Reinforcement Learning based approach for Underwater Environment to evaluate Agent Algorithm

Shruthi K R (shruthikr.ads@bmsce.ac.in)
Visvesvaraya Technological University
Dr. Kavitha C
Visvesvaraya Technological University

Research Article

Keywords: Reinforcement Learning, Q Learning, Underwater Networks, Action Selection Methods

Posted Date: August 31st, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3291459/v1

Abstract

A lot of research is undergoing in Underwater as it has huge applications. An underwater network is a delay-tolerant network [1][2] due to its intermittent characteristics. Underwater acoustic communication enables communication undersea. Wireless sensor nodes underwater are sparsely placed due to environmental characteristics [3] to gather information. Communication undersea is tedious because of noise and varying environments. Since the underwater environment is highly unpredictable due to its nature, there doesn't exist a constant path or route between wireless sensor nodes. And the battery of sensor nodes is a major concern as they cannot be replaced frequently. Therefore, it's necessary to design an algorithm that can establish a path to the destination dynamically based on the environmental conditions and the node's battery level. In this paper, the authors have proposed a Reinforcement Learning approach to evaluate sensor nodes' performance. Many machine learning algorithms have used only the epsilon greedy action selection method. But here, four different types of action selection methods are used for the routing purpose. Based on the threshold level, an appropriate action selection method is chosen. The validation of the proposed approach is carried out by comparing the RL algorithm with other baseline algorithms. Experimental results showcase RL algorithm outperforms other baseline algorithms.

I. INTRODUCTION

Underwater is home to a lot of valuable species, minerals, oils, and many more. Tactical surveillance, weather monitoring, disaster management, and pollution monitoring is gaining more importance in monitoring the Underwater environment [2]. So, it is important to monitor the underwater environment. The underwater environment can be monitored by underwater sensor networks. Underwater sensor networks are considered delay-tolerant networks. These networks face enormous difficulties including transmission delay, multipath interference, heavy noise, fading, and harsh environmental conditions. Routing becomes a major challenge in these networks.

Communication in underwater sensor networks is achieved through acoustic communication. Amongst other communications, acoustic communication is very efficient in the transmission of data in underwater [5].

Machine Learning in particular the Reinforcement learning approach is ideal to resolve the challenges of underwater sensor networks. Reinforcement Learning is a kind of learning where the agent learns itself based on environmental conditions. RL tries to understand the environment and performs actions accordingly. This behavior of RL is appropriate to the networks which do not have a consistent environment. An underwater environment is one such environment that is inconsistent and intermittent [2]. This feature of adaptability helps us to solve many issues including routing [7]. RL algorithms have been used to address various issues and challenges in underwater sensor networks.

Reinforcement Learning is one of the prominent approaches for underwater acoustic networks and the most promising area for research [8], authors decided to contribute in this particular area. Let us understand the RL environment using a few terminologies, there is Agent ‘A’, state S, environment E, Action AK, and reward R. Let us now understand the working of the RL environment as shown in Fig. 1. Let's assume agent A is in state S, upon observing the environment agent A moves to state S + 1 by performing action AK and producing reward R. Here, R can be positive or negative based on the action. Now depending on the R, the agent changes its action accordingly. This strategy makes an agent find a route to the destination even in an unknown environment. RL-based learning takes sequential decisions dynamically. Many other approaches take sequential decisions including swarm particle optimization as a deep learning [9], game theory [10], and other probabilistic approaches [11]. However, all these approaches have their pros and cons.

Authors have chosen RL-based learning as it can be used in complex situations. The underwater environment being intermittent in nature, RL-based learning can be used to handle routing decisions in this precarious situation [1]. Since communication undersea is very essential [2], dynamic routing based on the environment becomes crucial. So, RL-based learning can suffice the requirement of dynamic routing undersea. The basic logic of RL includes an agent, a decision-maker, and the environment where the agent has to interact. At each iteration, the agent acts and moves to the next station by achieving rewards. Finally, RL tries to optimize its performance based on total rewards gained by agents. So, the RL environment for networking depends on the general backbone structure of the network which uses a partially observable Markov decision process. RL environment is described here by two attributes Observation space and Action space. Observation space defines the actual structure and the observed values for the state of observation. Consider an example where the agent has to learn to play a game. Here, observation space provides the vector of screenshots of the game played. Depending upon various environments, observation space provides appropriate values accordingly. On the other hand, action space defines the numerical structure of appropriate actions that can be taken by the agent in an appropriate environment. Consider an example shown in Fig. 2. Here the action space specifies the values of right and left directions along with angle specifications. These values help the robo agent as shown in Fig. 2 to perform an action, here specifying the direction the robo needs to take. The learned agent earns a reward based on the action that was taken and ends the episode once the task is completed.

The paper is divided into the following sections: Section II portrays a detailed survey of existing reinforcement learning approaches. Section III depicts an implementation of the AI Routing algorithm. Section IV contributes the results of the proposed algorithm using a customized environment. Section V concludes the summary of the paper.

II. RELATED SURVEY

Reinforcement Learning is very appropriate for dynamic environments as agents learn based on environmental conditions. This feature of RL helps to handle various tasks in networks. In this section, the use of RL in solving various tasks in respective networks has been presented.

Authors in [14] used the Q learning approach in finding the next forwarder by considering network topology. In this paper, authors have used the Q learning approach that aids in making global decision making of Next Forwarder. Here, the methodology uses cut vertex recognition to find appropriate NF and reduce energy wastage that occurs due to forwarding data packets to inappropriate nodes. The hierarchical level is also used in this method which defines the number of hops from the sink along the routing path. This approach has notable improvement in energy consumption, latency, and network lifetime.

A balanced routing protocol based on ML has been proposed by authors in [15] enumerates the pros of using the Q learning technique for lower network latency and reduced energy consumption. In this paper, authors consider environmental characteristics including latency, power limitations, and void area issues. Here, the protocol is divided into 4 phases – Initialization Phase, Discovery Phase, Clustering Phase, and Data Forwarding Phase. Q values are initialized in Initialization Phase. Routing tables are updated in Discovery Phase and Cluster head selection happens in Clustering Phase. Finally, reward calculations and updating of the q value happen in the Data Forwarding Phase. Authors have used the void area mechanism in this approach which finds out the area where there are no nodes or nodes that have drained their batteries. This methodology has good results in propagation delay and better channel response.

Authors in [16] have used RL for energy management in the network. Three approaches have been used in this paper to maximize network lifetime. One approach aims at reducing the length of routes using RL. The second approach tries to improve battery consumption by using sleep scheduling techniques. The third approach avoids unnecessary data transmission by the node depending on the data that is received. These three approaches have given promising results as compared to other approaches.

Reinforcement Learning with Particle Swarm Optimization [17] uses Particle Swarm Optimization, a bio-inspired algorithm that aims at providing optimal solutions in the solution space. In this paper, authors have used reward acting on RL with particle swarm optimization. Three strategies have been used in this paper. In this first step, a reward-based real-time rescue system is proposed using the particle swarm optimization technique. Next, a global optimal swarm is selected using R-RLPSO algorithm which eventually generates a path using the c_reward function. Then, using the linear reward function it calculates the rewards of all rescue areas. Thus, the RL-PSO algorithm is used to mark rescue areas and states in a cost-effective and using reduced amount of time.

An adaptive clustering routing protocol for underwater sensor networks is based on multi-agent reinforcement learning algorithms which is proposed by authors in [18]. This algorithm tries to minimize energy consumption by using adaptive cluster head selection. Here, cluster heads are selected without any communication and without any intimation from surrounding nodes. This avoids unnecessary communication between the nodes which in turn minimizes energy consumption. A biased reward function is used with an adaptive clustering algorithm to fetch feedback on routing phases. This feedback helps to select cluster heads as relays. This approach has shown higher routing efficiency, lower energy consumption, and longer network lifetime.

Authors in [19] have proposed a machine learning-based protocol for single-hop underwater sensor networks. As discussed earlier, transmission in underwater suffer from narrow bandwidth and end-to-end delay. Here, authors have designed various access schemes for uplink and downlink transmission depending upon their usage. Command Information with a low contention rate is delivered using a downlink scheme. Uplink transmission is designed as a hybrid scheme that delivers the information collected by sensor nodes with better network throughput. With these schemes, authors were able to achieve better throughput and end-to-end delay.

Reinforcement learning-based routing for underwater sensor networks using Steiner points has been proposed in [20]. Steiner points are used to minimize energy consumption and increase network lifetime. A Steiner tree is constructed from the sink node to another sensor to collect the information. This tree can contain Steiner points or isolated nodes. If it is an isolated node, the agent collects the data directly. If it is a Steiner point, the agent broadcasts beacon messages of its arrival. Then the anticipated group of sensor nodes communicates with the agent and forwards the sensed data. This strategy enhances network lifetime and reduces the energy consumed by the nodes.

Algorithms discussed in Table 1 address various issues in Underwater sensor networks. Various Reinforcement Learning approaches are used to deal with some of the problems faced mainly in Underwater Sensor Networks.

TABLE I

SURVEY OF REINFORCEMENT LEARNING ALGORITHMS USED FOR UNDERWATER SENSOR NETWORKS

Authors	Protocol Used	Problems Handled
14	Topology aware Q learning	Low energy consumption, shorter latency, longer network lifetime
15	Balanced Routing Protocol – ML Approach	Low signal propagation delay, Predictable channel response
16	Reinforcement Learning based Energy efficient control	Maximizes network lifetime
17	Reinforcement Learning and Particle Swarm Optimization	Cost-effective and Time saving
18	Adaptive clustering routing protocol	Higher routing efficiency, lower energy consumption, and longer network lifetime
19	Machine Learning based performance efficient MAC protocol	Network throughput, End to end delay, Transmission fairness
20	Reinforcement learning-based routing	Reduce energy consumption, Enhance network lifetime

In this paper, a reinforcement learning approach proposed by authors in [1] is implemented. The authors have created an underwater environment suitable to the problem statement.

III. IMPLEMENTATION

As described in sections I and II, Reinforcement learning is most suitable for routing in Underwater Sensor Networks. Here, the authors have used Q learning to perform to find a path from source to destination. Let us discuss the implementation of Q learning for underwater networks

A. Q Learning

Q learning is one of the popular Reinforcement learning algorithms that enables an agent to iteratively learn from the environment and improves its learning over time by choosing optimal action. It is an iterative process that involves agent learning by exploring the environment which in turn updates the model as the exploration continues.

Components of Q learning include Agent, Action, State and Rewards, and Q values.

Agent – is a learning entity that takes optimal actions and operates in a specified environment

State – The current position of an agent in the specified environment is labeled as the state of the agent

Actions – Any operation taken by an agent when it is in a specified state.

Rewards – A positive or negative response given to an agent based on its action.

Q values – It is the metric used to measure an action at a particular state.

Some of the ways of finding this Q value are -

Temporal Difference – It calculates the Q value by incorporating the value of the current state and action by comparing the differences with the previous state and action.

Bellman’s Equation – It calculates the Q value of a given state and assesses its relative position. The state with the highest value is considered as an optimal state. Bellman’s equation is represented as in (1).

$Q\left(s,a\right)=Q\left(s,a\right)+\propto\left(r+\gamma(\max\left(Q\left(s',a'\right)\right)-Q\left(s,a\right)\right)$ (1) [22]

Q(s,a) → represents the expected reward for taking action a in the state s

r → represents the actual reward received for any particular action

s’ → represents the next state

α → represents the learning rate

γ → represents a discount factor

max(Q(s’, a’) → is the highest expected reward for all possible actions a’ in state s’

In this paper, the authors have used Bellman’s equation to find the Q value. These Q values are stored in a Q table which contains rows and columns with the list of rewards for the best action of each state in a specified environment.

Table 2
Q TABLE

State/Action	**A1**	**A2**	**A3**
S1			
S2			
S3			

Here, the rows represent different situations the agent might encounter and the columns portray the actions. Agents would look up the Q table to fetch expected future rewards for any given state and action pair.

B. AI Routing

An underwater environment is simulated with an adhoc number of sensors. An AI-based routing is designed to route the packets from the source sensor to the destination sensor. Here, the state is represented either as position precision or as sensor path. Algorithm 1 represents AI-based routing used in this paper.

Algorithm 1

```
AI Based Routing
2. Initialize state, action, reward and Q-table
3. Initialize $\alpha$, $\gamma$, $\varepsilon$ values
4. Choose action based on $\varepsilon$ greedy method:
Update Q table using (1)
Calculate reward:
If outcome == -1
Return – 2
Else
Return 2*(1-delay/event-duration)
Next action is selected based on the highest score in the table
Repeat step 4 for various episodes
```

Sensors can take random actions to forward the packets to the next sensor. These random actions are selected based on ε greedy selection or geo-location-based routing which are described in section C. Upon selecting random action, the Q value and reward are calculated as specified in Algorithm (1), and the Q table is updated accordingly. Once the Q table is updated for various episodes, the sensor now chooses the action which is having highest value in the Q table. These appropriate actions fetch an optimal path to depot the packet to the destination sensor.

C. Action Selection Method

In this section, let us discuss various action selection methods used in AI Routing algorithms.

Epsilon Greedy Action Selection Method – is a method to balance between exploration and exploitation randomly. Here, ε value refers to the probability of choosing exploitation or exploration. Generally, the ε value is initialized to a smaller value, giving a lesser chance of exploration.

Action taken at time(t) = $\begin{cases} \max Qt\left(a\right) withprobability1-\epsilon \\ anyaction\left(a\right) withprobablity\epsilon \end{cases}$

(2)

Geo Location Based Routing – is a geographical approach that returns the sensor closest to the requestor as shown in (3). In this approach, the nearest sensor is chosen based on the geographical positions of the sensors. The position of the nearest sensor is calculated using the acknowledgment of the hello packet.

Geo location and Epsilon greedy routing are again based on two important parameters next target and position precision. Next target finds the nearest sensor using a dictionary containing earlier hello messages. This method of search has a threshold. Once, the threshold has been reached, it switches to position precision where the nearest neighbors are again calculated based on hello packet response times.

In this paper, authors have implemented epsilon greedy and geo location routing with next target and position precision parameters. The results are presented in the next section.

IV. RESULTS

To evaluate the performance of the proposed methodology, an AI routing algorithm is implemented using epsilon greedy and geo location-based random selection methods. In the initial simulation, the AI routing algorithm is evaluated against other baseline algorithms including Geo routing, Random routing, and closest-to-me routing. Simulation parameters portrayed in Table 3 depict parameters used for the simulation purpose.

Table 3
SIMULATION PARAMETERS

Parameters	Value
Discount Factor, γ	0.4
Learning Rate, α	0.2
Steps of Simulation	3000000
Seconds of step duration	0.15 sec
Environment Width * Height	1500m*1500m
Communication Range	200 m
Sensor speed	8m/s
Sensor Buffer size	100

Geo routing is a geographical approach that takes the sensor closest to the destination. The selection of a sensor happens by calculating the coordinates of the sensor. And then, Euclidean distance is used to calculate the sensor nearest to the destination. Random routing is a routing approach that randomly selects a neighbor to forward the packet to. In this approach, a random function is applied to the neighbors of the source. The source forwards the packet to the randomly selected sensor. Closest to me routing selects a neighbor that is nearest to the source. Neighbors of a sensor are chosen based on a hello message response. Then Euclidean distance is applied to find the sensor nearest to the source. These approaches are evaluated against AI routing which uses a Q table to select the optimal path to the destination. Table 4 illustrates the delivery ratio of various algorithms.

Table 4
DELIVERY RATIO OF VARIOUS ALGORITHMS

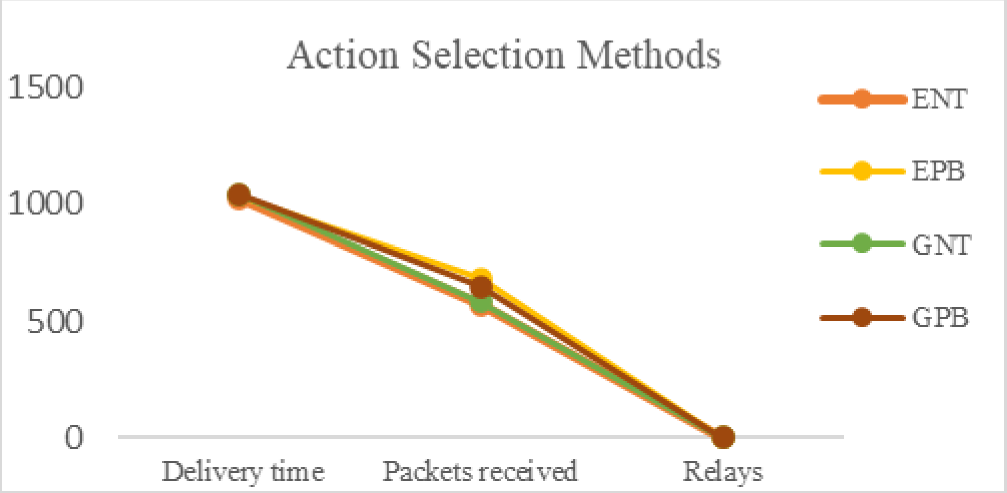
Algorithm	Delivery Ratio
Geo Routing	54.49
Random Routing	60.82
Closest to me Routing	60.82
AI Routing	70.82

As shown in the Table 4, AI routing which uses reward-based calculation outperforms other algorithms by delivering more packets in lesser time. The (4) depicts the average rewards the AI routing algorithm gains for different no. of episodes (X-axis). The graph conveys that the average no. of rewards doesn’t vary much as the episodes increase. This indicates that the agent would have learned better in the course duration. This results in less variation of rewards gained.

The AI routing algorithm is also executed using various action selection methods with two neighbor selection parameters as described in section III. Let us analyze the same with other network characteristics. The following graphs illustrate the average rewards received for the random action selection methods described in the previous section.

Table 5
ANALYSIS OF VARIOUS ACTION SELECTION METHODS

Action Selection Method	Packet mean delivery time	No. of packets to Depot	The mean number of relays
Epsilon Next Target Routing (ENT)	1027.29	571	1.28
Epsilon Position Based Routing (EPB)	1038.81	684	1.32
Geo Next Target Routing (GNT)	1044.04	586	1.27
Geo Position Based Routing (GPB)	1043.02	648	1.30



As seen in Table 5, Epsilon Next Target Routing delivers the packet faster than compared to other approaches. But, a greater number of packets deported to the destination is given by epsilon position-based routing. Since no. of packets delivered to the destination and delivery time is important for any routing approach, there should be a balance between these two parameters. Therefore, Epsilon Position based routing has a perfect balance of both the parameters and can be considered as an optimal approach. The following graphs depict the average rewards earned by different approaches. All the approaches have gained similar no. of rewards. The only difference between them can be seen in Table 5.

One of the main criteria in any communication network is packet mean delivery time. The following table and graphs illustrate how other factors depend on several sensors.

Table 6
NO. OF SENSORS v/s OTHER FACTORS

No. of sensors	Delivery Ratio	Packet mean delivery time	No. of packets reached
5	70.82	1038.81	684
10	60.55	1001.96	529
15	61.31	1011.60	491
20	58.02	952.39	478
25	56.79	1002.66	305

As observed in Table 6 and Fig. 9, the delivery ratio and the number of packets sent to the destination have decreased as the number of sensors is increased. This is because an increased number of sensors can cover a larger geographical area and the path and number of hops from source to destination can be longer than compared to the lesser number of sensors. But packet mean delivery time has reduced as the number of sensors increased. This is because the average time required to deliver the packets takes less time as there would be a greater number of options (sensors) available to send the data.

It is good to observe the analysis of all algorithms in a single plot. Figures 10 and 11 illustrate the parameter-based analysis of all algorithms. Simulation parameters mentioned in Table 3 are retained for analysis purposes.

As observed in Fig. 10 and discussed earlier in this section, AI-based algorithms demonstrate a perfect balance between the delivery ratio and the number of sensors. The delivery ratio of the even number of sensors is less than compared to an odd number of sensors. This might be because several relays and paths formed for an odd number of sensors are better than compared to an even number of sensors.

Events refer to something happening at a particular point in time. Upon sensor sensing an event, the sensor has to deport the same to the destination. Figure 11 illustrates the delivery time of events of the algorithms described above. So, here AI-based algorithms are again performing better than compared to other algorithms.

V. Conclusion

Routing in underwater sensor networks is very crucial because of its noisy nature [21]. The authors here have designed an AI-based routing that understands the environment by using an exploration and exploitation strategy to perform routing. The results as described in section IV illustrates AI based routing algorithms outperform other baseline algorithms considered here. In the AI-based routing algorithm, Bellman’s equation [22] and other action selection methods are used to construct a Q table. This method of table construction and neighbor selection has shown better performance with a better delivery ratio, packet mean delivery time, and event mean delivery time. So, we conclude here that AI routing is more suitable for underwater environments.

The research can be further extended by applying a neural network approach in the construction of a Q table. As neural networks have proved that they establish the relationship between the parameters very well, this analogy can be applied in the construction of the Q table.

Declarations

Ethical Approval – Our research doesn't include any plant, animal, or human study. Therefore, ethical approval is not applicable to our study.

Competing interests - The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Authors' contributions – The authors confirm their contribution to the paper as follows:

Title and Abstract: Shruthi K R; Introduction and Related Survey: Dr. Kavitha C, Shruthi K R; Implementation: Shruthi K R, Results: Shruthi K R, Dr. Kavitha C; Conclusion: Shruthi K R. All the authors have reviewed the results and have confirmed the final version of the manuscript.

Funding – No funding was received to assist in the preparation of this manuscript.

Data Availability – The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

References

1. Shruthi, K. R., & Dr. Kavitha, C., "Reinforcement learning-based approach for establishing energy-efficient routes in underwater sensor networks", 8th International Conference on Electronics, Computing and Communication Technologies, IEEE CONECCT-2022.
2. Shruthi, K. R., "An Artificial Intelligence Based Routing for Underwater Wireless Sensor Networks," 4th International Conference on Electrical, Electronics, Communication, Computer Technologies, and Optimization Techniques.
3. Qin, Q., Tian, Y., & Wang, X. (2021). "Three-Dimensional UWSN Positioning Algorithm Based on Modified RSSI Values", Mobile Information Systems, Volume Article ID 5554791, 8 pages https://doi.org/10.1155/2021/5554791.
4. Bouk, S. H., Ahmed, S. H., & Kim, D. (2016). "Delay Tolerance in Underwater Wireless Communications: A Routing Perspective", Mobile Information Systems, vol. Article ID 6574697, 9 pages, 2016. https://doi.org/10.1155/2016/6574697.
5. Jiang, S. (2018). State-of-the-art Medium Access Control (MAC) protocols for underwater acoustic networks: a survey based on a mac reference model. *Ieee Communication Surveys And Tutorials*, *20*(1), 96–131.

6. Hu, T., & Fei, Y. (2010). "An Adaptive and Energy-efficient Routing Protocol Based on Machine Learning for Underwater Delay Tolerant Networks," 2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, pp. 381–384, 10.1109/MASCOTS.2010.45.
7. Zhang, W., Li, J., Wan, Y., et al. (2022). Machine Learning-Based Performance-Efficient MAC Protocol for Single Hop Underwater Acoustic Sensor Networks. *J Grid Computing*, *20*, 41. https://doi.org/10.1007/s10723-022-09636-9.
8. Varshini Vidyadhar, Nagaraj, R., & Ashoka, D. V. (2021). NetAI-Gym: Customized Environment for Network to Evaluate Agent Algorithm using Reinforcement Learning in Open-AI Gym Platform. *International Journal of Advanced Computer Science and Applications(IJACSA)*, *12*(4), http://dx.doi.org/10.14569/IJACSA.2021.0120423.
9. Hüttenrauch, M., Adrian, S., & Neumann, G. (2018). Deep reinforcement learning for swarm systems. *Journal of Machine Learning Research*, *20*(54), 1–31.
10. Sun, C., & Duan, H. (2015). Markov decision evolutionary game theoretic learning for cooperative sensing of unmanned aerial vehicles. *Sci China Technol*, *58*, 1392–1400.
11. Ghoul, R., He, J., Djaidja, S., Al-qaness, M. A. A., & Kim, S. (2020). “PDTR: Probabilistic and Deterministic Tree-based Routing for Wireless Sensor Networks”, Sensors, 20(6), pp.1697.
12. Wang, H., Liu, N., & Zhang, Y. (2020). Deep reinforcement learning: a survey. *Front Inform Technol Electron Eng*, *21*, 1726–1744.
13. Greg Brockman, V., Cheung, L., Pettersson, J., Schneider, J., & Schulman, Jie Tang and Wojciech Zaremba, “OpenAI Gym”, arXiv:1606.01540, 2016.
14. Nandyala, C. S., Kim, H. W., & Cho, H. S., QTAR: A Q-learning-based topology-aware routing protocol for underwater wireless sensor networks, *Computer Networks*, Volume 222, 2023, 109562, ISSN 1389 – 1286, https://doi.org/10.1016/j.comnet.2023.109562.
15. Alsalman, L., & Alotaibi, E. (2021). "A Balanced Routing Protocol Based on Machine Learning for Underwater Sensor Networks," in IEEE Access, vol. 9, pp. 152082–152097, 10.1109/ACCESS.2021.3126107.
16. Abadi, A. F. E., Asghari, S. A., Marvasti, M. B., Abaei, G., Nabavi, M., & Savaria, Y. (2022). "RLBEEP: Reinforcement-Learning-Based Energy Efficient Control and Routing Protocol for Wireless Sensor Networks," in *IEEE Access*, vol. 10, pp. 44123–44135, 10.1109/ACCESS.2022.3167058.
17. Wu, J., Song, C., Ma, J., Wu, J., & Han, G. (July 2022). Reinforcement Learning and Particle Swarm Optimization Supporting Real-Time Rescue Assignments for Multiple Autonomous Underwater Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *23*(7), 6807–6820. 10.1109/TITS.2021.3062500.
18. Yao, S., Zheng, M., Han, X., Li, S., & Yin, J. (2022). Adaptive clustering routing protocol for underwater sensor networks. *Ad Hoc Networks*, *136*, 1570–8705. https://doi.org/10.1016/j.adhoc.2022.102953.
19. Zhang, W., Li, J., Wan, Y., et al. (2022). Machine Learning-Based Performance-Efficient MAC Protocol for Single Hop Underwater Acoustic Sensor Networks. *J Grid Computing*, *20*, 41.

https://doi.org/10.1007/s10723-022-09636-9.

20. Halakarnimath, B. S., & Sutagundar, A. V. (2021). Reinforcement Learning-Based Routing in Underwater Acoustic Sensor Networks. *Wireless Personal Communications*, *120*, 419–446. https://doi.org/10.1007/s11277-021-08467-3.
21. Coutinho, R., & Boukerche, A. (2017 pp). "Opportunistic Routing in Underwater Sensor Networks: Potentials, Challenges and Guidelines," in 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS), Ottawa, ON, Canada, 1–2. 10.1109/DCOSS.2017.42.
22. Brendan, O. D., Osband, I., Munos, R., & Mnih, V., The Uncertainty Bellman equation and exploration, arXiv:1709.05380v4 [cs.AI] 22 Oct.

Figures

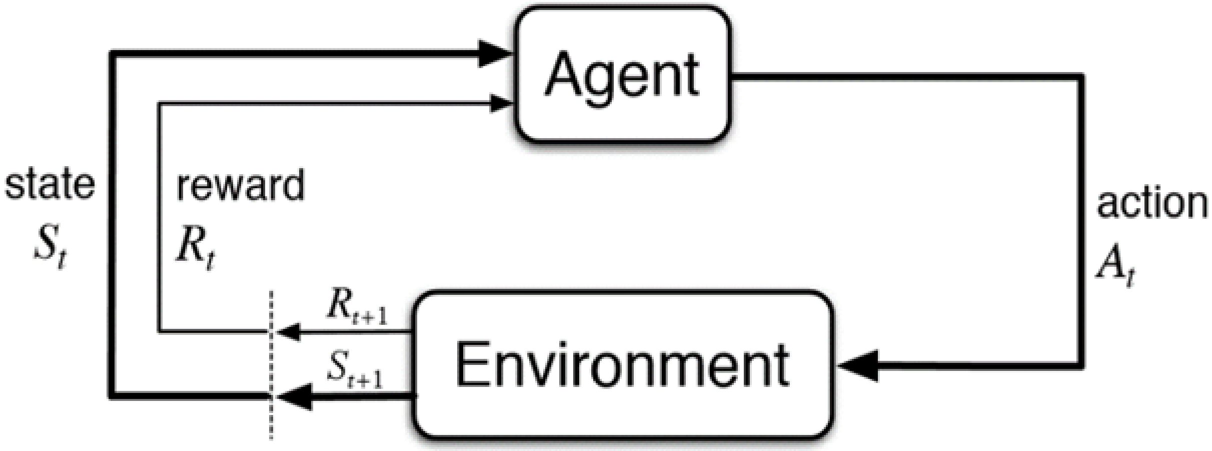


Figure 1

Reinforcement Learning

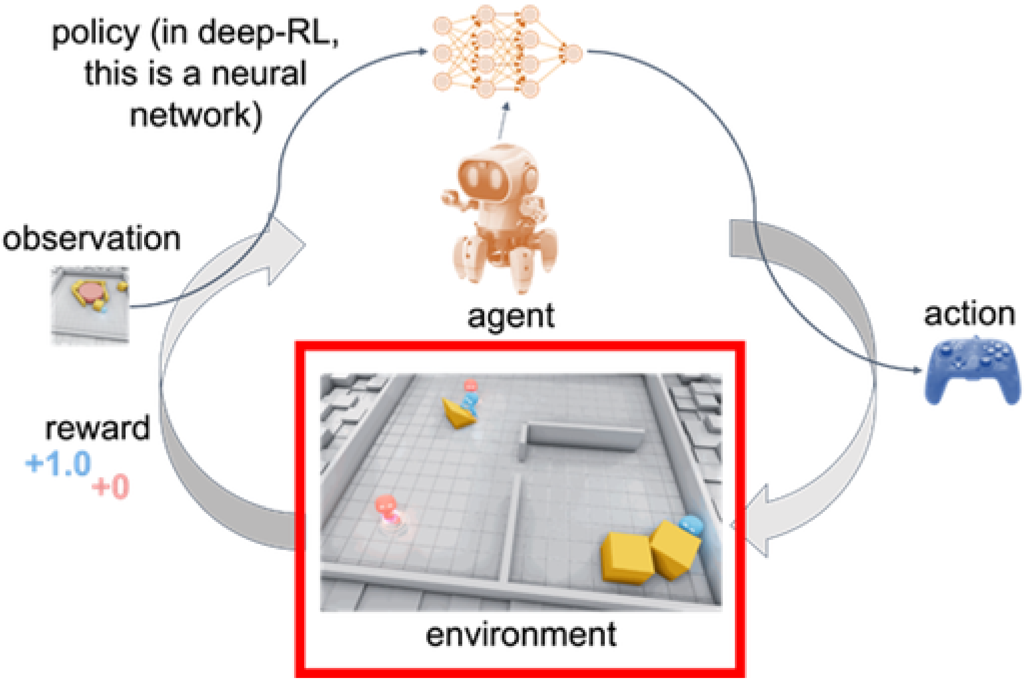


Figure 2

Robo finding a path to the destination

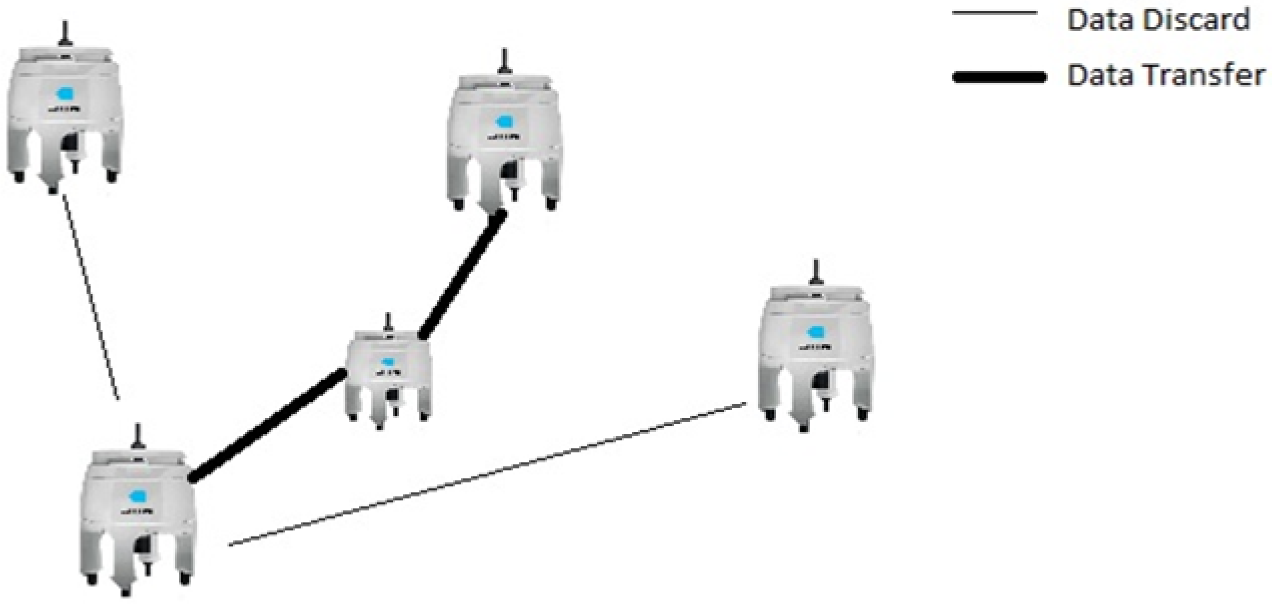


Figure 3

Geo Location Based Routing

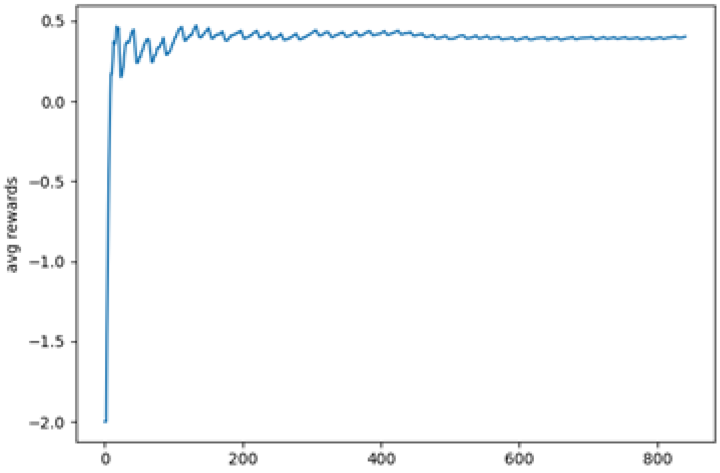


Figure 4

Average rewards gained by AI routing algorithm

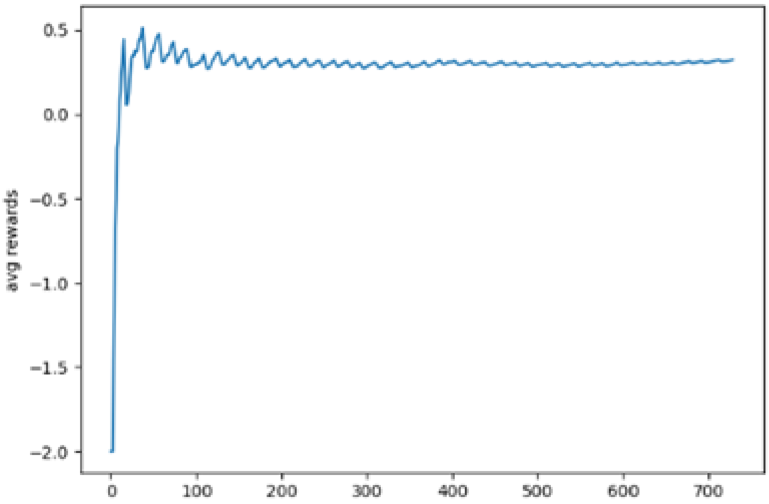


Figure 5

Average rewards gained by AI using Epsilon next target method

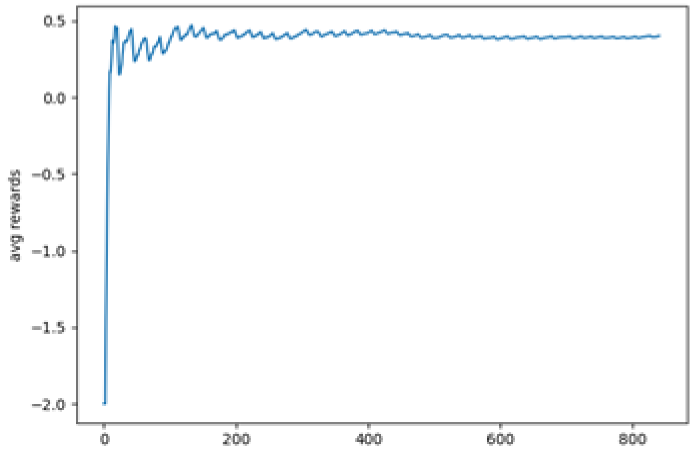


Figure 6

Average rewards gained by AI using Epsilon position precision

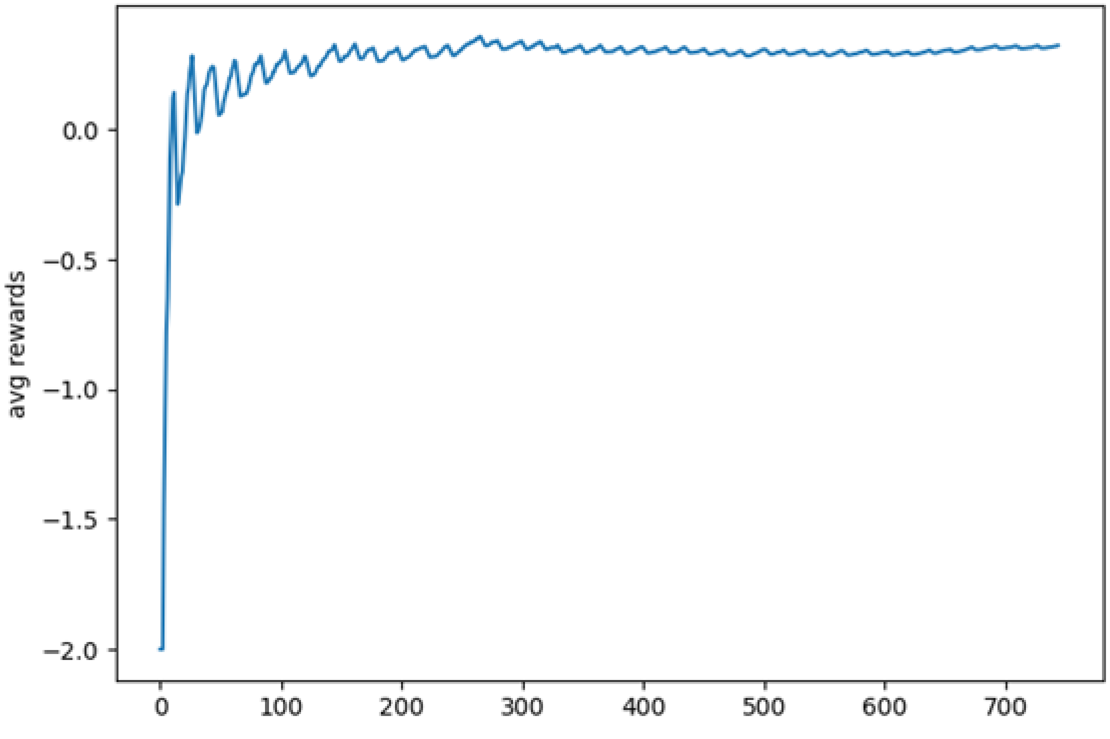


Figure 7

Average rewards gained by AI using Geo location next target

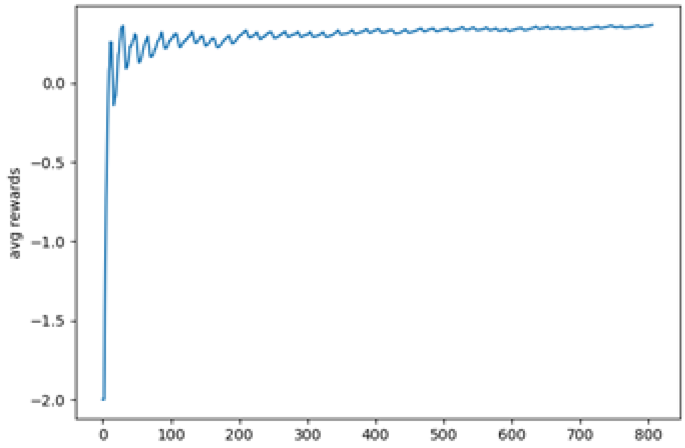


Figure 8

Average rewards gained by AI using Geo location position precision

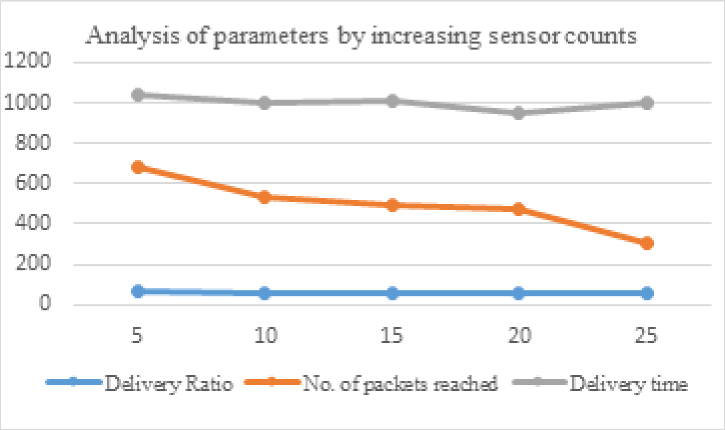


Figure 9

Analysis of parameters by increasing sensor counts

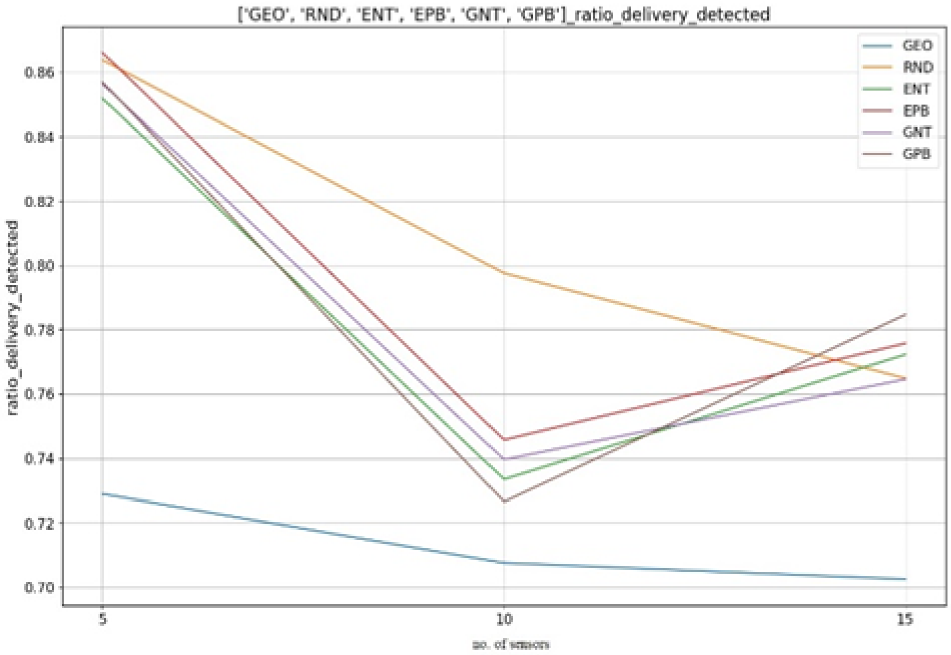


Figure 10

Analysis of algorithms for delivery ratio

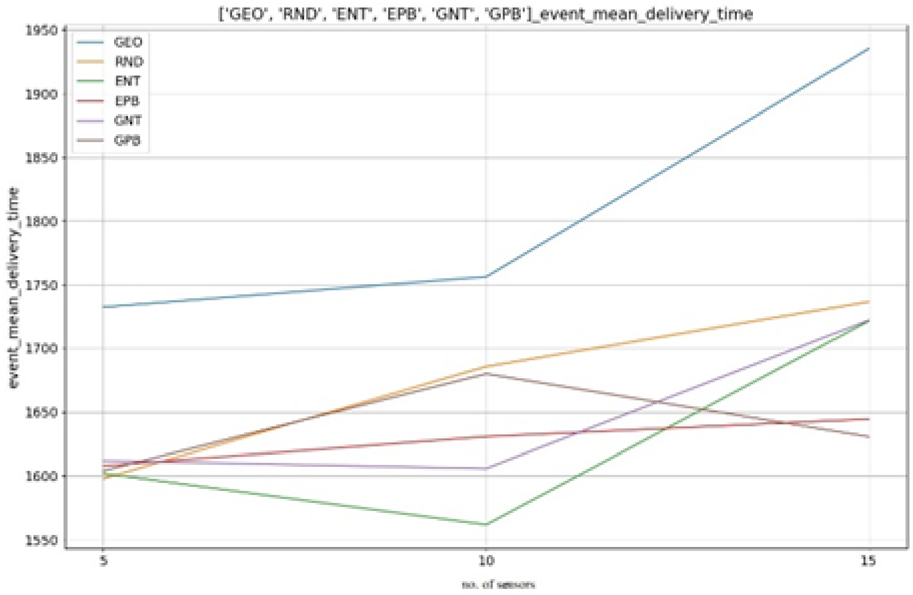


Figure 11

Analysis of algorithms for event mean delivery time