OPEN ACCESS JOURNALS
Indiana Journal of Multidisciplinary Research
Abbreviated Tittle- Ind J Multi Res
ISSN *(Online)*- 2583-3820
Journal Homepage Link- https://indianapublications.com/Journals/IJMR
DOI: https://doi.org/10.5281/zenodo.21557339

INDIANA PUBLICATIONS
PRODUCTIVE AND QUALITY RESEARCH

Research Article

Volume-06|Issue-04|2026

An Energy Efficient Deep Learning Approach to Find Optimal Path in Underwater Networks Using ns3-AI

Dr. Shruthi K R[1] and Dr. Kavitha C[2]

[1]Dr. Shruthi K R, Assistant Professor, Department of Artificial Intelligence & Data Science, BMS College of Engineering, Bengaluru, Karnataka, India
[2]Dr. Kavitha C, Professor & Head, Department of Computer Science & Engineering, RNS Institute of Technology, Bengaluru, Karnataka, India.

Article History
Received: 05.05.2026
Accepted: 22.06.2026
Published: 25.07.2026

Citation
Shruthi, K. R. & Kavitha, C. (2026). An Energy Efficient Deep Learning Approach to Find Optimal Path in Underwater Networks Using ns3-AI. *Indiana Journal of Multidisciplinary Research, 6*(4), 22-30.

Abstract: Undersea communication plays a vital role in numerous applications but remains unstable due to its intermittent and noisy characteristics. To address these challenges, this paper proposes an environment-aware routing approach based on reinforcement learning. An underwater sensor network is simulated using ns-3, while agent training is performed through a deep learning framework integrated via ns3-ai. The framework leverages data from the ns-3 simulator to optimise routing decisions, with agent actions fed back into packet forwarding. The proposed model introduces dual reward strategies to enhance learning efficiency and employs the Huber loss function to mitigate the influence of outliers. Furthermore, an energy-efficient mechanism is incorporated to extend network lifetime and reduce overall transmission energy. The results were analyzed with respect to various parameters. The results demonstrate that the proposed approach significantly outperforms conventional Q-learning in terms of delay and packet delivery time.

Keywords: Reinforcement Learning, Deep Learning, Q learning, ns3-ai, Underwater Networks

INTRODUCTION

Underwater networks are the networks that monitor pollution, seismic and collect oceanographic information undersea [1]. Communication undersea is needed to gather this information. Acoustic communication is best suited for underwater networks [2]. These acoustic networks are classified as delay-tolerant networks [5]. Because of its intermittent and noisy nature, routing becomes an essential component in these networks. A routing approach that understands the environment is needed. One such routing approach is reinforcement learning, a class of machine learning algorithms.

Reinforcement learning is the learning paradigm that trains the agent to learn from its experience. Reinforcement learning can handle complex environments and is applied to various networking applications [6]. Here, the authors have chosen the deep Q-learning method, a class of reinforcement learning, to design a routing approach for the underwater sensor networks. The applications of Deep Q learning are many, including Cloud computing [13], routing [8], and many more. Here, Deep learning is used to discover the optimal path between the source and the destination. Q-learning algorithm trains the agent by using the Q table and maximizes its rewards appropriately. This method works fine for smaller environments. Deep Q-learning can be used for larger networks and to maximize efficiency. Deep Q-learning is a reinforcement learning method that trains the agent using a neural network approach. The fundamental step of Deep Q learning is that the initial state is fed into the neural network. Deep Q learning fetches the Q-value of all possible actions as an output, as shown in Figure 1.

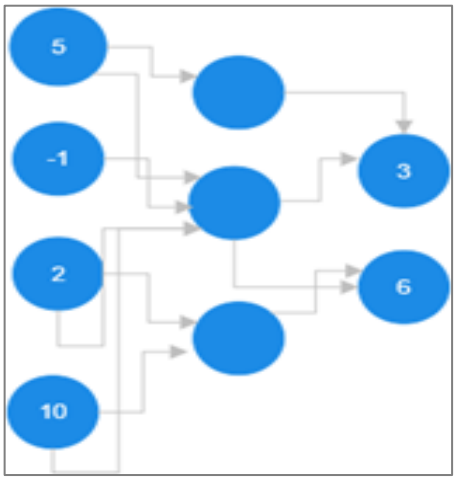

Fig. 1. Deep Q learning Example

In deep Q learning, the agent stores experiences in the form of a tuple (s, a, r, s'). Once the agent reaches the threshold, it starts learning from the batch of experiences. The agent selects randomly a uniformly distributed sample from the batch. Now the agent chooses an appropriate action either by using epsilon greedy, epsilon

***Corresponding Author:** Dr. Shruthi K R

soft, or the softmax function. The exploitation and exploration tradeoff strategy is managed by choosing an appropriate action based on these methods. Exploitation fetches more rewards to the agent as it selects the best path based on available information. Exploration allows the agent to explore new paths to choose an optimal path. In this paper, the authors have used Deep Q learning with appropriate action selection methods to find an optimal path from source to sink.

The ns3-ai framework allows high-speed data exchange between the ns3 simulator and the AI engine. There are two components in the ns3–ai framework. The first is a C++-developed ns3 interface, and the second is a Python-developed AI interface. To enable data communication between these two interfaces or different processes, data has to pass through the buffer in the kernel. A simple example of the same is process X can put the data in the kernel buffer, and process Y can read the data from the same [4]. There are different ways of establishing communication between two different processes. Communication can be established either through sockets or pipes [7]. But in ns3-ai, a shared memory pool is used to establish communication between processes. A shared memory pool is the fastest way to exchange data between processes.

In our paper, the authors have simulated an underwater environment using Aqua-sim-ng as an ns3 interface. Information regarding sensor nodes (including source and destination), packets, simulation time, and other related information is stored in a shared buffer. The AI interface reads the data present in the memory pool and applies the DL algorithm. The algorithm trains the agent and chooses the appropriate action. The optimal action is again placed into a shared memory pool. Now, the ns3 interface reads the buffer and uses the optimal action specified by the algorithm. Now, the sensor node can use this action to route the packets. Along these lines, to determine the best route between the source and destination in underwater sensor networks, a deep learning approach is employed.

The remainder of the document is structured as follows: Section II describes a survey of related work. Section III outlines the implementation details of the proposed method. Section IV illustrates the results of the proposed method. Section V of the report provides a conclusion and recommendations for further investigation.

RELATED SURVEY

Deep Q learning, a class of reinforcement learning, is an appropriate strategy for an uncertain environment. This helps to monitor the environment continuously and take appropriate action. This section describes the use of DL algorithms in various network scenarios. This section also describes several applications using ns3-AI.

Deep neural networks are used for network routing in [8]. The authors of this article have presented a DL model that trains on the best decisions for flows using inputs of known traffic demands. Here, the authors have used two objective functions to minimise congestion in the network. These include maximum link utilization and Fortz and Thorup's congestion measure technique. This model resulted in a quasi-optimal performance with two objectives.

Opportunistic IoT networks belong to a class of IoT where humans and machines work together in a network to share data. A Deep Q Learning model to address security attacks, including hello flood, sinkhole, and distributed denial of service, has been designed by the authors in [9]. Deep Q learning-based secure routing protocol is used to learn the behavior of the nodes and later predict the said attacks that might occur in the network. Here, the algorithm is divided into two phases: the training phase and the prediction phase. In the training phase, preprocessing, feature extraction, feature normalization, and feature selection are completed. The normalized data is divided into training and testing datasets. The training dataset is given to the DQN model. Here, a multiple-agent approach is employed instead of an experience relay to reduce memory requirements. The training is done several times until the error is minimal. Finally, the nodes with a higher Q value are preferred to route the packets. Thus, authors in [9] conclude that Deep Q learning based secure routing secures the network by predicting attacks.

Authors in [10] have designed a powerful deep Q extreme learning for underwater communication. An adaptive reward mechanism is used in this paper to improve the packet delivery ratio and throughput. A powerful firefly routing mechanism is also used in this paper to gain energy efficiency.

In this paper, the authors have used Yang's [11] firefly equation to define an energy-efficient routing mechanism.

An adaptive deep learning mechanism has been designed by the authors in [12]. Here, a deep Q algorithm with an on and off policy strategy is used to achieve optimal routing decisions. The energy and depth information of the nodes is considered for finding the Q value of the nodes. Nodes with maximum Q value are chosen as forwarders. In order to reduce network overhead, a combination of unicast and broadcast communication is used by the authors in [12].

In [14][24], the authors created an LSTM model based on an attention mechanism. The input features are improved through the usage of the attention method. This is given as input to the LSTM model, which helps to predict the channel with better accuracy. The use of an attention mechanism prior to the LSTM helps in learning the model better.

***Corresponding Author:** Dr. Shruthi K R

Spectral Social Spider Optimization (SSSO) feature selection method, along with multi-path routing, is proposed in [15]. Here, data from the standard repository is used to reduce delay and for better feature selection. Initially, redundant data and missing values are dealt with in the cluster head as part of preprocessing. Then, feature selection is done using the SSSO algorithm. The same algorithm is used in data delivery. I- DES algorithm is used in encrypting and decrypting the data. After decrypting the data, feature weights are validated using Softmax Neuron Classifier. Finally, the Re-cursive Spectral Neural Network is used to enable the paths for transmission. At the sink or at the receiver, the hop count and decryption key are used in the decryption of the data. So, the methods discussed enhance the security in UWASN.

Energy-efficient routing protocol for underwater sensor networks using meta-heuristic algorithm [21] finds an optimal path destination with minimal time. This algorithm is based on the genetic algorithm solution. Here, authors have used a combination of global search and local search algorithms to find the best route to the destination. The routing framework here [21] is divided into 4 sections, namely energy management, neighboring sensor management, information management, and routing management. Neighboring management helps to collect information about neighbors. The data management section performs data retrieval and data storage. Routing management helps in the routing process using the algorithm described. Energy management helps to find the energy consumed by each sensor node as soon as it sends the information to other nodes.

Table 1: Use of Deep Learning Algorithms in UWSN

Authors	Protocol Used	Problems Handled
8	Deep neural networks for network routing	Traffic congestion
9	Deep Q learning for secured routing	Securing the network against attacks
10	A Deep extreme Q learning firefly energy efficient protocol	Packet delivery ratio and throughput
12	An Adaptive Deep Q- Network Based Energy and Latency Aware Routing Protocol	Better Latency and Energy efficiency
14	LSTM Based Attention based model	Channel prediction accuracy
15	Deep learning based multipath routing protocol	Encryption in underwater wireless sensor networks
21	Energy efficient routing protocol using hybrid metaheuristic algorithm	Optimal path in a minimal time.
22	Energy efficient Clustering Protocol	Energy efficiency and network lifetime

Table 1 portrays various deep learning techniques used in multiple applications. The table also conveys the problems handled by deep learning techniques. However, deep learning is not typically used for routing in underwater wireless communication. Environment-aware routing is indeed in need of an hour in underwater wireless networks.

In the paper, the deep learning method addresses the routing issues in underwater networks by using reward strategies and energy energy-efficient model to find a better route from source to destination and to increase the network's lifetime.

As discussed in Section I, the ns3-ai framework is used for the communication between the ns3 simulator and the RL algorithm. In this section, let us discuss the reason for choosing ns3-ai for the research.

The Ns3-gym framework was proposed by the authors in [7]. The OpenAI gym toolkit is used for many of the AI applications. Numerous network simulations are done in the ns3 simulator. However, there was a need to integrate RL and ns3. The ns3-gym framework has been designed to integrate ns3 and RL. ZeroMQ Sockets mechanism was used as an inter-process communication in ns3gym. This resulted in a slower communication rate between the ns3 simulator and the RL algorithm.

Ns3-ai framework, as proposed by the authors in [4], used a shared memory pool for the communication between ns3 and the AI algorithm. This increased the communication speed to a larger extent.

In this paper, the authors have proposed a deep Q-learning-based approach using the ns3-ai framework for underwater sensor networks.

MATERIALS AND METHODS

Deep Q learning is one of the efficient RL algorithms for learning the environment faster. The main advantage of Deep Q learning is that it learns from a batch of experiences. As a result, Deep Q learning is able to predict Q values more quickly and may be applied to bigger networks. Here, authors have designed Deep Q learning for underwater networks using the ns3-ai framework. Let us discuss the details of implementation in this section.

Deep Q Learning

Deep Q learning uses a neural network to approximate Q values. The sensor's current state is provided as an input, and the Q value for every action that could be taken is created as an output. Figure 2 portrays the difference between Q values generated by Q learning and Deep Q learning.

***Corresponding Author:** Dr. Shruthi K R

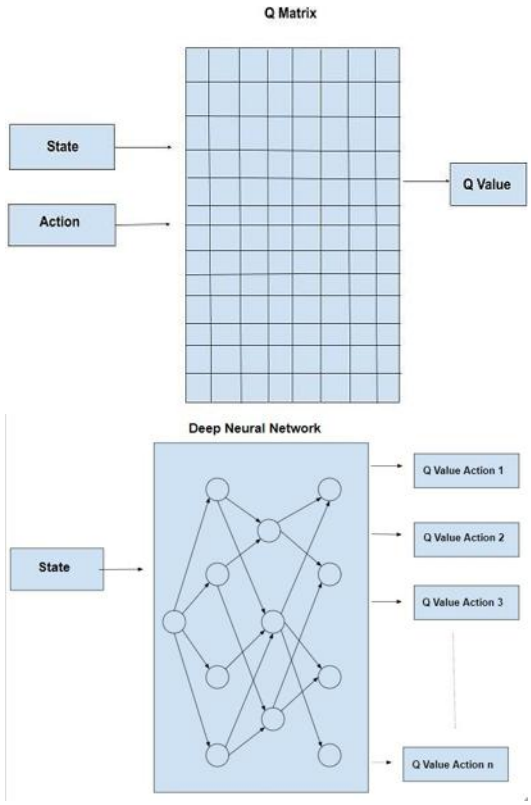

Fig. 2. Difference between Q learning and Deep Q learning

Deep Q learning uses the SARSA equation [16] to predict the Q value action in every state. Equation 1 is used in the design of Deep Q learning for underwater networks.

$$\boldsymbol{Q(s_t\ , a_t) = Q(s_t\ , a_t) + \propto [R_{t+1} + \gamma * max_a Q(s_{t+1}\ , a_t) - Q(s_t\ , a_t)\ ]} \quad \textbf{(1)}$$

Q(st,, at) – current state and action α – learning rate
γ – discount factor R – reward earned
Q(st+1,, at) – next state and action

Steps in Deep Q network

1. Preprocess and feed the state of the sensor node image to the deep neural network. It will return the Q-values of all possible actions of the state.
2. Select the action using the epsilon-greedy policy or geolocation as described in [3].
3. Perform the action selected in Step 2 in state **s** and move to state **s′**. This then becomes the preprocessed image of the next sensor node image. This transition is stored in the replay buffer as **⟨s, a, r, s′⟩**.
4. This replay (experience) buffer is used for learning the model. A uniform sample distribution from the buffer is taken for learning purposes.
5. Calculate the loss after learning the model as in Equation (2).

$\delta =$ threshold

If

$$\delta < (Q(s', a'; \theta') - Q(s, a; \theta))$$

then

$$\text{loss} = \left(\alpha + \gamma \max_{a'} Q(s', a'; \theta') - Q(s, a; \theta)\right) \quad (2)$$

else

$$\text{loss} = \frac{1}{2}\left(\max_{a'} Q(s', a'; \theta') - Q(s, a; \theta)\right)^2 \quad (3)$$

6. This represents the Huber loss difference between the target and predicted Q-value.
7. After **n** iterations, update the network weights as calculated in the loss function (Equation 2).
8. Repeat the above steps for **M** episodes.
9. Using the above procedure, Deep Q-learning helps predict Q-values for every state.

Huber Loss Model

The Huber loss method takes advantage of Mean Absolute Error (MAE) and Mean Squared Error (MSE). It calculates the loss based on the threshold value, **δ**. The loss model uses Mean Absolute Error if the loss is large, and for minor errors, the loss model is calculated using Mean Squared Error. The concept of the Huber loss model is used in the calculation of loss in Deep Q Learning.

If

$$| f(x) - y | \leq \delta$$

$$L(\delta, y, f(x)) = \frac{1}{2}(f(x) - y)^2$$

If

$$| f(x) - y | > \delta$$

$$L(\delta, y, f(x)) = \delta | f(x) - y | - \frac{1}{2}\delta^2 \quad (4)$$

Equation (4) represents the general Huber loss model. Equations (2) and (3) represent the application of the Huber loss model for Deep Q Learning. The loss is computed using Equation (2), which represents the absolute error difference, if the errors are greater than the threshold value. Equation (3) penalizes minor errors using the absolute error difference. The calculation of the loss model using the Huber loss model has illustrated finding an optimal path with lesser delay. Section IV depicts the results of Deep Q Learning compared to other models.

Energy Efficient Model

Sensor node energy is very crucial in UWSNs, as described in Section I. It is essential to design an energy-efficient model for underwater sensor networks that considers the nodes with optimal energy to forward the packet. The authors have calculated rewards using other observation parameters and the residual energy of the node. Two different reward strategies are employed in this paper. If the node has sufficient energy to forward the packet, Equation (5) is used to calculate the reward. Otherwise, an alternative method that considers the residual energy of the node is used in reward calculation.

$$\text{Reward} = \left(\frac{\text{currentenergy}}{\text{initialenergy}}\right) \times pr \quad (6)$$

***Corresponding Author:** Dr. Shruthi K R

Equation (6) demonstrates how the reward is determined using the node's energy. Given the significant influence that packet priority has in underwater networks, the reward computation takes this into account as well. The appropriate Q-value based on the reward is calculated for every neighboring node, and the Deep Q agent selects the optimal node as the forwarder of the packet based on the aforementioned parameters. Thus, the overall network model increases the lifespan of the network. Section IV depicts the results that showcase stabilized throughput even when the load on the network increases.

Deep Q-Based Routing

Deep Q-based routing is implemented using ns3-ai. The structure of the environment, here the underwater environment, is simulated using the ns-3 simulator. The parameters, sensor states, and environment of the underwater environment created by the ns-3 simulator are given as input to the Deep Q-learning algorithm. Deep Q-learning analyzes the parameters to approximate Q-values for the appropriate state. Algorithm 1 describes Deep Q-based routing for finding an optimal path from source to sink.

Algorithm 1: Deep Q-Based Routing

Step 1: Simulate a **1500 × 1500 m** underwater environment in ns-3.

Step 2: Initialize environment parameters such as bandwidth, delay, duration, and the number of source nodes.

Step 3: The environment and its parameters are sent to the RL algorithm (Python side) via the message interface.

Step 4: Select the DQN agent algorithm.

1. Initialize **s, a, r, s′**.
2. Store the above initialized values as a transition.
3. If stored transitions reach memory capacity:
   - Learn the neural network.
4. Else:
   - Initialize **s** as the source state.
5. Use observation parameters as command window, segments acknowledged, bytes inflight, and residual energy.
6. Calculate reward.

If

energy of sensor node > threshold

$$\text{Reward} = \text{segments acknowledged} - \text{bytesinflight} \quad (5)$$

Else

$$\text{Reward} = \left(\frac{\text{currentenergy}}{\text{initialenergy}}\right) \times pr \quad (6)$$

- Find the action using Equation (1) in the neural network.
- Find the next state **s′**.
- Store the transition as **(s, a, r, s′)** in the memory buffer.
- Repeat the above steps for **m** episodes.

The NS3-ai framework is used to simulate the above-described algorithm. It contains two interfaces- ns3, Python (i.e RL algorithm). A 1500 * 1500m underwater environment is created using ns3. Acoustic sensor nodes are created using Aquasim NG. Node id's and socket id's are created as a part of initialization. Environment parameters are set according to Table 1. The data and events are generated at the source end. In order to route the data to the destination or sink node, deep Q-learning is used to find the next optimal node. The ns3-ai framework uses a message passing method to transfer the environment's status and parameters to the Python interface. In the RL part, a deep Q-learning agent is used to analyze the parameters received from ns3 and find the optimal path.

In deep Q learning, every transition is stored in the buffer as a quadruple (s, a, r, a'). s represents the state; here, it represents the source node that sends the data. a represents action; here, it represents to which node the data has to be transmitted. r represents reward; here, it calculates the reward earned by the node for choosing an appropriate action. s' represents the next target; here, it represents the next node to which the data is forwarded. Let us discuss the working of deep Q-learning in finding the appropriate action. Initially, the parameters of the RL environment state, action, reward, and next state are initialized. Every transition is stored in the buffer. If the number of transitions exceeds the defined capacity, then the neural network takes a uniformly distributed sample from the stored transitions. The neural network then uses this sample to learn and train the model using Equation 2. Otherwise, using observation parameters like segments acknowledged, segment size, and bytes in flight, q values, and action are calculated using equation 1. Then, an appropriate reward is calculated as discussed in the algorithm. It then identifies the next state based on the action selected. This transition is stored in the buffer for further use.

Q value, along with action, is stored in the shared memory pool that is shared by the ns3 and ai interfaces. Ns3 simulator now reads the value stored in shared memory and uses the same to act on the source node. Source node now forwards the data to the node as suggested by the deep Q-learning algorithm (ai interface). The updated parameters are again sent to the AI interface using the shared memory pool. The process is repeated for 'm' no. of episodes.

RESULTS

Figure 3 illustrates that the underwater environment is simulated in ns3 using Aqua-sim-ng with acoustic sensor nodes. The topology can have n no. of source nodes and 1 sink node. In this case, the source nodes' data must be sent to the sink. Deep Q-based routing finds the optimal neighbor to forward the packet to the destination. The Deep Q Agent, as described in Algorithm 1, finds the next nearest neighbor. The source node then forwards the

***Corresponding Author:** Dr. Shruthi K R

packet to the selected neighbor node. The same pro-cess continues until the packet reaches the destination.

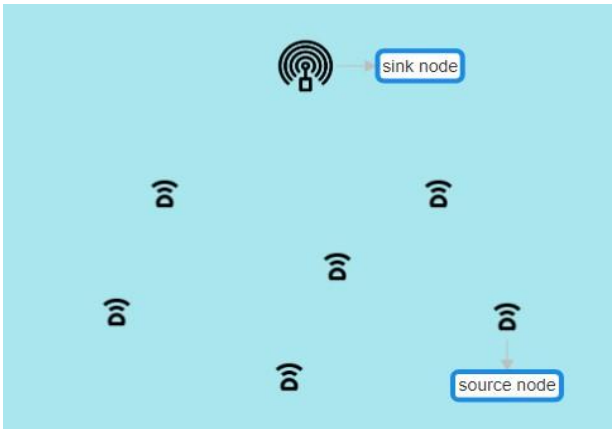

Fig. 3. Topology of the underwater environment

The simulation settings listed in Table II are used to model the topology shown in Figure 3.

Table 2: Simulation Parameters

Parameters	Value
Discount Factor, γ	0.2
Learning Rate, α	0.0001
Simulation	10000
Environment Width * Height	1500m*1500m
Bandwidth	10Mbps
Delay	20ms
Sensor Buffer size	4Mb

Let us now see the simulation results of Deep Q-based routing for the simulation parameters (Table II) and topology (Figure 3) described above. The quantity of packets that arrive at their destination is the primary determinant of the algorithm's efficiency. Figure 4 illustrates the number of packets acknowledged.

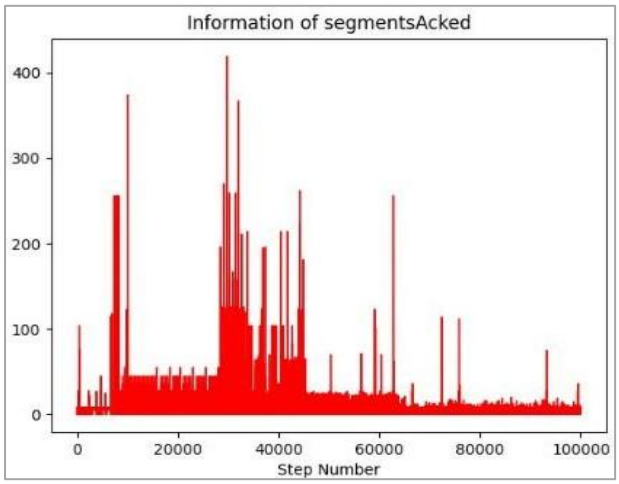

Fig. 4. No. of segments acknowledged

The sensor nodes in this scenario generate the packets at random. Eventually, these packets are sent to their intended location. The number of packets in the transition is shown in Figure 5.

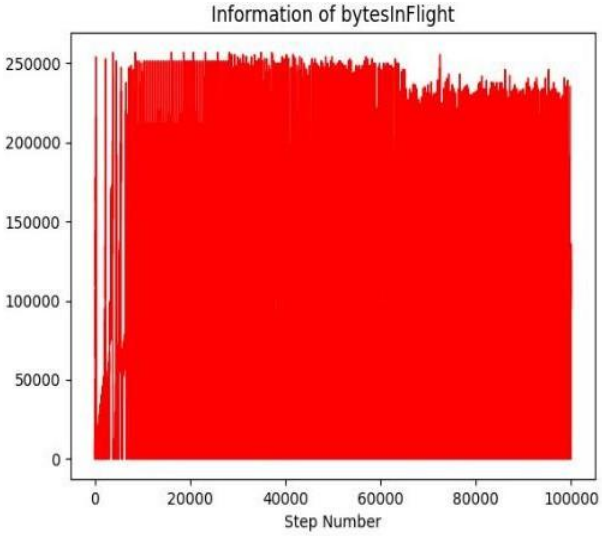

Fig. 5. No. of bytes in transition

Figure 6 represents throughput and goodput at various nodes for the simulation parameters depicted in Table

2. Throughput represents the total amount of data transmitted through the network, which includes even overhead data, whereas Goodput represents the useful data transmitted through the network. Deep Q learning is successful in transmitting the packets faster than Q learning and a TCP-based algorithm. TABLE III depicts the comparison parameters of TCP-based, Q learning, and Deep Q-based learning algorithms. As observed from TABLE 3, the Deep Q learning algorithm can transmit a greater number of packets to the destination.

Comparison of all these algorithms was computed using the message interface using ns3ai as described in the Introduction section.

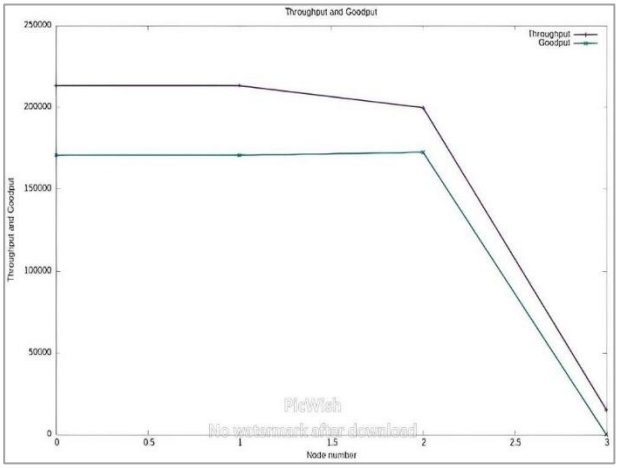

Fig. 6. Throughput and Goodput at various nodes

Table 3. Comparison of Algorithms

Parameter	TCP Based	Q Learning	Deep Q-based
Throughput	2.22654Mbps	2.26527Mbps	2.8Mbps
Delay	22ms	25ms	20ms

***Corresponding Author:** Dr. Shruthi K R

The rest of the comparisons in the preceding sections were done using the gym interface [23]. Gym interface is an open-source library used for comparing RL algorithms. Further comparisons were analyzed using the gym interface. Learning rate and discount factors play a crucial role in calculating the Q value in deep learning. It controls how much to change the model in response to the estimated error each time the model weights are updated. TABLE 4 portrays the throughput and delay results for various learning rates.

Table 4. Throughput, Delay, And Packet Loss Ratio Results for Varying Learning Rate

Sl No.	Learning Rate	Throughput (KB/s)	Delay (ms)	Packet loss Ratio
1	0.0001	160337.3581	178.07	0.94%
2	0.0002	159443.2882	120.11	0.23%
3	0.0003	159040.5614	114.13	0.23%
4	0.0004	167492.8557	218.80	1.17%
5	0.0005	162389.4431	107.15	0.12%

It can be observed from Table 4 that as the learning rate increases, the model is able to find the shorter route in a minimal amount of time and with nominal throughput. It was also observed that the packet loss ratio is minimized to a greater extent. The throughput, latency, and packet loss ratio results for different discount factor values are shown in table 5. In essence, the discount factor controls the reinforcement learning agent's concern for rewards in the distant future in comparison to those in the near future.

Table 5. Throughput, Delay and Packet Loss Ratio Results for Varying Discount Factor

Sl No.	Discount Factor	Throughput (KB/s)	Delay (ms)	Packet loss Ratio
1	0.2	1829019.1526	148.33	0.54%
2	0.4	152265.6639	178.61	0.77%
3	0.6	156251.2964	105.00	0.12%
4	0.8	162389.4431	107.15	0.12%

It can be observed from Table 5 that increasing the discount factor to 0.8 has fetched minimal packet loss and enabled transmission of packets with minimal delay and nominal throughput. In this case, the authors found that the model's learning rate of 0.0001 and discount factor of 0.8 yielded better results.

Table 6 illustrates the values of throughput for different numbers of nodes. The sustainability of the network can be analyzed by increasing the number of nodes. The network needs to be able to send a sufficient number of packets even when there is a lot of traffic on it. The following table demonstrates the throughput of the network as the number of nodes increases.

Table 6. No. of Nodes V/s Throughput

Sl No.	Nodes	Throughput
1	4	162389.4431
2	6	192636.6108
3	8	160090.1622
4	10	135662.5794

It can be observed from the table that the topology simulated in ns3 can send a good number of packets even when the number of nodes has increased.

DISCUSSION

Huge applications in underwater have resulted in tremendous research interests in underwater wireless sensor networks [2]. So, the routing of packets in these networks is essential [18]. In this paper, the authors have designed a Deep Q-based routing that helps in finding an optimal path between the source and destination. Deep Q learning observes the environment and suggests the optimal next node for the transmission. Deep Q learning uses SARSA and a loss function to update Q values and find the best action. Section IV illustrates the results of Deep Q-based routing for the underwater sensor networks. The algorithm has outperformed other baseline algorithms, including Q learning and TCP routing, in terms of packet loss ratio and delay. The authors have experimented with varying learning rates and discount factors, which majorly affect the deep learning model. Using these results, the nominal values of learning rate and discount factor can be used for further research. The simulations were also experimented with an increasing number of sensors. The results conveyed that the network was sustainable enough for transporting data even if there was traffic on it. Hence, the deep learning model was resourceful in finding the best route from source to destination with minimal delay and loss. This outperforms other terrestrial routing algo-rithms and Q learning used for routing in underwater sensor networks[3]. The two reward strategies based on the energy of the sensor node elevate the performance of the model in selecting the best hop and transmitting packets to the destination with less delay.

CONCLUSION

Here, we conclude that Deep Q-based routing understands the environment before forwarding any data and eventually results in the optimal path towards the destination. So, Deep Q learning is best suited for routing in underwater sensor networks.

***Corresponding Author:** Dr. Shruthi K R

Bio-inspired algorithms have huge applications in various fields [19]. The research can be extended to apply bio-inspired algorithms in underwater sensor networks. These methods are applicable to underwater sensor networks since they have demonstrated superior performance in UAV routing.

ACKNOWLEDGEMENTS

I would like to thank BMS College of Engineering for the support and encouragement.

REFERENCES

1. Shruthi, K. R., & Kavitha, C. (2022). Reinforcement learning-based approach for establishing energy-efficient routes in underwater sensor networks. In *Proceedings of the 8th International Conference on Electronics, Computing and Communication Technologies (CONECCT 2022)*.
2. Shruthi, K. R. (2022). An artificial intelligence-based routing for underwater wireless sensor networks. In *Proceedings of the 4th International Conference on Electrical, Electronics, Communication, Computer Technologies, and Optimization Techniques*.
3. Shruthi, K. R., & Kavitha, C. (2025). Reinforcement learning based approach for underwater environment to evaluate agent algorithm. *Wireless Personal Communications, 145*, 93–111. https://doi.org/10.1007/s11277-025-11845-w
4. Yin, H., Liu, P., Liu, K., Cao, L., Zhang, L., Gao, Y., & Hei, X. (2020). Ns3-ai: Fostering artificial intelligence algorithms for networking research. In *Proceedings of the 2020 Workshop on ns-3 (WNS3 '20)* (pp. 57–64). ACM. https://doi.org/10.1145/3389400.3389404
5. Zhang, Z., Lin, S.-L., & Sung, K.-T. (2010). A prediction-based delay-tolerant protocol for underwater wireless sensor networks. In *Proceedings of the 2010 International Conference on Wireless Communications and Signal Processing (WCSP)* (pp. 1–6). IEEE.
6. Luong, N. C., Hoang, D. T., Gong, S., Niyato, D., Wang, P., Liang, Y.-C., & Kim, D. I. (2019). Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials, 21*(4), 3133–3174. https://doi.org/10.1109/COMST.2019.2916583
7. Gawłowicz, P., & Zubow, A. (2018). ns3-gym: Extending OpenAI Gym for networking research. *arXiv*. https://arxiv.org/abs/1810.03943
8. Reis, J., Rocha, M., Phan, T. K., Griffin, D., Le, F., & Rio, M. (2019). Deep neural networks for network routing. In *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). IEEE. https://doi.org/10.1109/IJCNN.2019.8851733
9. Kandhoul, N., & Dhurandher, S. K. (2022). Deep Q-learning-based secure routing approach for OppIoT networks. *Internet of Things, 20*, 100597. https://doi.org/10.1016/j.iot.2022.100597
10. Anitha, D., & Karthika, R. A. (2021). DEQLFER—A deep extreme Q-learning firefly energy-efficient and high-performance routing protocol for underwater communication. *Computer Communications, 174*, 143–153. https://doi.org/10.1016/j.comcom.2021.04.030
11. Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. In O. Watanabe & T. Zeugmann (Eds.), *Stochastic Algorithms: Foundations and Applications* (Lecture Notes in Computer Science, Vol. 5792, pp. 169–178). Springer. https://doi.org/10.1007/978-3-642-04944-6_14
12. Su, Y., Fan, R., Fu, X., & Jin, Z. (2019). DQELR: An adaptive deep Q-network-based energy- and latency-aware routing protocol design for underwater acoustic sensor networks. *IEEE Access, 7*, 9091–9104. https://doi.org/10.1109/ACCESS.2019.2891590
13. Shin, D.-J., & Kim, J.-J. (2021). Deep reinforcement learning-based network routing technology for data recovery in exa-scale cloud distributed clustering systems. *Applied Sciences, 11*, 8727. https://doi.org/10.3390/app11188727
14. Zhu, Z., Tong, F., Zhou, Y., et al. (2023). Deep learning prediction of time-varying underwater acoustic channel based on LSTM with attention mechanism. *Journal of Marine Science and Application, 22*, 650–658. https://doi.org/10.1007/s11804-023-00347-5
15. Prakash, K., & Sathya, S. (2023). A deep learning-based multi-path routing protocol for improving security using encryption in underwater wireless sensor networks. In *Proceedings of the 4th International Conference on Electronics and Sustainable Communication Systems (ICESC 2023)* (pp. 581–588). IEEE. https://doi.org/10.1109/ICESC57686.2023.10193733
16. Rummery, G., & Niranjan, M. (1994). *On-line Q-learning using connectionist systems* (Technical Report CUED/F-INFENG/TR 166). Cambridge University Engineering Department.
17. O'Donoghue, B., Osband, I., Munos, R., & Mnih, V. (2018). *The uncertainty Bellman equation and exploration* (arXiv:1709.05380v4). arXiv.
18. Coutinho, R., & Boukerche, A. (2017). Opportunistic routing in underwater sensor networks: Potentials, challenges and guidelines. In *Proceedings of the 13th International Conference on Distributed Computing in Sensor Systems (DCOSS 2017)* (pp. 1–2). IEEE. https://doi.org/10.1109/DCOSS.2017.42
19. Darwish, A. (2018). Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications. *Future Computing and Informatics Journal, 3*(2), 231–246. https://doi.org/10.1016/j.fcij.2018.06.001

***Corresponding Author:** Dr. Shruthi K R

20. Beegum, T. R., Idris, M. Y. I., Ayub, M. N. B., & Shehadeh, H. A. (2023). Optimized routing of UAVs using bio-inspired algorithm in FANET: A systematic review. *IEEE Access, 11*, 15588–15622. https://doi.org/10.1109/ACCESS.2023.3244067
21. Saemi, B., & Goodarzian, F. (2024). Energy-efficient routing protocol for underwater wireless sensor networks using a hybrid metaheuristic algorithm. *Engineering Applications of Artificial Intelligence, 133*, 108132. https://doi.org/10.1016/j.engappai.2024.108132
22. Jha, A. V., Appasani, B., Khan, M. S., & Song, H. H. (2024). A novel clustering protocol for network lifetime maximization in underwater wireless sensor networks. *IEEE Transactions on Green Communications and Networking*. https://doi.org/10.1109/TGCN.2024.3375011
23. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). *OpenAI Gym* (arXiv:1606.01540). arXiv.
24. Shruthi, K. R., & Patil, S. B. (2021). An LSTM-based approach to predict stock price movement for IT sector companies. *International Journal of Cognitive Informatics and Natural Intelligence, 15*(4), 1–12. https://doi.org/10.4018/IJCINI.20211001.oa3

***Corresponding Author:** Dr. Shruthi K R