1828
AKADÉMIAI KIADÓ

Energy-efficient threshold-based reinforcement learning for WSN routing

Archana Chaudhari[1], Masuk Abdullah[2*], Vivek Deshpande[3] and Divya Midhunchakkaravarthy[4]

[1] Department of Instrumentation Engineering, Vishwakarma Institute of Technology, Savitirbai Phule Pune University, Pune, India

[2] Department of Vehicles Engineering, Faculty of Engineering, University of Debrecen, Debrecen, Hungary

[3] Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Information Technology, Savitirbai Phule Pune University, Kondhwa, India

[4] Centre of Postgraduate Studies, Lincoln University College, Petaling Jaya, Malaysia

Pollack Periodica • An International Journal for Engineering and Information Sciences

DOI: 10.1556/606.2025.01336

Received: January 30, 2025 • Revised manuscript received: May 19, 2025 • Accepted: May 21, 2025

ORIGINAL RESEARCH PAPER

ABSTRACT

The proposed method uses a threshold-based reinforcement learning algorithm, Q-learning, for efficient routing in wireless sensor networks. It penalizes node energy, hop count, and distance to sink using three key thresholds to select the next best forwarder. This prevents self-loops and ensures packet delivery by avoiding low energy or high hop count nodes. The combination of reinforcement learning and key threshold values allows intelligent data packet routing, handling congestion near sink nodes, and reliable transmission. Experimental results show a 35% enhanced network lifetime compared to the state-of-the-art algorithm.

KEYWORDS

wireless sensor networks, threshold-based Q-routing, hop count, node energy, sink

1. INTRODUCTION

Wireless Sensor Networks (WSN) are used in many applications in various fields, like agriculture, military, healthcare, etc. A WSN consists of various types of sensors deployed in a dynamic environment for monitoring different parameters. The sensors used for various applications include temperature, pressure, motion, optical, weather and many others.

Sensor networks consist of many sensors deployed in different areas for monitoring parameters. These sensor networks consist of source and sink nodes. Source nodes monitor parameters and send information to the sink nodes [1]. Figure 1 represents a network of WSN.

The sensor network environment is very dynamic. Additionally, the nodes have limited energy and mobility. Sensor networks gather and extract useful information from environment for transmission to sink. The routing of similar nodes for data collection is an important part of wireless sensor networks. While routing, sensor nodes encounter changing network conditions due constrained environment. Thus, the aim of the sensor networks is to maintain the energy of the node and to ensure that all nodes are alive to maintain the network lifetime. For enhanced performance, it is advantageous to learn the changing environment for better routing performance [2, 3].

Reinforcement Learning (RL) can help in learning the environment, as it is a stream of machine learning that can be used when datasets are not available. In reinforcement learning, the system or agent learns from interactions with the environment, and actions can be

*Corresponding author.
E-mail: masuk@eng.unideb.hu

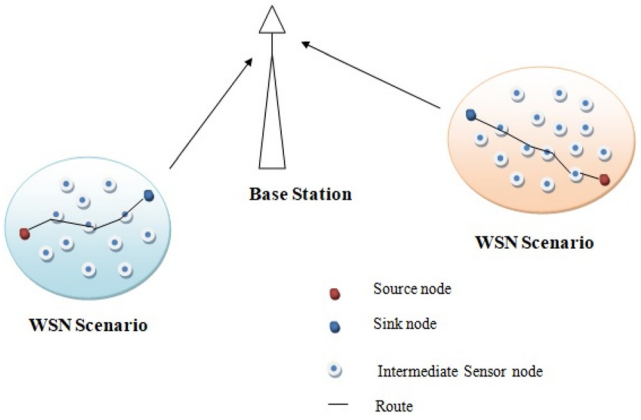

Fig. 1. WSN with a base station and source and sink nodes

selected [4–7]. The actions taken by the agent depend on initial state and agent policy [8]. Upon implementation of action, the agent receives feedback for the action taken, termed as reward. Reward therefore plays a vital role to achieve long-term or short-term goals. Several reinforcement learning techniques based on various rewards are presented in the literature for routing in WSN [9].

2. RELATED WORK ON RL FOR ROUTING IN WSNS

Two types of approach are proposed based on the learning of environment by the agent in RL: model-based approach and a model-free approach. Model-based approach depends on initial conditions of the environment and follows the greedy approach for learning. In the greedy approach, the agent takes action that receives the maximum reward irrespective of the effect of the action. This type of approach is more suitable for static environments.

In the model-free approach, the agent does not depend on the initial conditions of the environment and learns through interactions and experience with the environment. In a model-based approach, the agents learn through algorithms similar to the policy gradient or Q learning. The agent takes action on the basis of experiences from past feedback and current actions. This type of approach is therefore suitable for changing environments in wireless sensor networks in particular routing. The goal of most reinforcement-based routing algorithms is to select the next best forwarder neighbor node for packet delivery at the destination node.

This section discusses a brief review of literatures on RL-based routing algorithms. Q routing was the first RL-based algorithm proposed by Boyan and Littman [10] based on Q-learning. Action value function is exploited in Q-routing optimal routes are obtained by exploiting the number of hops a packet takes along the routes. In the work, the node itself is the current state. The next forwarder neighbor node is selected based on minimum Q value of neighbor nodes. Link cost between two nodes and delivery delay between two nodes is explored as the reward to obtain Q value. The lower the Q value, the lower the delivery delay and the better the link between the nodes. Since Q routing is a model-free approach, it does not depend on the initial network and traffic conditions to obtain the optimal route [11]. The Q routing algorithm is sensitive to initial parameter settings and requires many trials.

The limitation of the Q-routing algorithm was overcome in adaptive routing, which was proposed by the authors of [12]. In AdaR the authors proposed a method based on Least Squares Policy Iteration (LSPI), which explores the best routing strategy with a small number of trials. In the proposed work, the node is considered as the state, and an action is performed on the basis of the node state. The hop count among two neighboring nodes to the base node, the remaining energy, and number of routes crossing neighboring nodes are the features considered when performing the action.

In RL-based Quality-of-service-alert Routing Protocol (RL-QRP) [13], the authors proposed RL based on Quality-of-Service (QoS)-aware routes. QoS routes are explored individually and independently of each other, which is known as distributed Q learning. Packet delivery and the end-to-end delay form QoS requirements. Node with high Q-value is selected as next forwarder node based on the QoS. A combination of the reward function and expected future reward is used for Q-value updation. Future decisions depend on Q value of the corresponding node. Thus, distributed Q learning helps exploit optimal routes through the environment and rewards in a changing environment.

An extension of the RL-QRP is proposed on the basis of multi-agent reinforcement learning with QoS support [14]. The multiple agents in the Multiple Reinforcement Learning-based (MRL-QRP) exploit the local and network information of neighbors to obtain optimal performance. When the node takes an action of forwarding a data packet to its neighbor, a positive reward is received by the node acting as the agent. Delay and packet loss rates contribute to the reward. The Q values of the agent are updated on the basis of the reward and expected long-term reward. At convergence, data is forwarded to node having highest Q value. In MRL-QRP weighted sum of own and neighbor node expected rewards are explored. Action in chosen in a cooperative manner for maximizing rewards.

An energy-aware routing protocol based on QoS was proposed by the authors in [15]. In the proposed work, while the data are forwarded to the neighbor node, each neighbor in the route provides feedback on the time needed to send the packet forward. The node that forwards the data computes round-trip and node receiving the data selects next hop of data packet, updates the receiver information along with its own end-to-end delay in the packet header. It also adds its own remaining energy and the residual energy in the route. The feedback of the time is then encapsulated in the header of the data packet. This helps to minimize the overhead cost and saves energy to the node. This is used to update Q table as well as the expected minimum future cost. As the algorithm updates the route cost, this helps the

algorithm successfully handle mobility and failure recovery. The algorithm therefore learns the environment through RL and selects the next node on the basis of current state of the network instead of selecting next forwarder with the best QoS.

The Reinforcement Learning-Based Routing (RLBR) algorithm, which focuses on energy efficiency and network lifetime optimization, was proposed by Guo et al. [16]. The nodes that act as states perform the action of forwarding packets to the next best neighbor node by exploiting the information of path quality. The proposed method initially sends control packets within the network and collects the node IDentifiers (ID), location coordinates, remaining energy, and hop count as information of previous node. *Q*-value is thus computed from previous sender node and the defined reward. Reward is defined based on residual energy and hop count. *Q*-value represents path quality among nodes. Node with maximum *Q*-value is chosen as optimal next forwarder.

Akbari et al. [17] explored a highly reliable route for Internet of Things's via RL and fuzzy logic [17]. Optimal route is chosen using fuzzy logic along with RL. The node residual energy, available bandwidth, and distance to sink constitute the reward for enhancing network lifetime. A set of fuzzy rules is used to rate the rewards to obtain the optimal path to the sink node. The fuzzy system thus rates the quality of the path for better performance [18, 19].

Abadi et al. [20] propose an energy management reinforcement learning-based algorithm. The algorithm optimizes the routing policy for long-term reward of each node to improve the network lifetime. The authors propose three energy management approaches. The first approach focuses on reducing the route length to reduce energy consumption. In second approach, author explores the sleep scheduling of the node. The third focuses on data transmission.

In the proposed work [21], multiple greedy action changes are used to form a tree structure. During learning process, tree structure is adjusted dynamically. At root of the tree, every sensor node obtains next optimal hop, and at the leaf of the tree, every sensor node achieves updates from the learning errors.

The RL-based routing approach in [22] takes into account the current state of the network for finding optimal best forwarder node. Optimal routes with minimum transmission delays and better reliability on the basis of the reward function for the computation of the *Q* values are explored.

To reduce energy consumption and increase lifetime, the authors in [23] exploits grid-tree-based clustering. Root Node (RN) is selected using Reinforcement Learning-based Fuzzy Interface System (RLFIS) approach. Bat algorithm (BA) is used to locate the sink node.

This work proposes a new reinforcement-based routing algorithm. In the proposed work, neighbor node energy, hop count, and distance of neighbor nodes with respect to sink along with the rewards define *Q* values of each node. The rewards are significant computing *Q* values. The rewards chosen are based on a certain pre-defined threshold. Optimal forwarder is node with highest *Q* value.

3. PROPOSED THRESHOLD-BASED RL APPROACH FOR ROUTING

The proposed threshold-based RL algorithm is based on Q learning. Q learning algorithm consists of state (s) and action (a) pairs. Each state has a corresponding action. The action represents the decision a sensor node makes, which involves selecting the next-hop neighbor for data transmission, and it can take values on the basis of the available neighbor nodes. A state signifies the current state of a sensor node in sensor network. It comprises of the neighbor node energy (E), neighbor node hop count (H) to reach sink and neighbor node distance to sink (D).

The reward plays an important role in Q learning. It represents the feedback signal a sensor node receives on the basis of its actions. Figure 2 illustrates Q-learning algorithm.

In the proposed method, a threshold-based reward is exploited for better performance of the RL algorithm. The threshold-based reward is based on the energy of neighbor node, hop count of the neighbor node and its distance to sink. The threshold-based reward is defined as Eq. (1):

$$Reward = \begin{cases} \dfrac{1}{D^2}, & if \begin{cases} (neighbornode = 1), \\ (H \geq hopCountThreshold), \\ (E \leq EnergyThreshold), \end{cases} \\ \dfrac{100}{D^2}, & otherwise. \end{cases} \quad (1)$$

If the node has only one neighbor and hop count (H) of the neighbor is greater than *hopCountThreshold* and neighbor node energy (E) is less then *EnergyThreshold*, then reward is penalized. The reward obtained by the neighbor node is less than that obtained by other neighboring nodes. The threshold-based reward ensures better link quality in-between forwarder and sink and reduces chances of packet dropping. This is because only nodes with node energies

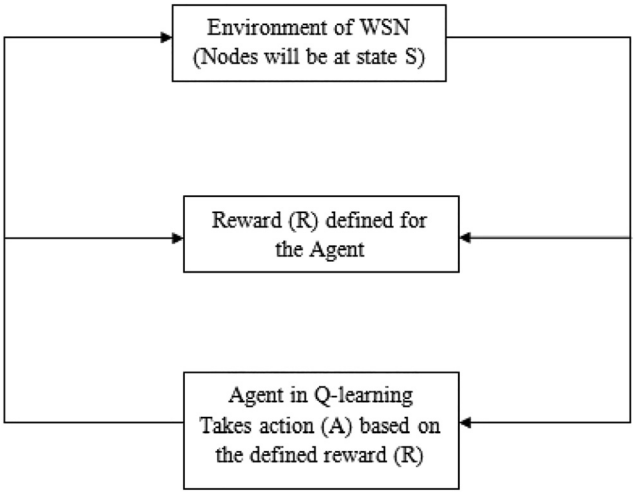

Fig. 2. Illustration of Q-learning algorithm

higher than the selected threshold and lower hop counts are selected.

The Q value of the proposed RL algorithm depends on the threshold-based reward proposed in Eq. (1). The Q value of every neighbor is computed via Eq. (2):

$$Q_{new} = \begin{cases} (1-\alpha)\cdot Q(s,a) \\ +\propto \cdot [Reward + \max(Q(\acute{s}, n_{actions}))], \end{cases} \quad (2)$$

where, $Q(s,a)$ is the Q value for state s and action a, α is the learning rate, a parameter between 0 and 1, *Reward* is defined as immediate reward received after taking action a in state s, γ is discount factor, which determines the status of future rewards ($0 \leq \gamma \leq 1$), and $\max(Q(\acute{s}, n_{actions}))$ is maximum Q value among possible actions $n_{actions}$ in next state s. Best forwarder is node with the highest Q value.

Steps of proposed threshold-based RL routing algorithm are as follows:

1. Initialize the network via initial parameters similar to number of nodes (N), sink node, source node, transmission range (R), area of the network and location or position (x, y) of each node;
2. The distance between two nodes $N_1(x_1, y_1)$ and $N_2(x_2, y_2)$ is computed by Euclidean distance between locations of the two nodes via Eq. (3):

$$d(N_1, N_2) = \text{sqrt}\left((x_2 - x)^2 + \left(y_2 - y_1\right)^2\right); \quad (3)$$

3. The neighboring nodes are the nodes whose Euclidean distance from each other is less than the transmission range (R). They can be represented by Eq. (4):

$$NeighborNode = d(N_1, N_2) \leq R; \quad (4)$$

4. Control packets are sent from sink to source node to obtain information of neighbor nodes, the location of neighbor nodes, neighbor node energy, neighbor node hop count to sink, and distance of neighbor to sink. This information is stored in each node neighbor table. This information is useful for obtaining the initial Q value of the neighboring nodes;
5. The initial Q value $Q(s,a)$ is obtained via Eq. (5):

$$Q(s,a) = \begin{cases} nodeEnergy \cdot EnergyWeight \\ +\dfrac{1}{nodeHopCount \cdot hopCountWeight}. \end{cases} \quad (5)$$

Q value of every node is updated after every iteration based on node energy and its hop count in the neighbor node Q value column. The node with highest Q value is chosen as next best forwarder. Q value is also based on the *hopCountWeight* and *energyWeight* thresholds. This ensures that the neighbor nodes with more hop counts to the sink and less energy are penalized using the respective threshold and not selected as the next forwarder nodes for a better packet loss ratio;

6. Data packets are transmitted from source node. Based on information in its neighbor table, the source node forwards data packet to its neighbor node with highest Q value on the basis of Eq. (2). If any of the nodes have only one neighbor node, the packet will be forwarded, but the reward will be penalized, as represented in Eq. (1);
7. As the packets are forwarder from the source to the next forwarder, the source node will update its energy in the node table. The node energy is updated via Eq. (6):

$$node\_energy = \begin{cases} alpha1 \cdot transmission_{rate} \cdot packet_{size} \cdot 8 \\ +alpha2 \cdot transmission_{rate} \cdot packet_{size} \cdot 8 \cdot dist_{nodes(cs,ns)}{}^{alpha}, \end{cases} \quad (6)$$

where $alpha1 = 50e-9, \%J/bit$, $alpha2 = 0.1e-9, \%J/bit/m^2$, $alpha = 2$, $cs = previous\ node$, $s = next\ \ forwarder\ \ node$ *and* $dist_{nodes}$ is a function to compute Euclidean distance among nodes;

8. The next forwarder will also update energy of source node or previous node in its neighbor table. This ensures the avoidance of self-loops;
9. The data packets are sent from source node via various best forwarder nodes until they reach sink. During data transmission process, nodes near sink may experience congestion. If energy of nodes immediate to sink is not above defined value of *energyThreshold* and no best forwarder node is found, the node drops packet, and data packet cannot reach the sink.

Figure 3 shows the flowchart of the Threshold-Based Reinforcement Learning (TBRL) routing algorithm.

4. RESULTS AND DISCUSSION

4.1. Performance metrics

Following metrics are computed for performance evaluation of proposed approach:

1. Throughput: Calculate the total amount of data transmitted per unit time. It is computed in bytes and represented as in Eq. (7):

$$throughput = (TotalpacketReceive \cdot packetSize)/totalPackets. \quad (7)$$

2. Packet Loss Ratio (PLR): Calculate ratio of lost packets to total number of packets transmitted. It is computed as in Eq. (8):

$$packetLossRatio = TotalpacketLost/totalPackets. \quad (8)$$

3. Packet Delivery Ratio (PDR): Calculate ratio of received packets to the total number of packets transmitted. It is obtained via Eq. (9):

$$packetDeliveryRatio = TotalpacketReceive/totalPackets. \quad (9)$$

4. Network Lifetime Ratio (NLR): Estimate network lifetime based on total energy consumed and energy consumption rate. Energy consumption rate is defined as 0.1 J per transmission. It is calculated via Eq. (10):

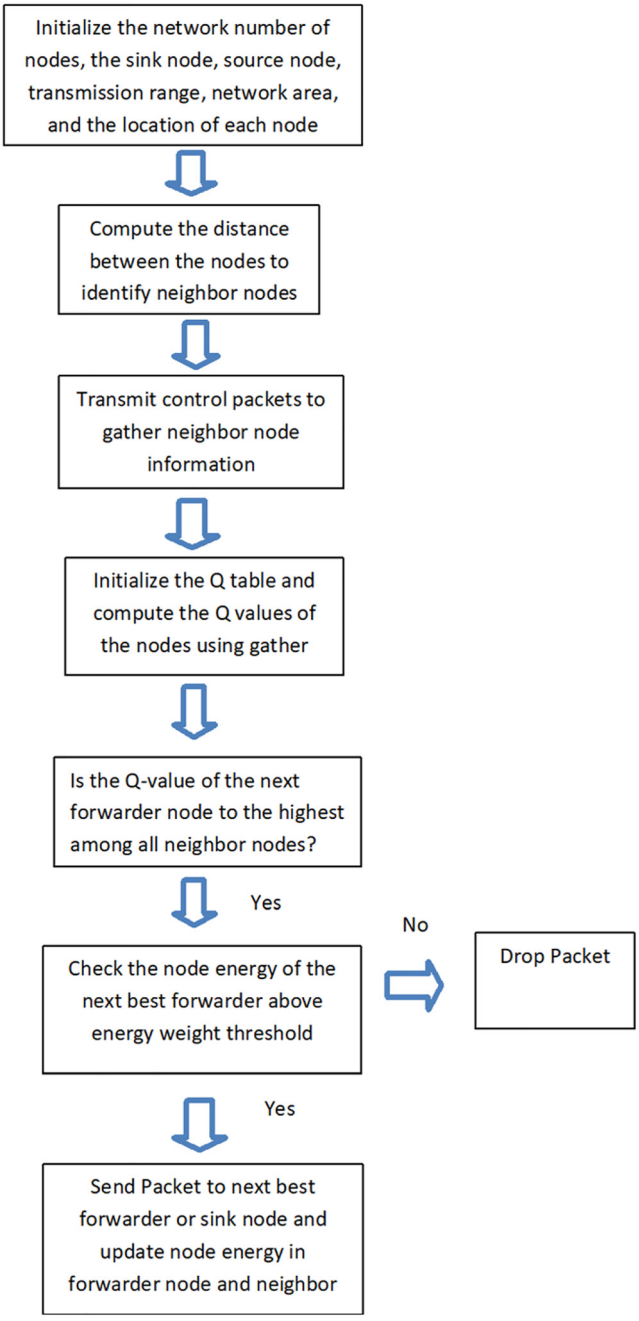

Fig. 3. Flow chart of the proposed threshold-based reinforcement learning algorithm for routing in wireless sensor networks

$$networkLifetime = \frac{totalEnergy}{(energyConsumptionRate \cdot totalPackets)}, \tag{10}$$

where *totalEnergy* is the sum of energies of all nodes.

4.2. Experimental analysis

The experiments are conducted via MATLAB 2022. One sink node and two source nodes are defined in the WSN scenario, with nodes placed randomly for experimental analysis. Constant Bit Rate (CBR) traffic of 5 s duration is attached to the source nodes. The scenario is simulated for a 10 s time interval. Figure 4 shows the wireless sensor network scenario of 50 nodes randomly placed in a 100 × 100 m area with a transmission range of 40 m. Nodes 31 and 19 are the source nodes, as it is shown in Fig. 4, whereas node 50 is the sink node.

The proposed threshold-based reinforcement learning algorithm is compared with the Q-routing simulated for wireless networks for comparison purposes and the RLBR algorithm [10]. Performance of proposed Threshold-Based RL (TBRL) algorithm is evaluated by varying number of nodes from 20 to 100 and observing throughput, PLR, PDR and network lifetime. Figure 5 illustrates performance of proposed threshold-based RL approach by varying number of nodes to observe the packet loss ratio (Fig. 5a), packet drop ratio (Fig. 5b), throughput (Fig. 5c) and network lifetime (Fig. 5d).

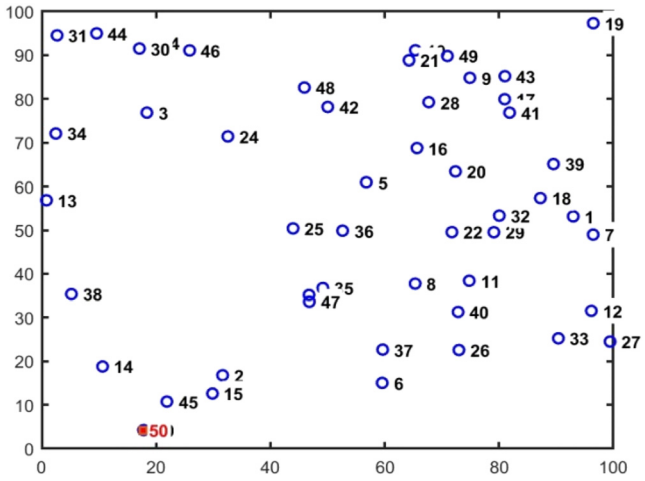

Fig. 4. Wireless sensor network scenario with nodes for simulation along with the source and sink

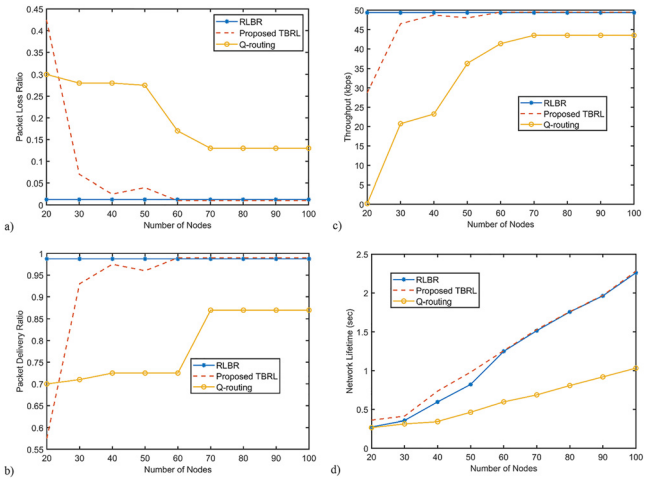

Fig. 5. a) Number of nodes against the PLR, b) number of nodes against PDR, c) number of nodes against throughput, d) number of nodes against the network lifetime

Figure 5a shows that when the number of nodes is lower for proposed TBRL method, packets are dropped because of congestion in the network, and the proposed method does not find any forwarder nodes to forward the packets. As number of nodes increase, performance of the network is equal to that of the RLBR algorithm and better as compared to Q-routing, and no packets are lost. Similarly, Fig. 5b shows that as number of nodes increase, the packet drop ratio increases and is comparable to that of the RLBR algorithm and better than that of the Q-routing algorithm. Figure 5c demonstrates throughput of the proposed threshold-based RL algorithm increases as number of nodes increases, as there is no congestion in the network. When number of nodes is less, owing to thresholds on hop count to sink and node energy of neighbor node, proposed RL algorithm is not able to find next best forwarder node; hence it drops the packets.

Figure 5d exhibits an increased network lifetime compared with that of the RLBR in cases where the number of nodes is less and equal in performance as number of nodes increases. When number of nodes in network is less, owing to energy threshold on neighbor node, the RL algorithm chooses the next best forwarder as the node with a sufficient amount of energy. Hence, the proposed threshold-based RL algorithm demonstrates enhanced network lifetime.

5. CONCLUSIONS

Wireless sensor networks form the backbone for data collection and transmission from remote locations to the base station. Wireless sensor networks face harsh and dynamic environments during data collection and transmission. Owing to changing and constrained environments, the transmission and routing of data packets is still a challenge. Q-learning is a form of reinforcement learning algorithm which learns from the environment and takes actions on the basis of past and current interactions with the environment. This work proposes a new threshold-based Q-learning algorithm. The rewards in the proposed algorithm are penalized by setting thresholds on energy of neighbor node and hop count of neighbor to sink. By means of penalizing the rewards with energy and hop count, the algorithm forwards data packet to next best forwarder node with more energy and a lower hop count to the sink. This ensures better transmission of data packets to sink node and avoids self-loops. Experimental analysis of the proposed threshold-based Q-learning algorithm demonstrates energy efficiency with enhanced network lifetime.

ACKNOWLEDGMENTS

This study was supported by the "University of Debrecen Program for Scientific Publications."

REFERENCES

[1] A. R. Basha, "A review on wireless sensor networks: Routing," *Wireless Personal. Commun.*, vol. 125, pp. 897–937, 2022.

[2] P. Naya, G. K. Swetha, S. Gupta, and K. Madhavi, "Routing in wireless sensor networks using machine learning techniques: Challenges and opportunities," *Measurements*, vol. 178, 2021, Art no. 108974.

[3] D. P. Kumar, T. Amgoth, and C. S. R. Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey," *Inf. Fusion*, vol. 49, pp. 1–25, 2018.

[4] N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev, "Reinforcement learning for combinatorial optimization: A survey," *Comput. Oper. Res.*, vol. 134, 2021, Art no. 105400.

[5] J. Czech, "Distributed methods for reinforcement learning survey," in *Reinforcement Learning Algorithms: Analysis and Applications*, B. Belousov, H. Abdulsamad, P. Klink, S. Parisi, and J. Peters, Eds, Berlin, Germany: Springer, 2021, pp. 151–161.

[6] S. Pateria, B. Subagdja, A. H. Tan, and C. Quek, "Hierarchical reinforcement learning: A comprehensive survey," *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–35, 2021.

[7] A. T. D. Perera and P. Kamalaruban, "Applications of reinforcement learning in energy systems," *Renew. Sustain. Energy Rev.*, vol. 137, 2021, Art no. 110618.

[8] Z. Mammeri, "Reinforcement learning based routing in networks: Review and classification of approaches," *IEEE Access*, vol. 7, pp. 55916–55950, 2019.

[9] B. Djail, W. K. Hidouci, and M. Loudini, "A comparative evaluation of techniques for N-way joins in wireless sensors networks," *Pollack Period.*, vol. 15, no. 2, pp. 13–24, 2020.

[10] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," in *Proc. 7th Int. Conf. Neural Inf. Process. Syst.*, Denver Colorado, US, November 29 – December 2, 1993, pp. 671–678.

[11] L. Hajdu, B. Dávid, and M. Krész, "Gateway placement and traffic load simulation in sensor networks," *Pollack Period.*, vol. 16, no. 1, pp. 102–108, 2021.

[12] P. Wang and T. Wang, "Adaptive routing for sensor networks using reinforcement learning," in *The Sixth IEEE International Conference on Computer and Information Technology*, Seoul, Korea (South), September 20–22, 2006, pp. 219–219.

[13] N. Ouferhat and A. Mellouk, "Optimal QoS and adaptative routing in wireless sensor networks," in *Proc. 2nd IEEE International Conference on Information and Communication Technologies*, Damascus, Syria, April 24–28, 2006, pp. 2736–2741.

[14] X. Liang, I. Balasingham, and S. S. Byun, "A multiagent reinforcement learning based routing protocol for wireless sensor networks," in *IEEE International Symposium on Wireless Communication Systems*, Reykjavik, Iceland, October 21–24, 2008, pp. 552–557.

[15] S. Z. Jafarzadeh and M. H. Y. Moghaddam, "Design of energy-aware QoS routing protocol in wireless sensor networks using reinforcement learning," in *IEEE 27th Canadian Conference on Electrical and Computer Engineering*, Toronto, ON, Canada, May 4–7, 2014, pp. 1–5.

[16] W. J. Guo, C. R. Yan, Y. L. Gan, and T. Lu, "An intelligent routing algorithm in wireless sensor networks based on reinforcement learning," *Appl. Mech. Mater.*, vol. 678, pp. 487–493, 2014.

[17] Y. Akbari and S. Tabatabaei, "A new method to find a high reliable route in IoT by using reinforcement learning and fuzzy logic," *Wireless Pers Commun.*, vol. 112, pp. 967–983, 2020.

[18] C. M. Horvath, J. Botzheim, T. Thomessen, and P. Korondi, "Bacterial memetic algorithm trained fuzzy system-based model of single weld bead geometry," *IEEE Access*, vol. 8, pp. 164864–164881, 2020.

[19] C. M. Horvath, J. Botzheim, T. Thomessen, and P. Korondi, "Bead geometry modeling on uneven base metal surface by fuzzy systems for multi-pass welding," *Expert Syst. Appl.*, vol. 186, 2021, Art no. 115356.

[20] A. F. E. Abadi, S. A. Asghari, M. B. Marvasti, G. Abaei, M. Nabavi, and Y. Savaria, "RLBEEP: Reinforcement-Learning-Based Energy Efficient Control and routing protocol for wireless sensor networks," *IEEE Access*, vol. 10, pp. 44123–44135, 2022.

[21] Z. Liu and X. Wang, "Energy-balanced routing in wireless sensor networks with reinforcement learning using greedy action chains," *Soft Comput.*, pp. 1–21, 2023.

[22] D. Prabhu, R. Alageswaran, and S. M. J. Amali, "Multiple agent based reinforcement learning for energy efficient routing in WSN," *Wireless Networks*, vol. 29, pp. 1787–1797, 2023.

[23] A. Keerthika and V. B. Hency, "Reinforcement-learning based energy efficient optimized routing protocol for WSN," *Peer-to-Peer Netw. Appl*, vol. 15, pp. 1685–1704, 2022.