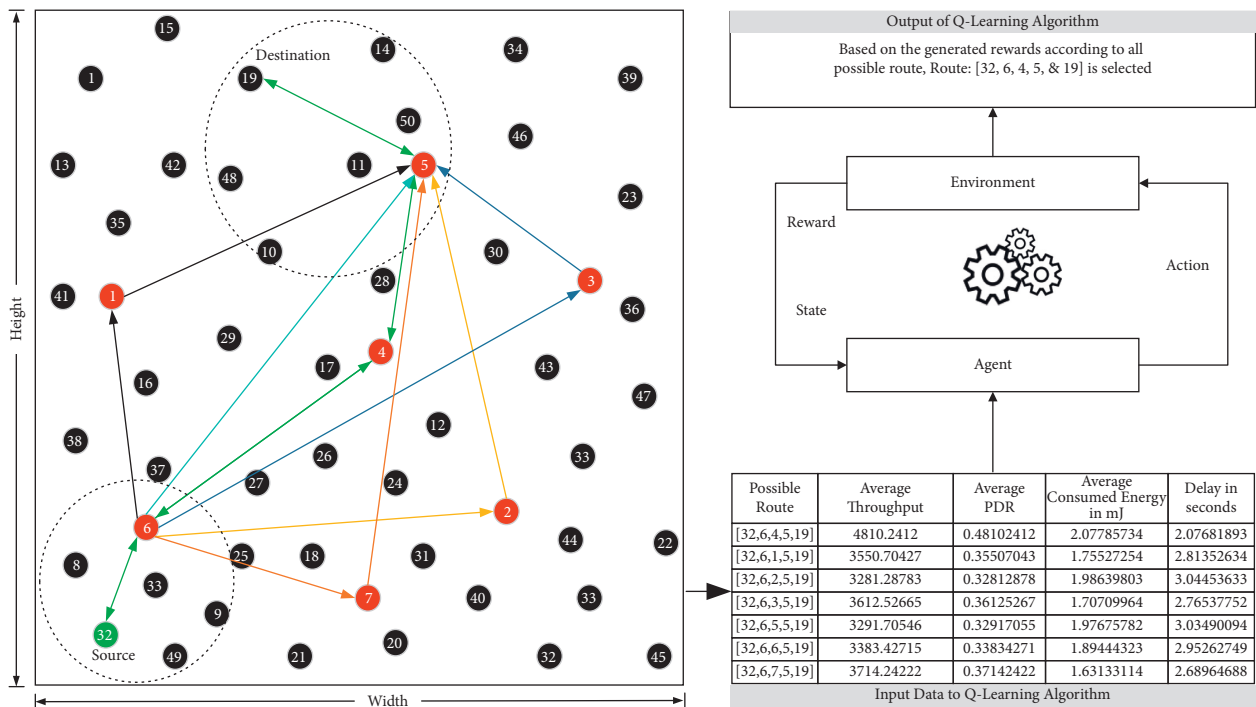


FIGURE 7: Q-learning-based optimal route selection.

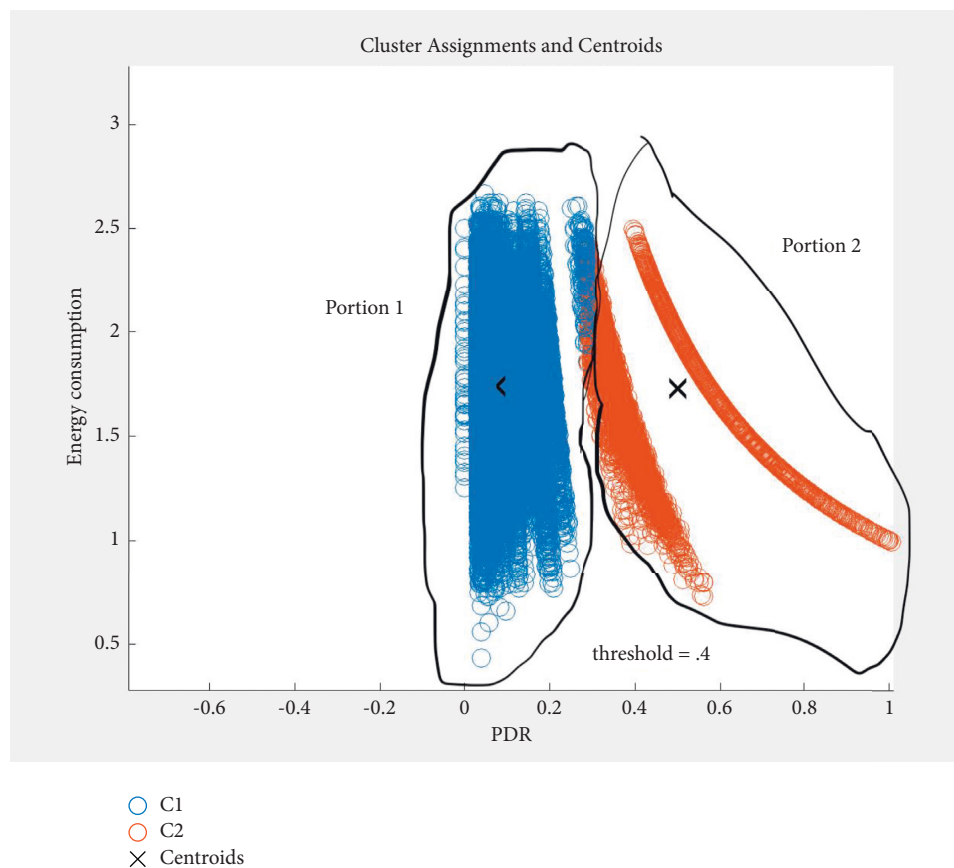


FIGURE 8: K-means cluster distribution.

TABLE 10: The index value of the discovered routes.

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 |   |
| 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | — | — | — |

TABLE 11: Centroid values.

| Throughput    | PDR        | Energy consumed | Delay      |
|---------------|------------|-----------------|------------|
| C1:5036.61675 | 0.50366167 | 1.72692851      | 2.16897609 |
| C2:767.734244 | 0.07677342 | 1.73980045      | 18.2631773 |

TABLE 12: SVM evaluation measures.

| Support vectors | Support vector indexes | Data scale shift | Scaling value |
|-----------------|------------------------|------------------|---------------|
| 0.94006433      | -0.3865938             | 2                | -1998.9299    |
| 0.83872792      | 0.34073533             | 7                | -1.8292453    |
| -0.1933212      | 0.00046155             | 10               | Scale factor  |
| -0.2165793      | 0.72779066             | 15               | 0.0006058     |
| -0.1716727      | -0.6472768             | 16               | 5.22616563    |
|                 |                        |                  | 0.61053863    |
|                 |                        |                  | 0.07607094    |
|                 |                        |                  | 0.04786985    |

TABLE 13: Classified learning and the rate of learning.

| Itr 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|-------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2     | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1     | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
|       |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | - |
|       |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | - |

directly proportional to produced delay. The proposed transition function is to take maximum reward in order to reduce the chances of a delay getting reduced through the most sustainable route [18]. The generation of the reward mechanism is illustrated by the algorithm reward mechanism as follows (Algorithm1).

With the updated set of Q-table, the proposed algorithm selects the route with maximum reward and transfers the data through the route.

### 4. Results and Analysis

The result section is illustrated in two subsequent segments. The first segment illustrates the working behavior analysis for Q-learning and its update in the Q-table, and the second segment illustrates the effects of the Q-learning mechanism on the quality-of-service (QoS) parameters. The parameters are evaluated on the base of throughput, PDR, and delay. The preliminary objective is to minimize the overall delay, and hence, segment 2 is more illustrated towards the delay architecture. Simulation parameters of the proposed work are summarized in Table 8.

4.1. Segment 1: The Q-Learning Behavior Results and Illustrations. The architecture system for Q-learning can be explained by utilizing the Q-table. In order to illustrate the

same, a selected route architecture system with Q-learning is presented in Table 9. The selected routes have demonstrated the least delay in each selection of Q-learning through the Q-table. Each possible discovered route is treated as a candidate in the table. To show the separation of the route, we have underlined the end of that particular route having multiple paths, and the route with the minimum delay is selected as shown and is marked in red color.

Figure 7 represents the route formation and optimal route selection mechanism in the network. In the network, total 7 routes are possibly based on the QoS parameters that is shown in the first segment of Table 9, and possible routes are (32, 6, 4, 5, 19), (32, 6, 1, 5, 19), (32, 6, 2, 5, 19), (32, 6, 3, 5, 19), (32, 6, 5, 5, 19), (32, 6, 6, 5, 19), and (32, 6, 7, 5, 19). Here, nodes 32 and 19 are the source and destination, respectively. In the figure, a total of possible 7 routes are represented with different colors, but the selected route is marked with a bidirectional green color. Based on the reward mechanism using the Q-learning algorithm, an optimal route (that is represented by green color) is selected for data transmission within the network.

If the possible discoveries are passed to the k-means architecture system, as explained in pseudocode, it will be divided into two subsequent segments as shown in Figure 8.

49 routes were separated into two clusters by the k-means algorithm providing the index of each record as follows.



TABLE 15: Updated vector values using learning.

| Support vectors |            |                  |            | Scaling value |
|-----------------|------------|------------------|------------|---------------|
| 0.94006433      | -0.3865938 | Data scale shift |            | 0.63267223    |
| 0.83872792      | 0.34073533 | -1662.9294       | -1.7373876 | 0.63267223    |
| -0.1933212      | 0.00046155 |                  |            | -2.3843431    |
| -0.2165793      | 0.72779066 | Scale factor     |            | 0.63267223    |
| -0.1716727      | -0.6472768 | 0.0005181        | 3.19949267 | 0.63267223    |

TABLE 16: Throughput comparison of varying number of nodes.

| Number of nodes | Throughput proposed | Throughput Adil et al. [22] | Throughput Janakiraman et al. [32] | Throughput Tong et al. [33] | Throughput Jothikumar et al. [34] |
|-----------------|---------------------|-----------------------------|------------------------------------|-----------------------------|-----------------------------------|
| 50              | 3245                | 3101.4322                   | 3089.663                           | 3081.7748                   | 3097.776                          |
| 60              | 3446.17543          | 3112.39412                  | 3192.3479                          | 3134.48563                  | 3166.24921                        |
| 70              | 3637.52473          | 3192.03396                  | 3231.97788                         | 3216.00814                  | 3289.3447                         |
| 80              | 3644.766            | 3303.93115                  | 3288.22571                         | 3286.8048                   | 3413.21594                        |
| 90              | 3891.6457           | 3400.33877                  | 3488.9855                          | 3376.86181                  | 3530.3617                         |
| 100             | 4080.65244          | 3812.00244                  | 3674.8769                          | 3401.97335                  | 3709.4399                         |
| 110             | 4174.58523          | 4173.87104                  | 3760.50581                         | 3646.94716                  | 3875.78546                        |
| 120             | 4376.41123          | 4193.92941                  | 3940.53937                         | 3906.92483                  | 4091.9419                         |
| 130             | 4708.72529          | 4318.81971                  | 4087.94395                         | 3942.71978                  | 4241.58929                        |
| 140             | 5353.72388          | 4949.13731                  | 4315.18153                         | 4098.72728                  | 4479.95819                        |
| 150             | 5876.14379          | 5532.5628                   | 4424.02818                         | 4425.70629                  | 4731.42408                        |
| 160             | 6363.63952          | 6290.71477                  | 4504.16529                         | 4778.94179                  | 4998.89386                        |
| 170             | 6492.0629           | 7108.35414                  | 4920.28003                         | 4861.02262                  | 5226.9789                         |
| 180             | 6707.19731          | 7274.11871                  | 5305.95674                         | 5206.73921                  | 5503.06103                        |
| 190             | 7504.93633          | 7496.69471                  | 5608.26372                         | 5323.83566                  | 5818.46378                        |
| 200             | 7853.86691          | 7626.83521                  | 5709.70214                         | 5354.57379                  | 6179.06194                        |
| 210             | 8805.59068          | 7867.88746                  | 6222.60822                         | 5809.81012                  | 6397.3553                         |
| 220             | 8839.68004          | 8072.96055                  | 6696.45979                         | 5901.37601                  | 6729.67178                        |
| 230             | 8947.1148           | 8097.97255                  | 6805.44776                         | 5969.44892                  | 7118.51948                        |
| 240             | 9344.39954          | 8515.6455                   | 7198.29703                         | 6527.6943                   | 7377.55619                        |
| 250             | 10518.6957          | 8754.0543                   | 7702.78221                         | 6962.35297                  | 7756.21863                        |
| 260             | 10925.9141          | 9441.32893                  | 7879.73576                         | 7262.5191                   | 8206.20773                        |
| 270             | 12199.5986          | 10012.5662                  | 7937.42459                         | 7473.36552                  | 8570.05995                        |
| 280             | 13135.9515          | 11128.2879                  | 8072.55283                         | 7547.07922                  | 8856.04343                        |
| 290             | 13491.2721          | 11797.5584                  | 8379.82703                         | 7656.42254                  | 9270.27223                        |
| 300             | 14653.2772          | 12548.9491                  | 9005.73384                         | 7823.11966                  | 9734.57875                        |

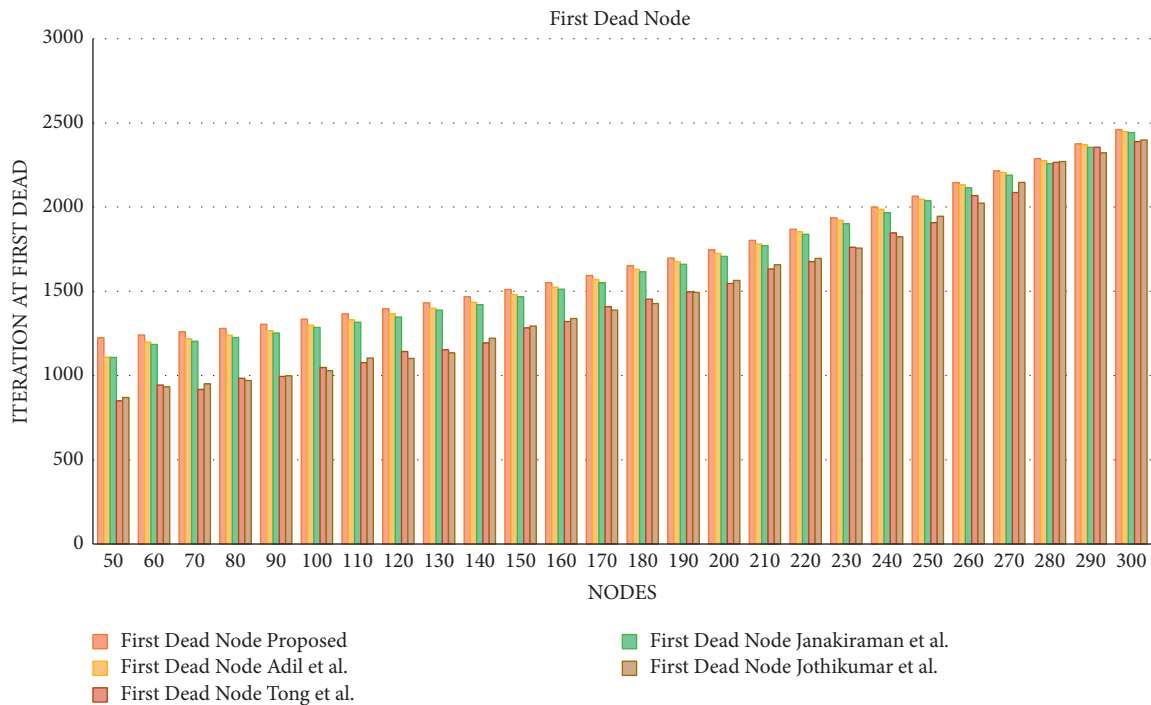


FIGURE 9: First dead.

TABLE 17: First dead node analysis.

| Total number of nodes | First dead node proposed | First dead node Adil et al. [22] | First dead node Janakiraman et al. [32] | First dead node Tong et al. [33] | First dead node Jothikumar et al. [34] |
|-----------------------|--------------------------|----------------------------------|---|----------------------------------|--|
| 180                   | 1652                     | 1631                             | 1616                                    | 1454                             | 1428                                   |
| 190                   | 1698                     | 1676                             | 1661                                    | 1497                             | 1494                                   |
| 200                   | 1747                     | 1725                             | 1708                                    | 1547                             | 1565                                   |
| 210                   | 1803                     | 1781                             | 1772                                    | 1633                             | 1658                                   |
| 220                   | 1869                     | 1854                             | 1839                                    | 1677                             | 1696                                   |
| 230                   | 1937                     | 1922                             | 1902                                    | 1762                             | 1757                                   |
| 240                   | 2001                     | 1987                             | 1967                                    | 1847                             | 1824                                   |
| 250                   | 2065                     | 2047                             | 2039                                    | 1908                             | 1946                                   |
| 260                   | 2146                     | 2132                             | 2115                                    | 2069                             | 2024                                   |
| 270                   | 2215                     | 2206                             | 2190                                    | 2087                             | 2147                                   |
| 280                   | 2288                     | 2276                             | 2258                                    | 2266                             | 2271                                   |
| 290                   | 2376                     | 2371                             | 2355                                    | 2356                             | 2322                                   |
| 300                   | 2459                     | 2448                             | 2443                                    | 2389                             | 2399                                   |

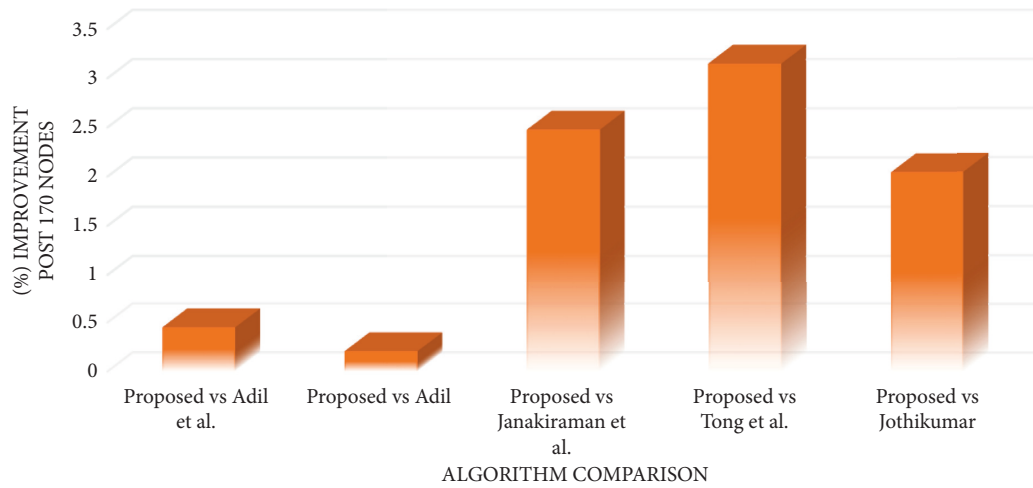


FIGURE 10: 30% improvement after 170 nodes.

TABLE 18: Area 1000 × 1000.

| “Node count” | “Delay proposed in ms” | “Delay Adil et al. in ms” [22] | “Delay Jankiraman et al. in ms” [32] | “Delay Tong et al.” [33] | “Delay Jothikumar et al. in ms” [34] |
|--------------|------------------------|--------------------------------|--------------------------------------|--------------------------|--------------------------------------|
| 50           | 32.2873679             | 36.932468                      | 39.1538966                           | 39.2373847               | 39.7768266                           |
| 70           | 33.1606471             | 37.7698548                     | 38.351566                            | 39.8051611               | 39.8076779                           |
| 90           | 30.9266673             | 34.9384715                     | 37.3427305                           | 39.3521663               | 37.3883953                           |
| 120          | 27.3980742             | 27.9429802                     | 36.465805                            | 36.9402624               | 37.3669213                           |
| 150          | 25.4706602             | 26.8721583                     | 33.5027698                           | 32.9370081               | 37.9056775                           |
| 170          | 23.0747941             | 24.2530223                     | 32.1434133                           | 31.9786625               | 36.8844247                           |

TABLE 19: Area 1200 × 1200.

| “Node count” | “Delay proposed in ms” | “Delay Adil et al. in ms” [22] | “Delay Jankiraman et al. in ms” [32] | “Delay Tong et al.” [33] | “Delay Jothikumar et al. in ms” [34] |
|--------------|------------------------|--------------------------------|--------------------------------------|--------------------------|--------------------------------------|
| 50           | 32.6179758             | 36.4508125                     | 39.1111538                           | 40.3419632               | 39.4658325                           |
| 70           | 32.4941836             | 36.5501697                     | 37.1592111                           | 39.8756931               | 40.1298338                           |
| 90           | 30.8102917             | 35.0508881                     | 38.1256756                           | 38.5018943               | 37.7813995                           |
| 120          | 27.8814697             | 28.5773844                     | 36.4270542                           | 35.8252124               | 37.4886949                           |
| 150          | 27.0045548             | 28.5593033                     | 33.9712062                           | 34.2376274               | 36.5340843                           |
| 170          | 23.1033163             | 24.888012                      | 31.9534859                           | 31.5716343               | 35.8185035                           |

TABLE 20: Area  $1500 \times 1500$ .

| “Node count” | “Delay proposed in ms” | “Delay Adil et al. in ms” [22] | “Delay Jankiraman et al. in ms” [32] | “Delay Tong et al.” [33] | “Delay Jothikumar et al. in ms” [34] |
|--------------|------------------------|--------------------------------|--------------------------------------|--------------------------|--------------------------------------|
| 50           | 35.0423311             | 40.8867587                     | 41.4966643                           | 42.6025533               | 44.1136766                           |
| 70           | 36.4971308             | 41.7499193                     | 41.6635964                           | 41.597064                | 43.0154386                           |
| 90           | 35.4264717             | 38.7476383                     | 39.911122                            | 43.0485669               | 41.5262606                           |
| 120          | 30.9057897             | 32.5120918                     | 38.297832                            | 40.3138084               | 41.1345227                           |
| 150          | 30.7798016             | 30.6571412                     | 37.2154995                           | 37.4275949               | 41.471187                            |
| 170          | 25.4352464             | 28.8872285                     | 35.8655314                           | 35.5298542               | 39.3456913                           |

TABLE 21: Area  $1800 \times 1800$ .

| “Node count” | “Delay proposed in ms” | “Delay Adil et al. in ms” [22] | “Delay Jankiraman et al. in ms” [32] | “Delay Tong et al.” [33] | “Delay Jothikumar et al. in ms” [34] |
|--------------|------------------------|--------------------------------|--------------------------------------|--------------------------|--------------------------------------|
| 50           | 36.7651755             | 41.6580115                     | 43.4463408                           | 44.0581323               | 43.8102033                           |
| 70           | 37.4555381             | 42.5361457                     | 41.7203543                           | 44.1986077               | 44.5245097                           |
| 90           | 34.7386653             | 39.6169688                     | 41.8785567                           | 43.2973583               | 41.6281601                           |
| 120          | 32.3024881             | 32.1662301                     | 40.8012008                           | 41.7540711               | 43.3890894                           |
| 150          | 30.4238857             | 31.2090686                     | 38.5450214                           | 38.2791028               | 42.5260459                           |
| 170          | 26.3347152             | 29.6358258                     | 36.1113005                           | 36.1144816               | 41.6276264                           |

K-means divides the aggregated data into two segments. “X” marks the centroid of the clusters. The  $x$ -axis contains the PDR  $\in \{0,1\}$ , and the energy consumption is calculated in kW which is plotted on the  $y$ -axis. It is evident from the distribution that one segment contains the PDR values that are just below the 0.4 mark.

The divided index can be seen in Table 10, and the formed centroids are given in Table 11.

The separated class is treated as states for the routes, and the action is set to be  $\forall R_A \in \{0,1\}$  where  $R_A$  is the action taken against the state variable. To modify the current policy update rule that is applied using Bellman’s equation, the separated data are passed to SVM for binary distributed labeling of the routes.

The trained SVM provides the training architecture as illustrated in Table 12.

The update over the trained sample based on the selected policy of learning is defined under the Q-learning architecture, and the following results have been attained that are displayed in Tables 13 and 14.

When compared to the ground truth values of the data, it is evident that the learning rate gets stable in the 3<sup>rd</sup> iteration, and more than 90% of data are classified correctly. The scaling values of SVM also get updated using the modified learning system and are presented in Table 15.

In order to demonstrate the performed algorithm action, a sample of 49 discoveries has been presented. The proposed algorithm has utilized more than 71361 discovered routes. Amplified learning helps in route optimization as false classified routes are not pursued for long intervals, and hence, QoS does not get affected. This results in optimal QoS values as illustrated in the second segment of this section.

*4.2. Segment 2: The Improvisation in QoS.* The QoS parameters are evaluated in order to check the effects of the proposed algorithm over the routes. The evaluation contains

two parameters, namely, the throughput and delay. As the network remains to sustain when the nodes are alive, and hence, first dead node and the last dead node for 5000 simulations have also been noted and are illustrated as a part of the overall result. The proposed work is compared to other state-of-the-art techniques that are illustrated in the literature as well. The evaluation has been conducted for 5000 test simulations apart from the data aggregation simulations. The comparison of throughput and PDR is demonstrated as an overall architecture of the increasing number of nodes. The delay is illustrated with the variation in size and variation in nodes as well to support the elasticity of the network.

The throughput is evaluated under a standard  $1500 \times 1500$  network layout. As it is evident from Table 16 that with the increasing number of nodes, every algorithm demonstrates an increase in the throughput. It is obvious that if the number of nodes will increase, the number of carriers to data packets will also increase, and hence, the table demonstrates an uprising behavior. The proposed algorithm stands high in terms of throughput in comparison to other state-of-the-art techniques. The significant improvement in throughput is due to the proposed algorithm architecture which boosts the route selection policy by embedding the classification system to discoveries. The proposed routing policies enable the sensor nodes to serve more in the network, and as a result, the average throughput is most efficient for the proposed simulation setup. The most second efficient algorithm next to the proposed algorithm is observed to be the routing policy proposed by Adil et al. For 300 nodes, the proposed algorithm provided 14653 packets/second, whereas Adil et al. resulted in 12548 packets/second. In terms of improvement, the proposed algorithm is 14.77% more efficient in terms of throughput as compared to Adil et al.

The evaluation of the first dead node and the last dead node is evaluated for 5000 iterations, but the maximum sustainability was attained at 3500

iterations. Figure 9 represents the evaluation of the first dead node.

As displayed in Figure 9, as the number of nodes increases, the active number of participants increases, and hence, less energy is consumed per node. It further results in less amount of battery drainage and fewer dead penalties. In comparison to other state-of-the-art techniques, the proposed algorithm performs significantly better. A margin of 8-9% has been attained prior to 170 nodes in the list. After 170 nodes, it has been observed that the growth in attaining efficiency becomes static in terms of dead nodes. When compared to Adil et al., the percentage difference is quite low after 170 nodes and is illustrated in Table 17.

After 170 nodes, it has been observed that the growth in attaining efficiency becomes static in terms of dead nodes. When compared to Adil et al., the percentage difference is quite low after 170 nodes and is illustrated in Table 16 and is presented in Figure 10.

This research article aims to minimize the overall propagation delay, and hence, the delay is evaluated under different varying area sizes as well. The evaluation has been performed keeping 4 different area sizes i.e.,  $\{1000 \times 1000, 1200 \times 1200, 1500 \times 1500, 1800 \times 1800\}$ . The purpose is to check the elasticity of the proposed algorithm against the other state-of-the-art algorithms. The tabular analysis is presented in Tables 18–21.

The evaluated delay values are computed in ms. As the network deployment area increases, the propagation delay increases throughout the network. As with the increasing number of nodes and as the participants are more available to transfer the data, the delay decreases in the deployment area size, but it is also evident that increase in the area raises the propagation delay. As for example, the delay for the proposed algorithm in area size  $1800 \times 1800 m^2$ , with a full stack node of 170, is noted to be 26.334 ms where as for the same simulation setup with decreased area by 200 units viz. in the area size of  $1500 \times 1500 m^2$  the delay is noted to be 25.435 ms which is approximately 1 ms less than next area size. This pattern could also be observed in other algorithmic architectures, but still the proposed algorithm outcasts the existing algorithms by more than 3 sec on an average in any overall scenario which marks an improvement of 7% from the existing algorithms.

## 5. Conclusion

In the present work, an improved route discovery mechanism is proposed based on the Q-learning model of reinforcement learning. This reward-based learning mechanism is used to significantly minimize the overall communication delay observed during communication performed over long distances in WSN. To evaluate the work, deployment areas are varied from  $1000 m^2$  to  $1800 m^2$  along with variation in the number of nodes deployed in the network. The number of nodes here varies from 50 to 300 that provided a comprehensive investigation of the designed mechanism. In addition to delay analysis, throughput, PDR, and first dead node analysis were also performed to justify the effectiveness of the enhanced routing mechanism to deliver a quality of service with minimal communication delay. The practical

difficulties of creating long-distance communication may now be overcome by adopting hop-oriented routing networks thanks to technical improvements. However, the quality of service (QoS), particularly in terms of significant communication latency, is often deteriorated by long-distance data transmissions. As a result, in the work that is being given, a reward-based routing system that tries to minimize the total latency and is assessed under various circumstances provided. Up until a threshold simulation was not reached, the routing method entailed fine-tuning the CH selection mechanism based on a mathematical model. To provide a high-quality service, examples of CH coverage estimations are also given for potential paths between the source and the destination. Based on this knowledge, the Q-learning model's learning process receives the data obtained from previous simulations. In order to obtain the shortest possible transmission delay, the job is assessed in terms of throughput, PDR, and the first dead node. Area variation is also investigated in order to determine how an increase in the deployment area and node count affects the Q-learning-based approach designed to reduce latency. The success of the suggested method in terms of throughput, the first dead node, and delay analysis is justified by a comparison to four other research studies. Q-learning is a prominent technique right now since it lacks modeling. Deep learning may be used to help the Q-learning model as well. Numerous artificial neural networks used in deep learning choose the appropriate weights to get the best potential answer. Deep Q-learning is a type of Q-learning using neural networks. These methods help firms make significant progress in job completion and decision making.

## Data Availability

The data shall be accessed on request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] G. S. Varsa and D. Sridharan, "A performance overview of contemporary hierarchical clustering algorithms in wireless sensor networks," *International Journal of Communication Networks and Distributed Systems*, vol. 27, no. 1, pp. 1–50, 2021.
- [2] M. Driberg, V. S. Asirvadam, and F. C. Zheng, "Accurate delay analysis in prioritised wireless sensor networks for generalized packet arrival," *IEEE Wireless Communications Letters*, vol. 3, no. 2, pp. 205–208, 2014.
- [3] S. Chopra, G. Dhiman, A. Sharma, M. Shabaz, P. Shukla, and M. Arora, "Taxonomy of adaptive neuro-fuzzy inference system in modern engineering sciences," in *Computational Intelligence and Neuroscience*, S. H. Ahmed, Ed., vol. 2021, , pp. 1–14, Hindawi Limited, 2021.
- [4] J. Wang, Z. Ren, B. Han, J. Ye, and C. Zhang, "Towards understanding cooperative multi-agent q-learning with value factorization," *Advances in Neural Information Processing Systems*, vol. 34, 2021.



- [5] A. Förster and A. L. Murphy, "Machine learning across the WSN layers," *Emerging Communications for Wireless Sensor Networks*, pp. 165–183, 2010.
- [6] R. Velmani and B. Kaarthick, "An energy efficient data gathering in dense mobile wireless sensor networks," *International Scholarly Research Notices*, vol. 2014, 2014.
- [7] 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.
- [8] G. Han, X. Jiang, A. Qian, J. J. Rodrigues, and L. Cheng, "A comparative study of routing protocols of heterogeneous wireless sensor networks," *The Scientific World Journal*, vol. 2014, 2014.
- [9] J. Levendovszky and H. N. Thai, "Quality-of-service routing protocol for wireless sensor networks," *Journal of Information Technology & Software Engineering*, vol. 133, no. 2, p. 37, 2015.
- [10] Y. Chen, Z. Wang, T. Ren, and H. Lv, "Lifetime optimization algorithm with mobile sink nodes for wireless sensor networks based on location information," *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, Article ID 857673, 2015.
- [11] K. Wang, T. Y. Chai, and W. C. Wong, "Routing, power control and rate adaptation: a Q-learning-based cross-layer design," *Computer Networks*, vol. 102, pp. 20–37, 2016.
- [12] 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 Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 550–586, 2017.
- [13] R. Kumar and D. Kumar, "Hybrid swarm intelligence energy efficient clustered routing algorithm for wireless sensor networks," *Journal of Sensors*, vol. 2016, pp. 1–19, 2016.
- [14] B. Sun and B. B. Park, "Route choice modeling with support vector machine," *Transportation Research Procedia*, vol. 25, pp. 1806–1814, 2017.
- [15] Z. Jin, Y. Ma, Y. Su, S. Li, and X. Fu, "A Q-learning-based delay-aware routing algorithm to extend the lifetime of underwater sensor networks," *Sensors*, vol. 17, no. 7, p. 1660, 2017.
- [16] N. Jan, N. Javaid, Q. Javaid et al., "A balanced energy-consuming and hole-alleviating algorithm for wireless sensor networks," *IEEE Access*, vol. 5, pp. 6134–6150, 2017.
- [17] C. Li, H. Niu, M. Shabaz, and K. Kajal, "Design and implementation of intelligent monitoring system for platform security gate based on wireless communication technology using ML," in *International Journal of System Assurance Engineering and Management* Springer Science and Business Media LLC, 2021.
- [18] J. Li, B. N. Silva, M. Diyan, Z. Cao, and K. Han, "A clustering based routing algorithm in IoT aware Wireless Mesh Networks," *Sustainable Cities and Society*, vol. 40, pp. 657–666, 2018.
- [19] M. Elappila, S. Chinara, and D. R. Parhi, "Survivable path routing in WSN for IoT applications," *Pervasive and Mobile Computing*, vol. 43, pp. 49–63, 2018.
- [20] K. Ilamathi and P. Rangarajan, "Intelligent computation techniques for optimization of the shortest path in an asynchronous network-on-chip," *Cluster Computing*, vol. 22, no. S1, pp. 335–346, 2019.
- [21] T. A. Alghamdi, "Energy efficient protocol in wireless sensor network: optimized cluster head selection model," *Telecommunication Systems*, vol. 74, no. 3, pp. 331–345, 2020.
- [22] M. Adil, R. Khan, J. Ali, B. H. Roh, Q. T. H. Ta, and M. A. Almaiah, "An energy proficient load balancing routing scheme for wireless sensor networks to maximize their lifespan in an operational environment," *IEEE Access*, vol. 8, pp. 163209–163224, 2020.
- [23] C. Li, H. Niu, M. Shabaz, and K. Kajal, "Design and implementation of intelligent monitoring system for platform security gate based on wireless communication technology using ML," *Int J Syst Assur Eng Manag*, vol. 13, no. S1, pp. 298–304, 2022.
- [24] L. Kapoor, S. Bawa, and A. Gupta, "Peer clouds: a P2P-based resource discovery mechanism for the Intercloud," *International Journal of Next-Generation Computing*, pp. 153–164, 2015.
- [25] S. Arjunan and P. Sujatha, "Lifetime maximization of wireless sensor network using fuzzy based unequal clustering and ACO based routing hybrid protocol," *Applied Intelligence*, vol. 48, no. 8, pp. 2229–2246, 2018.
- [26] C. S. Gowda and P. V. Y. Jayasree, "Rendezvous points based energy-aware routing using hybrid neural network for mobile sink in wireless sensor networks," *Wireless Networks*, vol. 27, no. 4, pp. 2961–2976, 2021.
- [27] W. Guo, C. Yan, and T. Lu, "Optimizing the lifetime of wireless sensor networks via reinforcement-learning-based routing," *International Journal of Distributed Sensor Networks*, vol. 15, no. 2, Article ID 155014771983354, 2019.
- [28] R. Elhabyan, W. Shi, and M. St-Hilaire, "A Pareto optimization-based approach to clustering and routing in Wireless Sensor Networks," *Journal of Network and Computer Applications*, vol. 114, pp. 57–69, 2018.
- [29] 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, 2018.
- [30] T. T. Huynh, A. V. Dinh-Duc, and C. H. Tran, "Delay-constrained energy-efficient cluster-based multi-hop routing in wireless sensor networks," *Journal of Communications and Networks*, vol. 18, no. 4, pp. 580–588, 2016.
- [31] Y. Lee, "Classification of node degree based on deep learning and routing method applied for virtual route assignment," *In Ad Hoc Networks*, vol. 58, pp. 70–85, 2017.
- [32] S. Janakiraman, "An energy-proficient clustering-inspired routing protocol using improved Bkd-tree for enhanced node stability and network lifetime in wireless sensor networks," *International Journal of Communication Systems*, vol. 33, no. 16, p. e4575, 2020.
- [33] M. Tong, Y. Chen, F. Chen, X. Wu, and G. Shou, "An energy-efficient multipath routing algorithm based on ant colony optimization for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 6, Article ID 642189, 2015.
- [34] C. Jothikumar, R. Venkataraman, T. Sai Raj, J. Selvin Paul Peter, and T. Y. J. Nagamalleswari, "EUCOR: an efficient unequal clustering and optimal routing in wireless sensor networks for energy conservation," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 5, pp. 9187–9195, 2021.