Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8874632, 12 pages
https://doi.org/10.1155/2021/8874632

Research Article

FQ-MEC: Fuzzy-Based Q-Learning Approach for Mobility-Aware Energy-Efficient Clustering in MANET

Naghma Khatoon,[1] Prashant Pranav,[1] Sharmistha Roy,[1] and Amritanjali[2]

[1] *Faculty of Computing and Information Technology, Usha Martin University, Ranchi, India*
[2] *Birla Institute of Technology, Mesra, Ranchi, India*

Correspondence should be addressed to Naghma Khatoon; naghma.bit@gmail.com

Received 25 September 2020; Accepted 16 July 2021; Published 26 August 2021

Academic Editor: Juan A. Gómez-Pulido

Different schemes have been proposed for increasing network lifetime in mobile ad hoc networks (MANETs) where nodes move uncertainly in any direction. Mobility awareness and energy efficiency are two inescapable optimization problems in such networks. Clustering is an important technique to improve scalability and network lifetime, as it relies on grouping mobile nodes into logical subgroups, called clusters, to facilitate network management. One of the challenging issues in this domain is to design a real-time routing protocol that efficiently prolongs the network lifetime in MANET. In this paper, a novel fuzzy-based Q-learning approach for mobility-aware energy-efficient clustering (FQMEC) is proposed that relies on deciding the behavioral pattern of the nodes based on their stability and residual energy. Also, Chebyshev's inequality principle is applied to get node connectivity for load balancing by taking history from the monitoring phase to increase the learning accuracy. Extensive simulations are performed using the NS-2 network simulator, and the proposed scheme is compared with reinforcement learning (RL). The obtained results show the effectiveness of the proposed protocol regarding network lifetime, packet delivery ratio, average end-to-end delay, and energy consumption.

1. Introduction

Wireless communication plays a major role in the previous few years and has become one of the most focused areas of the communication research world. Such communication is useful in establishing a kind of network which is completely established on the fly. One of the challenging issues in multihop ad hoc wireless networks is developing such a routing protocol, which can efficiently handle the frequently changing topology. In MANET, clustering and routing are the major problems that are taken into consideration. An efficient clustering algorithm has been designed in this paper where node selection for different responsibilities has been done using a fuzzy-based Q-learning approach of reinforcement learning, and loads on cluster heads are balanced using Chebyshev's inequality principle.

MANET is a collection of mobile nodes which establish a network very spontaneously. They share the information; basically, they communicate with each other through a shared wireless communication channel. Thus, MANET does not require any preexisting infrastructure. They are purely based on rapidly deployable infrastructure-less services which are self-controlled, and they do not require any preexisting centralized administrator and are configured with very few resources available.

Clustering is a significant methodology that takes care of numerous issues of MANET and gives network scalability and expands its lifetime. Here, nodes are divided into virtual groups called clusters. It makes hierarchical routing possible where paths are established between clusters. Each cluster has a cluster head (CH), which serves as a local coordinator for its cluster. CHs are prudently selected from the set of ordinary nodes, which can retain their role of coordination for a longer period compared with other nodes, i.e., CHs should be less mobile with high energy than the other cluster members (CMs). Communication from source to destination is done via CHs and gateway (G) nodes, which are within the transmission range of more than one CH (Figure 1).

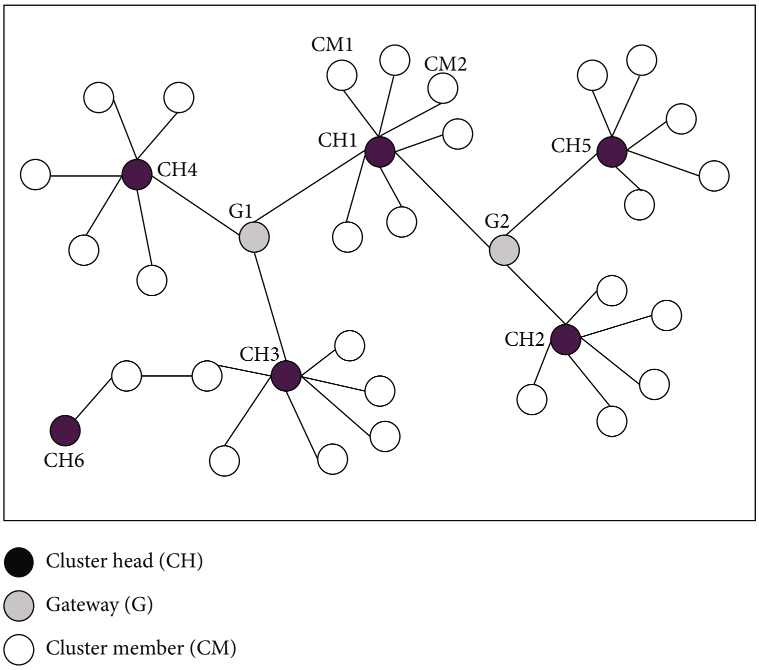

Figure 1: Categories of node's responsibilities in clustering.

With clustering, routing of packets can be more easily managed as the route setup is confined only to the CHs and gateway nodes. The energy dissipation of other nodes can be reduced to a greater extent. Also, it conserves communication bandwidth as nodes need to communicate only with their CHs thus reducing the overhead of redundant messages of disseminating routing packets. However, clustering requires CHs to have extra workload, and thus, prudential selection of a node's role of responsibility should be of utmost importance. A mobility-aware energy-efficient clustering algorithm is proposed in this paper using a reinforcement learning technique of Fuzzy Q-Learning to decide the role of nodes to become a CH, CM, or gateway wherein Chebyshev's inequality principle is applied for getting the node's connectivity for load balancing by taking history from the monitoring phase. Our proposed work consists of two phases wherein for the setup phase, the most eligible nodes are selected to act as CHs. The stability of nodes to become a CH is decided by considering their mobility, direction of motion, distances with other nodes, and their degree of connectivity. The setup phase takes two parameters to decide CHs which can retain their role for a longer duration of time, i.e., stability deviation and energy depletion of nodes. The focus is given on the monitoring phase of clustering, where the reinforcement learning method of Fuzzy Q-Learning with Chebyshev's inequality principle is used to improve the learning accuracy in deciding the node's role of responsibility and balancing loads on them.

The main contribution of our research is scrutinized in the given manner:

(i) Increasing stability of clusters by taking node's mobility and their direction of motion

(ii) Enhancement in the monitoring phase of clustering by reinforcement learning, i.e., Fuzzy Q-Learning

Autoscaling of load on CHs is maintained by using Chebyshev's inequality principle:

(i) The proposed protocol is flexible to be tuned with different network scenarios by changing the fuzzy membership functions and fuzzy rules

(ii) Simulation results show the efficacy of the proposed work on various parameters

The rest of the paper is organized as below. Section 2 presents a brief survey of related works. System model and terminologies related to the proposed work, including energy model and stability model, are described in Section 3 with the detailed background of the Fuzzy Q-Learning approach of reinforcement learning. The proposed work is described in Section 4. The effectiveness of the proposed work is emphasized in the results and discussion part of Section 5. Finally, the article is concluded in Section 6 with some directions for future work.

2. Related Works

Authors in [1] have proposed a model based on fuzzy logic and trust to enhance the security in MANETs and remove vulnerabilities and attacks. It has been observed that due to the distributive nature of MANET, it is vulnerable to various types of attacks, and trust is considered an essential requirement. Trust of the proposed model is generated using two different perspectives, and its value varies between 0 and 1. The objective of this model is to reduce the effect of vampire attacks, which improves the performance of the model. The proposed model is compared with an existing model of trust, and the accuracy of the model in finding the malicious nodes is measured with three parameters, namely, precision, recall, and communication overhead.

Wireless sensor nodes are considered an effective technology for gathering data from any unreachable and dangerous location, such as civil and military areas. The major problem of wireless sensor networks lies with the energy constraint, as the radio transceiver consumes more energy than the hardware components. Thus, designing of routing algorithms with optimized energy is an utmost requirement for longevity of network lifeline. Authors in [2] have designed three parameters related to energy optimization, namely, degree of closeness of node with respect to shortest path; degree of closeness of node to sink, either through single-hop or multihop; and degree of balance of energy. The values of these parameters are fed to a fuzzy logic-based routing algorithm to optimize energy, thus increasing the lifetime of the network, which also helps in effective data transmission between sensor nodes.

Two main important issues that are faced in a target-based wireless sensor network (WSN) are coverage and connectivity, which are considered essential for the transferring of data from the target area to the base station. However, the challenge lies in placing sensor nodes in potential positions to satisfy both coverage and connectivity. Authors in [3] have tried to resolve this NP-complete problem using a genetic-based algorithm scheme. The GA approach used in this paper considers suitable chromosome representation, fitness function derivation, and operations like selection, crossover, and mutation. Result shows that the proposed algorithm provides better time complexity in comparison to other GA-based approaches.

Wireless sensor networks face a key issue in localization, i.e., in locating the sensor nodes precisely. Precision plays a vital role in transmitting data effectively between sensor nodes. Authors in [4] have proposed a fuzzy-based localization algorithm. The proposed algorithm is based on the weighted centroid localization technique, which uses inference systems like Fuzzy Mamdani and Sugeno for calculating the location of unknown nodes. After precise calculation of the position of the sensor nodes, effect selection is done for the next-hop CH, which helps in energy reduction and increases the lifetime of the sensor nodes.

The authors in the paper [5] proposed a heuristic and distributed route, which deploys a new methodology to enhance the QoS requirement for MANETs. A distributed route searching scheme is combined with an RL (reinforcement learning) technique. The results of the proposed approach, compared to the traditional approach, show that the network performance has been improved with optimized timing, effectiveness, and efficiency.

In [6], the authors have presented a rich new realm for multiagent reinforcement learning. They have identified an important but overlooked realm in which nodes have no control over their movements, and as such, a theoretical bound has been presented to improve the connectivity of partially mobile networks. The empirical result showed that a lot has to be done to design a more strong movement learning algorithm that will deal directly with the partial observability of the existing realm.

The authors in [7] have focused on employing the reinforcement learning approach that enables us to achieve adaptive routing in MANETs. To carry the study forward, the authors have selected various Markov Decision Processes (MDP) that provide an explicit formalization of the routing problem. The future scope of the paper is to do a comparative study in the same field, which will be of significant interest.

In [8], the authors propounded an intelligent routing control algorithm for MANET, which was based on reinforcement learning. The employed algorithm can optimize the selection strategy of nodes through interplay with an environment and coverage with the optimal transmission pathway. The result depicts that in comparison to the other algorithms, the offered algorithm could select an appropriate pathway under restricting conditions and can also obtain better optimization objectives. Further research can be done to study the issue of intelligent network routing control based on Q function.

In the paper [9], the authors have presented a Collaborative Reinforcement Learning (CRL) approach that will allow groups of reinforcement learning agents to resolve optimization problems in dynamics as well as in decentralized networks. The result depicts how feedback in the selection links by routing agents allows the sample to modify and optimize its routing behavior to varying network situations and properties, proceeding to optimization of the network output. A future study can be done on the sampling protocol that will be engaged in tuning its configuration criterion to resolve default values for MANET environments.

The authors in the study [10] took the parameters to associate stability with route shortness, and reinforcement learning was employed to propound an approach to make the selection among the neighbors at any time to pass on the packet to the destination. The aim of the approach was to predict the behavior pattern of the nodes in association to get the node through utilizing reinforcement learning. The whole process used a Q-learning algorithm, which could calculate the value of actions. The result depicts the supremacy of the proposed approach over the MANET routing models. In a further study, the packet delivery rate as well as the time taken can be improved by the selection of better alternatives and policies.

In paper [11], a biobjective intelligent routing protocol has been proposed with an objective to reduce an expected long- run cost function consisting of an end-to-end delay and the pathway energy cost. For this purpose, a multiagent reinforcement learning-based algorithm has been set forth to estimate the optimal routing policy in the deficiency of the knowledge about the system statistics. The result showed that the model-based approach used in the study outperforms the model-free alternatives and drives nearly to the standard value iteration that expects perfect statistics.

The authors of the paper [12] have introduced a trust enhancement method to the MANET. The approach used in the study was based on the Reinforcement Learning Trust Manager (RLTM). The result showed that the generated reinforcement learning has high accuracy in making assumptions on calculating the trust levels.

3. System Model and Terminologies

A set of mobile nodes is considered for the MANET model, where nodes are deployed in a certain fixed geographical area and are free to move randomly in any direction. The nodes are assigned with unique IDs which are broadcasted using a Carrier Sense Multiple Access MAC layer protocol (CSMA/CA). Thus, each node in the MANET model becomes aware of its neighboring nodes within its transmission range through periodic transmissions of "HELLO" packets and maintains a neighbor information table. The nodes can start communicating over the wireless link as they come within the transmission range of each other. The list of terminologies that have been used to formulate the proposed work is described as follows:

(i) Let $N = \{n_1, n_2 \cdots, n_l\}$ be the set of mobile nodes

(ii) $\Psi = \{ch_1, ch_2 \cdots, ch_m\}$ be the set of elected CHs such that $(l > m)$

(iii) $T_{\text{range}}(ch_a)$ is the transmission range of a CH ch_a

(iv) $W(n_i)$ is the weight of a node n_i

(v) $N(n_i)$ is the cardinality of a node n_i

(vi) $\text{dist}(n_i, n_j)$ is the Euclidean distance between nodes n_i and n_j

(vii) M is the number of rules

(viii) $\mu_i(x)$ is the firing strength of rule i for input signal x

(ix) $Q(s, a)$ is the state-action pair of the Q Table

(x) $V(s')$ is the value of new state s'

3.1. Energy Model. In wireless ad hoc networks, the energy model is one of the important system attributes. To assess the node's remaining battery energy at the time of simulation, the proposed algorithm used the basic energy consumption model determined by the class EnergyModel in the NS-2 network simulator [13]. The various attributes used are initialEnergy, rxPower, txPower, sleepPower, and idlePower representing the energy of a node at the beginning, energy consumed in receiving one packet, energy consumed in transmitting one packet, energy consumed in the sleep state, and energy consumed in the idle state, respectively. Energy consumption of a node n_i at time interval Δt is given by

$$E_{\text{cons}}(v_i, \Delta t) = E_{\text{residual}}(v_i, t_0) - E_{\text{residual}}(v_i, t_1), \tag{1}$$

where $E_{\text{residual}}(v_i, t_0)$ and $E_{\text{residual}}(v_i, t_1)$ denote the residual energy of node v_i at times t_0 and t_1, respectively.

3.2. Stability Model. The node's movement in such a network is difficult to forecast. However, it produces position traces for the mobile nodes. When nodes move in a particular terrain, the traces are going to capture the different positions at different instants of time and these traces can help in understanding the performance of different protocols designed for MANET.

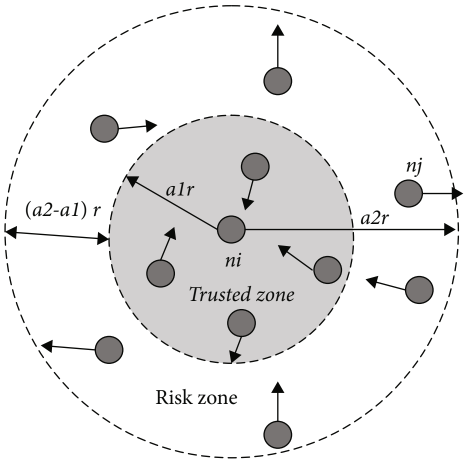

FIGURE 2: Transmission range zone with direction of mobility.

To choose the most qualified nodes for CHs, their combined weight of various parameters have been utilized. We are giving more focus on the mobility of nodes in the selection of CHs and deciding how stable a cluster should be because mobility is the major concern in such a network, which may lead to frequent reclustering and link updating. The range of transmission of any node (say) can be divided into a trusted zone or a risk zone [14]. The inner circle with radius $\alpha 1 r$ forms the trusted zone, and the zone having width $r(\alpha 2 - \alpha 1)$ forms the risk zone, as shown in Figure 2. The coefficients $\alpha 1$ and $\alpha 2$ are reasonably chosen depending on the mobility of nodes in the network system that we have proposed in our previous work [15].

There are different models which can be adapted to predict mobility, which are to determine how appropriate a node is to carry on the role of CH, its relative mobility distances with neighbors, and consideration of the number of neighbors in its direct communication range. The relative mobility is determined as dependent on the received signal strength between two progressive "HELLO" packets. This is inversely proportional to the distance between the sender and the receiver. The relative mobility at node n_i with respect to node n_j, $M^{\text{Rel}}_{n_i}(n_j)$, is calculated as

$$M^{\text{Rel}}_{n_i}(n_j) = 10 \log_{10} \frac{RxPr^{\text{new}}_{n_j \longrightarrow n_i}}{RxPr^{\text{old}}_{n_j \longrightarrow n_i}}, \tag{2}$$

where $RxPr^{\text{new}}_{n_j \longrightarrow n_i}$ is the new and $RxPr^{\text{old}}_{n_j \longrightarrow n_i}$ is the old receiving power of the "HELLO" packet from node n_j to node n_i. In case when $RxPr^{\text{new}}_{n_j \longrightarrow n_i} < RxPr^{\text{old}}_{n_j \longrightarrow n_i}$, $M^{\text{Rel}}_{n_i}(n_j)$ is negative; it means n_j is moving away from n_i as shown in Figure 2. When $RxPr^{\text{new}}_{n_j \longrightarrow n_i} > RxPr^{\text{old}}_{n_j \longrightarrow n_i}$, $M^{\text{Rel}}_{n_i}(n_j)$ is positive, i.e., n_j is coming closer to n_i. For each neighboring node, n_j, of node n_i, its range indicator $R_{\text{Ind}}(n_j, n_i)$ with respect to n_i is measured. Based on the separation distance between the nodes and their relative mobility, the range indicator is classified as follows:

$$R_{\mathrm{Ind}}\left(n_j, n_i\right) = \begin{cases} 0, & \text{if } \alpha_1 r < \operatorname{dist}\left(n_i, n_j\right) \le \alpha_2 r \wedge M_{n_i}^{\mathrm{Rel}}\left(n_j\right) < 0, \\ 1, & \text{if } \operatorname{dist}\left(n_i, n_j\right) \le r \wedge M_{n_i}^{\mathrm{Rel}}\left(n_j\right) > 0, \\ 1 + \dfrac{\operatorname{dist}\left(n_i, n_j\right) - \alpha_1 r}{\left(\alpha_{2-}\alpha_1\right) r}, & \text{if } \operatorname{dist}\left(n_i, n_j\right) \le \alpha_1 r \wedge M_{n_i}^{\mathrm{Rel}}\left(n_j\right) < 0. \end{cases} \quad (3)$$

In the situation when a node lies in the risk zone and its mobility is negative, then it is moving distant away from the node, which is computing its weight to be chosen as a CH. Such a neighboring node is discarded as it will escape the transmission scope of the concerned node. In the event that a node is anywhere inside the transmission range and its relative mobility is additionally positive, it implies that the node is coming nearer to the node ascertaining its weight. Lastly, in the third circumstance, when a node is in the trusted zone, however, it is moving away, at that point, its commitment to discovering the range indicator relies upon how far it is moving away from the concerned node, as shown in Equation (3). Finally, the stability deviation of the node $n_i(\mathrm{STD}(n_i))$ is determined as the summation of the range indicator multiplied by the distance $(\operatorname{dist}(n_i, n_j))$, and the whole divided by its cardinality of nodes $(N(n_i))$ as given in the following equation:

$$\mathrm{STD}\left(n_i\right) = \frac{\sum_{j=1}^{k} R_{\mathrm{Ind}}\left(n_j, n_i\right) \times \operatorname{dist}\left(n_i, n_j\right)}{N}\left(n_i\right). \quad (4)$$

3.3. Fuzzy Q-Learning. Fuzzy Q-Learning is a type of reinforcement learning based on scalar rewards given by the environment. It is employed to reinforce a system's learning capability.

The steps in Fuzzy Q-Learning (FQL) are discussed as follows:

(1) *Initialize the q values*: initially, the q values of the Q-matrix are set with all zeros or any random value. Each entry of the value table was set in accordance with a specific rule and eventually gets updated during the learning process

(2) *Select an action*: based on the exploration of the system, actions are chosen next. The chosen action tends to give best reward

(3) *Calculate the control action from the logic controller*: the weighted average of the fuzzy rules is called as a fuzzy output which is calculated as

$$a = \sum_{i=1}^{M} \mu_i(x) \times a_i, \quad (5)$$

where M is the number of rules, $\mu_i(x)$ is the firing strength of rule i for input signal x, and a_i is the consequent function for the fired rule

(4) *Approximate the Q function*: based on the current q values combined with the firing strength of the rules, the Q function is calculated. A fuzzy inference system has the advantage that actions composed of many rules can be executed at once. Hence, the q value for the state-action pair $Q(s, a)$ is calculated as

$$Q(s, a) = \sum_{i=1}^{M} (\mu_i(s) \times q\,[i, a_i]. \quad (6)$$

The $Q(s, a)$ value tells the desirability of stopping at state s by either taking a unique action a or continuously employing the same action a to the current state

(5) Let the system go to the next state $s(t + 1)$ upon taking the action a

(6) *Calculate reward value*: the controller upon receiving the current values of input parameters for the current state of the system s calculates the reward for going from one state to other

(7) *Calculate the value of the new state s'*: the value of the new state s' on being reached from the state s with the deployment of the action a is calculated as

$$V\left(s'\right) = \sum_{i=1}^{M} (\mu_i\left(s'\right) \times \underset{k}{\max} \; (q\,[i, a_k]), \quad (7)$$

where max $(q[i, a_k])$ is the maximum of the q values achievable in state s'

(8) *Calculate the error signal*: if the maximum reward given to the system deviates from the predicted one, then an error signal is calculated as follows:

$$\Delta Q(s, a) = r + _{\gamma} V\left(s'\right) - Q(s, a), \tag{8}$$

where γ is the discount rate determining the future reward

(9) *Update q values at each step*: the q value is updated by the following equation:

$$q[i, a_i] = q[i, a_i] + \eta \cdot \Delta Q \cdot \mu_i(s(t)), \tag{9}$$

where η is the learning rate

4. Proposed Algorithm

The objective of the proposed algorithm is to prolong the network lifetime by considering the mobility of nodes as well as their direction of motion. For this, we have proposed a clustering algorithm based on Fuzzy Q-Learning and Chebyshev's inequality principle. The proposed FQ-MEC consists of two phases of clustering, i.e., the setup phase, and the monitoring phase as depicted in Figure 3. The setup phase is the initial stage of clustering, where the nodes once deployed choose their role of clustering based on their weighted sum of stability deviation (STD) and energy depletion (ED). After the initial clusters have been formed, the clusters are monitored periodically for load balancing. Thus, the second phase of FQ-MEC is the monitoring phase, which consists of load balancing and reclustering. Chebyshev's inequality principle analyzes the history of degree of connectivity of each CH so as to adaptively maintain their load. In load balancing, the reaffiliation of nodes to a CH is based on the current load on CH which adaptively makes the node join a CH with minimum weight. The reclustering of nodes is done based on the predefined fuzzy rules that are described in the following subsections.

4.1. The Setup Phase. The setup phase is the initial phase of clustering. Each node calculates its combined weight using STD and ED as given in the following equation:

$$W(n_i) = W_1 \times \mathrm{STD}(n_i) + W_2 \times \mathrm{ED}(n_i). \tag{10}$$

The nodes with minimum weight declare themselves as CHs and broadcast their role to their neighbors [16].

The neighboring nodes get attached with CHs, which are within their transmission range. The setup phase algorithm is shown in Algorithm 1.

4.2. The Monitoring Phase. Monitoring is the important phase that is given, focused upon in the proposed FQ-MEC. Once clusters are formed, the periodic monitoring of nodes is done. The aim of this phase is to prevent the early dissolving of clusters by reducing the number of reclustering thus improving the network lifetime. It consists of load balancing and reclustering. The loads on CHs are balanced periodically, so that no CH gets overloaded in transmitting packets. Reclustering is done if the number of isolated clusters increased to a certain threshold, which depends upon the total number of nodes in the network. Load balancing and reclustering are described as follows.

4.2.1. Load Balancing. The loads on CHs are balanced using Chebyshev's inequality principle, which determines the degree of connectivity for each node. In due course, after the formation of clusters, CH witnesses varying degrees of connectivity, i.e., the number of nodes in the range and adjacent to the CH. At the very start, when clusters are formed in the setup phase, nodes start communicating with their respective CHs. Each CH maintains its cluster information table, indicating its cluster members and gateways in its range. It is the responsibility of CH to find the route and transmit packets of its cluster members via gateway nodes, if any. Thus, CHs dissipate their energy more quickly. The nodes are mobile, and this mobility of nodes affects the degree of connectivity of the CHs. To prevent CHs from overloading and their early death, the load is balanced on CHs. To keep a check on their degree of connectivity, we have incorporated Chebyshev's inequality principle and formulated a rule which fits well for the case of the load balancing of CHs.

(1) Chebyshev's Inequality and Getting the Degree of Connectivity of CHs. Chebyshev's inequality guarantees that no more than $1/k^2$ of the distribution's values can be more than k standard deviations away from the mean (or equivalently, at least $1 - 1/k^2$ of the distribution's values is within k standard deviations of the mean).

In other words, this means that the chances that the "mean" of any variable, say "A" upon getting subtracted from the variable, is always lower than k times the "standards deviation" of the variable, which in turn is greater than or equal to $(1 - 1/k^2)$ [17]. Mathematically,

$$P(|A - E(A)| < k\sigma_A) \geq \left(1 - \frac{1}{k^2}\right). \tag{11}$$

where $E(A)$ is the expectation or the mean of the sample, σ_A is the standard deviation, and k is a constant generally taken to be either 3 or 6.

This implies that

$$A - E(A) \geq 0, \tag{12}$$

$$|A - E(A)| = A - E(A), \tag{13}$$

$$A - E(A) = k\sigma_A, \tag{14}$$

$$A < E(A) + k\sigma_A. \tag{15}$$

Similarly,

$$A - E(A) < 0, \tag{16}$$

$$|A - E(A)| = -(A - E(A)) = E(A) - A, \tag{17}$$

$$E(A) - A = k\sigma_A, \tag{18}$$

$$E(A) - k\sigma_A < A. \tag{19}$$

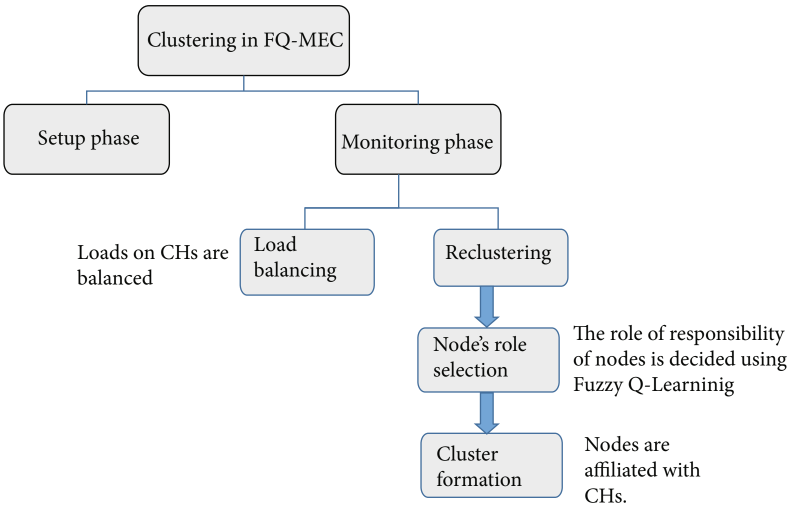
Clustering in FQ-MEC
Setup phase
Monitoring phase
Loads on CHs are balanced
Load balancing
Reclustering
Node's role selection
The role of responsibility of nodes is decided using Fuzzy Q-Learninig
Cluster formation
Nodes are affiliated with CHs.

Figure 3: Phases of FQ-MEC.

```
Input: a set of nodes, $N = \{n_1, n_2 \cdots, n_l\}$, weighing factors $w_1$ and $w_2$.
Output: a set of elected CHs, $\Psi = \{ch_1, ch_2 \cdots, ch_m\}$.
Begin:
Step 1: forI = 1 to $l \forall n_i \epsilon N$ do
  1.1: Each node $n_i$ broadcast and receive "HELLO" message to and from all its one hop neighbors.
  1.2: Estimate the total number of one-hop neighbors.
  1.3: Find STD($n_i$) and ED($n_i$).
  1.4: Calculate weight, $W(n_i)$ using Equation (10) and broadcast it to all its one-hop neighbors.
Step 2: set flag = 1
Step 3: while (flag = = 1 $\bigwedge$ $n_i$ is receiving $W(n_j)$ $\forall n_j \epsilon T_{\text{range}}(n_i)$) do
  3.1: if$W(n_j) < W(n_i)$ then
  3.2: $n_i$ gives up the competition for CH election
  3.3: set flag = 0
    end if
  end while
Step 4: if (flag= = 1) then
  4.1: $n_i$ declares itself as CH and broadcasts CH advertisement message with its ID and weight, $W(n_i)$, to $n_j \forall n_j \epsilon T_{\text{range}}(n_i)$).
  4.2: else$n_i$ is an isolated node, so it declares and advertises itself as a CH after timeout.
      end if
Stop
```

Algorithm 1: Setup phase of FQ-MEC.

From equations (12) and (16) above, we conclude that

$$P(E(A) - k\sigma_A < A < E(A) + k\sigma_A) \geq \left(1 - \frac{1}{k^2}\right). \quad (20)$$

Taking $k = 3$ (as it is giving 95% correction than taking $k = 6$), we get

$$P(E(A) - 3\sigma_A < A < E(A) + 3\sigma_Y). \quad (21)$$

This equation when applied to get the degree of connectivity of CHs gives

Table 1

Cluster heads	Degree of connectivity	Degree of connectivity (estimated)
CH1	(3.252453-6.08088)	(3-6)
CH2	(2.91912-5.747547)	(3-6)
CH3	(0.925009-8.408324)	(1-8)
CH4	(2.55051-7.44949)	(3-7)
CH5	(1.83824-7.495094)	(2-7)
CH6	(1.504906-7.16176)	(2-7)

```
Input: degree of connectivity of CHs for different time intervals
Output: range of degree of connectivity of CHs in the cluster
Begin:
Step 1: for$I = 1$ to $ch_i$do
  1.1: Calculate the total number of connected nodes for equal intervals
  1.2: Calculate the mean of connectivity as $\mu_\varphi$ (where $\varphi$ represents the degree of connectivity)
  1.3: Calculate the standard deviations of the connectivity as $\sigma_\varphi$
Step 2: take $k = 3$
Step 3: for each $ch_i$do
  3.1: $\mu_\varphi - k * \sigma_\varphi$
  3.2: $\mu_\varphi + k * \sigma_\varphi$
Step 4: calculate the range of degree of connectivity $\varphi$ as
  4.1: $\mu_\varphi - k * \sigma_\varphi < \varphi < \mu_\varphi + k * \sigma_\varphi$
Stop
```

Algorithm 2: Load balancing algorithm of FQ-MEC.

$$\begin{aligned} P\big(\text{Mean}(\text{Node Connectivity}) - 3\big(\text{Standard Deviation}_{\text{Node Connectivity}}\big)\big) \\ < \text{Node Connectivity} < (\text{Mean}\ (\text{Node Connectivity}) \\ + 3\big(\text{Standard Deviation}_{\text{Node Connectivity}}\big)\big). \end{aligned} \tag{22}$$

We collected a sample of CH's connectivity at the interval of 10 seconds to see their variation of connectivity and applied the above formula to find the range of degree of connectivity of the CHs of our created hypothetical cluster. The results are tabulated in Table 1.

The algorithm for load balancing is given in Algorithm 2.

4.2.2. Reclustering. Once clusters are formed in the setup phase, reclustering is required when the maximum number of clusters suitable for a particular network size becomes greater than some threshold value which depends upon the total number of nodes in the network as follows:

$$N_{\text{Max}} = \frac{N}{\delta}, \tag{23}$$

where N_{Max} is the maximum number of clusters suitable for a particular network size, N is the total number of nodes in the network, and δ is the ideal degree for a particular network such that $\delta \le 2 \ln (N)$.

(1) Fuzzy Q-Learning for Reclustering. For reclustering, we have used the Fuzzy Q-Learning approach, which selects an appropriate action based on the predefined fuzzy rules. The selection of action state depends on the optimized value of the Q table. As discussed above, the Fuzzy Q-Learning approach learns from the reward given to an entity in going from one state to other. For the case of reclustering of nodes in MANET, we have taken three actions in the form of role selection for each node. The nodes can get the responsibility of acting as either a cluster head (CH), cluster member (CM), or gateway (G). Since we have formulated nine fuzzy rules, our initial Q table contains nine rows and nine columns, where each column corresponds to a selected node responsibility. The first three columns correspond to the selection of CHs, the next three columns specify gateways, and the last three columns correspond to the selection of CMs.

Initially, the entries in the Q table are set to 0 and get updated and optimized as the algorithm progresses. Next, the rules are fired and an action "a" is taken based on the outcome of the fired rule. So if a fired rule states that on both stability deviation and energy depletion being low, the node responsibility is to be assigned as the cluster head, then any three out of the first three columns of the Q-matrix must get optimized with the maximum value. The value we get depending on the firing strength of the rule gets entered into the corresponding column of the Q-matrix as approximated by the Q function. The selection of controlled action $Q(S, a)$ depends on the value of the reward given for selecting the action. The reward value is given in the reward matrix, which in our case depends on the weighted average of two controlled factors, viz., stability deviation and energy depletion.

According to the rules, the values are set to be minimum in the corresponding columns. The calculation of the reward value has been done by using the following equation:

$$R_i = \min \left\{ \frac{\text{STD}_1 + \text{ED}_i}{2} \forall 1 \le i \le N_i \right\}. \tag{24}$$

```
Input: a set of nodes, $N = \{n_1, n_2, \cdots n_l\}$, reward matrix $R$, fuzzy rules $\text{Rule}_i$, empty Q-Matrix $Q\ (i, a)$,
        learning rate $\eta = 0.1$ and discount factor $\gamma = 0.9$
Output: selected responsibility of each node as cluster head, cluster member, gateway
Begin:
Step 1: forI = 1 to Rule_i do
  1.1: Calculate the action $a_i = \max\ Q(i, a)$ as
      $a(s) = \sum_{i=1}^{N} \alpha_i(\text{Rule}) * \underset{k}{\max} (q[i, a_k])$
      where $\sum_{i=1}^{N} \alpha_i(\text{Rule})$ is the firing strength of the rule and max $(q[i, a_k])$
      is the maximum of the $q$ values which can be achieved in state $s'$.
  1.2: Update the current state $s$ of the Q-matrix corresponding to the selected action and approximated by the Q function as
      $Q(s(i), a) = \sum_{i=1}^{N} (\alpha_i(s) * q\,[i, a_i])$
Step 2: take a controlled action $a$ and based on the selected action go to the next state $s'$,
        $s\,(i + 1)$
Step 3: calculate the minimum reward value $R_i$ based on the weighted average of the two parameters, energy depletion and stability deviation, as
    $R_i = \min \{(\text{STD}_i + \text{ED}_i)/2\forall_1 \leq i \leq N_i\}$
Step 4: upon taking action $a$ and leaving from state $s$ to $s'$, observe the reward value for the next state $r(i + 1)$ and calculate the action of the new state $s'$ as
    $a(s') = \sum_{i=1}^{N} \alpha_i(\text{Rule}) * \underset{k}{\max} (q[i, a_k])$
    where max $(q[i, a_k])$ is the maximum of the $q$ values which can be achieved in state $s'$.
Step 5: if the reward function deviates from the original, calculate the error signal as
    $\Delta Q(s, a) = r\,(i + 1) + \gamma * a\,(s') - Q(s\,(i), a)$
Step 6: update the $q$ values as
    $Q(i, a) = Q(s(i), a) + \eta \cdot \Delta Q \cdot \alpha_i(\text{Rule})$
Step 7: nodes broadcast their role of responsibilities once decided.
Step 8: neighboring nodes join CH within their transmission range with minimum combined weight.
Stop
```

ALGORITHM 3: Reclustering algorithm of FQ-MEC.

The next state to be optimized, S' is then selected based on the gathered reward value. Finally, the values in the Q table are gradually updated based on the reward value and error signal. Once the values in the Q table get optimized, we can select an action corresponding to the optimized column in the fired rule.

Implementing the Fuzzy Q-Learning approach for our proposed reclustering algorithm is detailed as follows.

(2) Role Selection Algorithm. Once CHs are decided in the setup phase of clustering by calculating and broadcasting the combined weight of each node and deciding CHs as nodes which are less deviated from stability and have depleted less energy, in the reclustering of the maintenance phase, nodes learn by themselves to decide their role of responsibility using Fuzzy Q-Learning. The two parameters used in clustering are STD and ED. The rules formed by expert knowledge using these two parameters are framed as follows:

(1) If STD is low and ED is low, then select the role as CH

(2) If STD is low and ED is medium, then select the role as gateway

(3) If STD is low and ED is high, then select the role as CM

(4) If STD is medium and ED is low, then select the role as CH

(5) If STD is medium and ED is medium, then select the role as gateway

(6) If STD is medium and ED is high, then select the role as CM

(7) If STD is high and ED is low, then select the role as CH

TABLE 2: Simulation parameters.

Parameter	Value
Number of nodes	10-50
Simulation area	750 m × 750 m
Simulation time	50 sec
Initial energy	45000 nJ
Packet size	512 bytes
Transmission range of nodes	10 m-180 m
Routing protocol	AODV
Movement model	Random-way point
Radio propagation model	Two ray ground

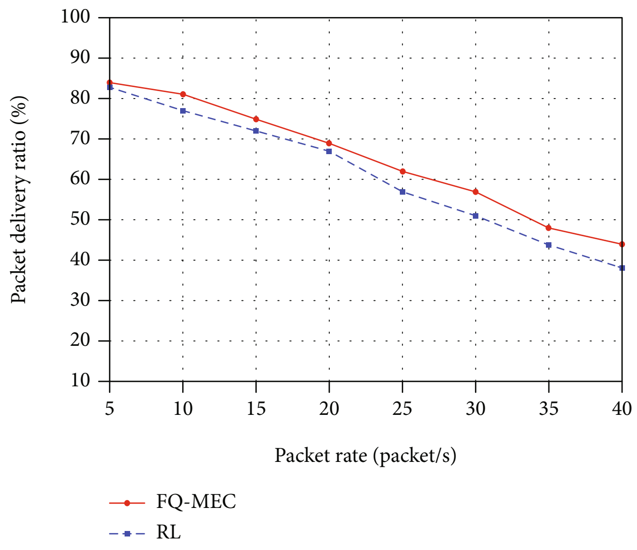

Figure 4: Packet delivery ratio versus packet rate.

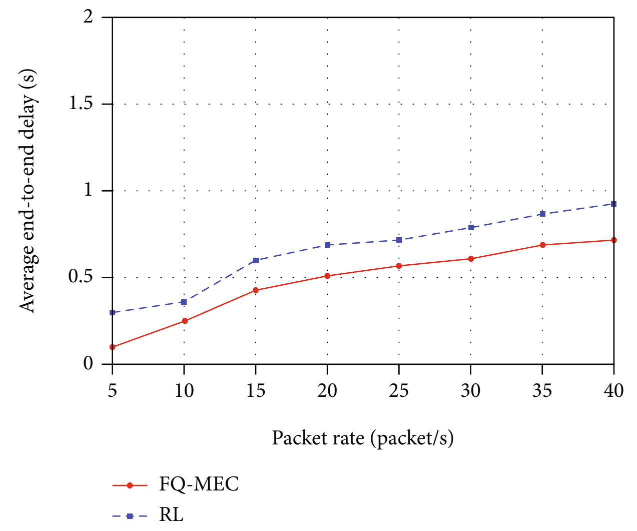

Figure 5: Average end-to-end delay versus packet rate.

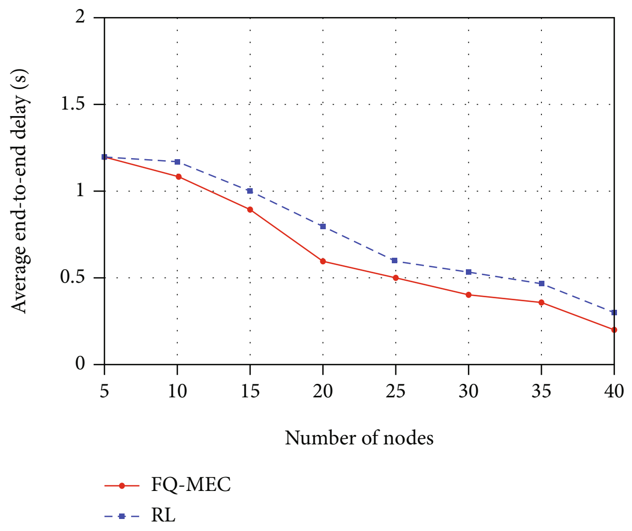

Figure 6: Average end-to-end delay versus no. of nodes.

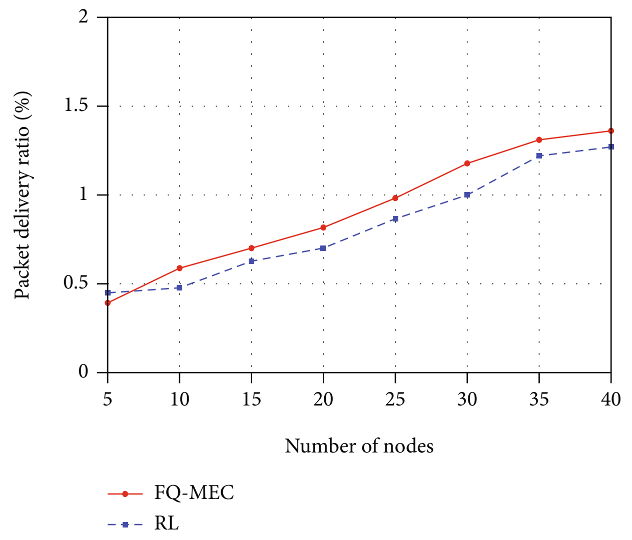

Figure 7: Packet delivery ratio versus no. of nodes.

(8) If STD is high and ED is medium, then select the role as CH

(9) If STD is high and ED is high, then select the role as CM

(3) Cluster Formation. Once nodes decide their role of responsibility, they broadcast their role. A cluster member joins a CH within its transmission range. If a node receives a CH join message from more than one CH, it will join a CH with the minimum combined weight (Algorithm 3).

5. Results and Discussion

In this section, we describe the simulation of the proposed work. We performed a pervasive simulation experiment. We have used the network simulator NS-2 to perform the simulation with the experimental setup as given in Table 2.

We first implemented the already existing real-time routing algorithm for MANET using reinforcement learning (RL) and heuristic algorithms [6] and compared the result with our proposed work of clustering in MANET. The simulation result shows the efficacy of the fuzzy-based Q-learning approach for mobility-aware energy-efficient clustering in MANET (FQ-MEC) over RL.

Figure 4 presents the packet delivery ratio (PDR) with increase in packet rate. It shows the number of packets successfully received by the destination node. The routing of packets is done via CHs and gateway nodes, and these nodes are prudently selected with most eligible nodes to handle intracluster and intercluster routing. The constraints of MANET, i.e., node's mobility and battery dependency, are handled with the parameters of stability deviation and energy

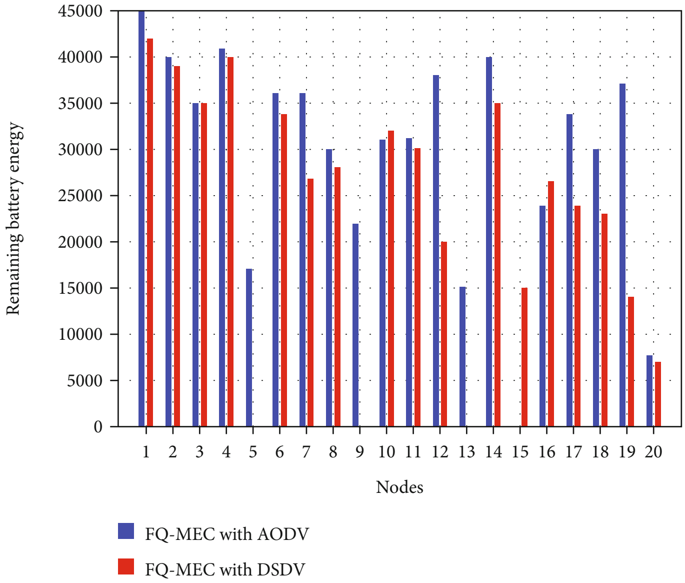

Figure 8: Remaining battery energy per node.

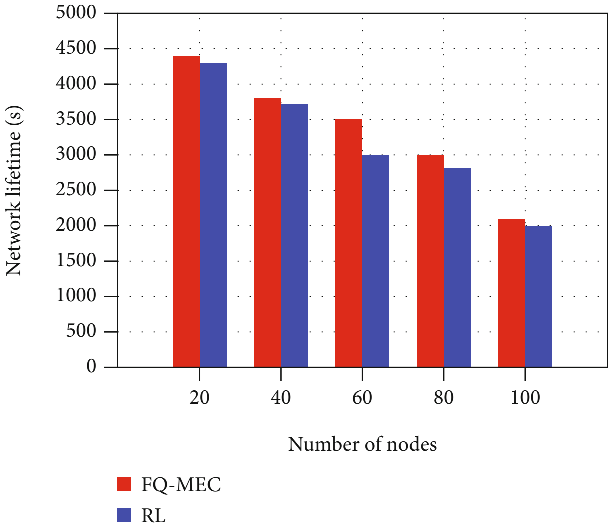

Figure 9: Network lifetime versus no. of nodes.

depletion in clustering. A node becomes inactive when its energy level is depleted to zero. The probability of nodes to become inactive has been reduced by load balancing using Chebyshev's inequality principle. Therefore, the route disruption due to movement or death of nodes is minimized to a great level. This also reduced average end-to-end delay of FQ-MEC as illustrated in Figure 5.

In Figure 6, average end-to-end delay is presented with an increase in the number of nodes. As shown in the graph, by increasing the number of nodes, the end-to-end delay has been decreased. As the number of gateway nodes responsible for intercluster transmission is increased, the delay is reduced and packet delivery ratio is improved which is depicted in Figure 7. In both cases, it can be seen that the FQ-MEC gives a better result when compared to RL.

In Figure 8, we show the comparison of the number of inactive nodes per round when the proposed clustering algorithm is applied with AODV and DSDV routing protocols. A node becomes inactive when its energy level reaches to zero or it is no longer within the transmission range of any CHs.

As AODV is an on-demand driven routing protocol, there is no need to maintain or continuously update the routing path setup between the source and destination nodes like DSDV, which is a table-driven routing protocol. Due to less overhead of maintaining the routing table all the time, it gives a better result of coupling the FQ-MEC with AODV than the same coupled with DSDV. As shown in the graph, the number of nodes that died due to energy dissipation is three when DSDV is used, and it is one when AODV is used.

Next, we ran the algorithm to compare the network lifetime by varying the number of nodes from 20 to 100. A CH may die quickly because of improper load balancing. When loads on a CH increase due to increase in the number of nodes within its transmission range, the CH got overloaded with data forwarding. As a result, due to the death of the CH, some nodes got isolated as they were unable to find any CH within their transmission range. The periodic load balancing of nodes in the monitoring phase of the proposed algorithm to create near-to-homogeneous clusters using Chebyshev's inequality principle shows the effectiveness in the overall network lifetime, as shown in Figure 9.

6. Conclusion and Future Work

Designing energy-efficient and mobility-aware clustering in MAET has been considered in this paper. Mobility and battery dependency are the challenging constraints in increasing the network lifetime of such networks. To manage these constraints, a fuzzy-based Q-learning approach for mobility-aware energy-efficient clustering in MANET (FQ-MEC) has been proposed. The setup phase of clustering takes the node's stability deviation and energy depletion factors to decide CHs, which can retain their responsibility for a longer duration of time. Here, more focus has been given on the monitoring phase of clustering, where the fuzzy-based Q-learning approach of reinforcement learning has been implemented, aimed at deciding the behavioral patterns of nodes with Chebyshev's inequality principle to adaptively maintain the loads on CHs. With this, the speed of convergence and learning is achieved, which reduces the rate of reclustering with an improved network lifetime.

As a part of future research work, SARSA learning methodology will be applied to obtain the optimal solution for both the routing and clustering in MANET.

Data Availability

Data are available on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. Popli, V. Juneja, K. Garg, and D. V. Gupta, “Fuzzy based trust evaluation models for enhancing security in MANETs,” *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 5, pp. 506–510, 2019.

[2] H. Jiang, Y. Sun, R. Sun, and H. Xu, “Fuzzy-logic-based energy optimized routing for wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 9, no. 8, 2013.

[3] S. K. Gupta, P. Kuila, and P. K. Jana, “Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks,” *Computers and Electrical Engineering*, vol. 56, pp. 544–556, 2016.

[4] S. Amri, F. Khelifi, A. Bradai, A. Rachedi, M. L. Kaddachi, and M. Atri, “A new fuzzy logic based node localization mechanism for wireless sensor networks,” *Future Generation Computer Systems*, vol. 93, pp. 799–813, 2019.

[5] T. Kumanan and K. Duraiswamy, *Reinforcement learning for quality of service in mobile ad hoc network (MANET)*, 2010.

[6] H. Y. Chang, T. Ho, and P. L. Kaelbling, *Mobilized ad-hoc networks: a reinforcement learning approach*, 2004.

[7] S. Chettibi and S. Chikhi, “Routing in mobile ad-hoc networks as a reinforcement learning task,” in *Networked Digital Technologies. NDT 2011. Communications in Computer and Information Science*, S. Fong, Ed., vol. 136, Springer, Berlin, Heidelberg, 2011.

[8] F. Dong, O. Li, and M. Tong, “Intelligent routing control for MANET based on reinforcement learning,” *MATEC Web of Conferences*, vol. 232, article 04002, 2018.

[9] J. Dowling, E. Curran, R. Cunningham, and V. Cahill, “Using feedback in collaborative reinforcement learning to adaptively optimize MANET routing,” *IEEE Transactions on Systems Man and Cybernetics-Part A Systems and Humans*, vol. 35, no. 3, pp. 360–372, 2005.

[10] A. Ghaffari, “Real-time routing algorithm for mobile ad hoc networks using reinforcement learning and heuristic algorithms,” *Wireless Networks*, vol. 23, no. 3, pp. 703–714, 2017.

[11] M. Maleki, V. Hakami, and M. Dehghan, “A model-based reinforcement learning algorithm for routing in energy harvesting mobile ad-hoc networks,” *Wireless Personal Communications*, vol. 95, no. 3, pp. 3119–3139, 2017.

[12] G. Jinarajadasa, L. Rupasinghe, and I. Murray, “A reinforcement learning approach to enhance the trust-level of MANETs,” in *2018 National Information Technology Conference (NITC)*, pp. 3119–3138, Colombo, Sri Lanka, October 2018.

[13] K. Fall, K. Varadhan, T V Project et al., *The ns Manual (Formerly ns Notes and Documentation) 1, (3)*, 2011, http://www.isi.edu/nsnam/ns/.

[14] M. Aissa and A. Belghith, “A node quality based clustering algorithm in wireless mobile ad hoc networks,” *Procedia Computer Science*, vol. 32, pp. 174–181, 2014.

[15] N. Khatoon and Amritanjali, “Mobility aware energy efficient clustering for MANET: a bio-inspired approach with particle swarm optimization,” *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 1903190, 12 pages, 2017.

[16] N. Khatoon and Amritanjali, “Retaliation-based secured model for enhanced weighted clustering algorithm in mobile ad-hoc network,” *International Journal of Wireless and Mobile Computing*, vol. 14, no. 2, pp. 164–177, 2018.

[17] P. Pranav, S. Chakraborty, and S. Dutta, “A new cipher system using semi-natural composition in Indian raga,” *Soft Computing*, vol. 24, no. 2, pp. 1529–1537, 2020.