Evaluation of Agent-Network Environment Mapping on Open-AI Gym for Q-Routing Algorithm

Varshini Vidyadhar[1]
Research Scholar
Department of Computer Science & Engineering
Bangalore Institute of Technology, Bengaluru, India

Dr. R. Nagaraja
Professor
Department of Information Science and Engineering
Bangalore Institute of Technology, Bengaluru, India

***Abstract*—The changes in network dynamics demands a routing algorithm that adapts intelligently with the changing requirements and parameters. In this regard, an efficient routing mechanism plays an essential role in supporting such requirements of dynamic and QoS-aware network services. This paper has introduced a self-learning intelligent approach to route selection in the network. A Q-Routing approach is designed based on a reinforcement learning algorithm to provide reliable and stable packet transmission for different network services with minimal delay and low routing overhead. The novelty of the proposed work is that a new customized environment for the network, namely Net-AI-Gym, has been integrated into Open-AI Gym. Besides, the proposed Q-routing with Net-AI-Gym offers optimization in exploring the path to support multi-QoS aware services in the different networking applications. The performance assessment of the NET-AI Gym is carried out with less, medium, and a high number of nodes. Also, the results of the proposed system are compared with the existing rule-based method. The study outcome shows the Net-AI-Gym's potential that effectively supports the varied scale of nodes in the network. Apart from this, the proposed Q-routing approach outperforms the rule-based routing technique regarding episodes vs. Rewards and path length.**

***Keywords*—Reinforcement learning; environment; agent; network; Net-AI-Gym; Q-routing; rule-based routing**

I. Introduction

The collaboration of entities either in the form of computing devices or the people or be it any things through some specific form of connectivity and set of communication protocols forms a network [1]. Examples of networks may include computer networks [2], social networks [3], the network of things as the Internet of Things (IoT) [4]. The adoption of machine learning is a requirement to bring automation in the process of routing.

A. Machine Learning Models

The machine learning models learn to perform a specified task(T). The machine learning models (MLM) are broadly classified into three categories as i) Supervised learning Model (SLM), ii) Unsupervised Learning Model (USLM), and iii) Reinforcement Learning (RL) model, as shown in the Fig. 1. The selection of the MLM depends upon the type of task(T) to be performed by the machine. Whereas the learning experiences (LE) in different MLM comes from the different sources of the data. In the SML, the 'LE' comes from the input and output mapping of the data. The USML gains the 'LE' from the pattern of the data.

B. Reinforcement Learning

The RL Model is a goal-oriented ML approach, where the 'LE' for performing a 'T' comes by interacting with the uncertain and dynamic environment. The RL enables the computer to make a sequence of such decisions that ensure to maximize their cumulative rewards (CR) automatically even if the computer is not explicitly programmed to complete the 'T.' Fig. 2 illustrates the architectural diagram of the typical RL context.

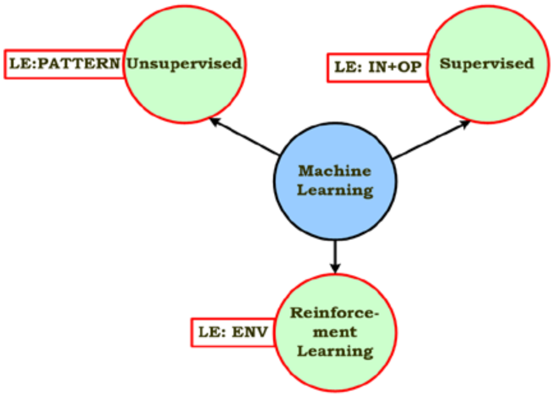

Fig. 1. Classification of Machine Learning Models

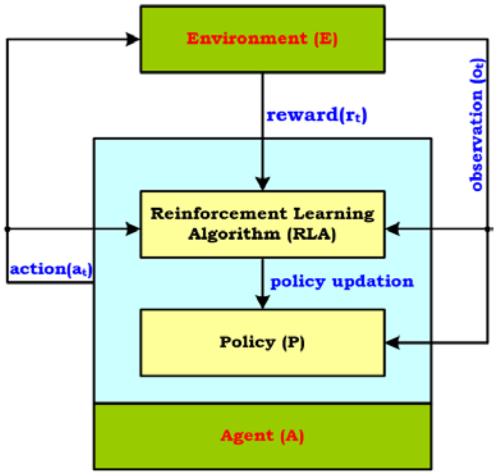

Fig. 2. Architectural Diagram of Typical RL Context.

The 'E' in RL is partially observable Markov decision process (POMDP). The mathematical modeling of any problem as an RL model takes a set of attributes (Sa) as a core building component for building the intuition of $\forall E \in S$. The attribute set Sa={ AE-R, ,MP , ,MP-MC , MRP ,MDP, , ,BE}, where, AE-R= agent and environment relationship , MP=Markov process , MP-MC= Markov process and Markov chain , MRP= Markov rewards process, MDP = ‘Markov decision process, and finally, BE=Bellman equation.

C. Agent-Environment Relation (AE-R)

Basically, in RL Model, the 'A' is a software component that takes intelligent decisions based on the learning by an iterative cycle of gaining reward and penalty. The 'E' mimics the representation of the problem through a simulated environment to which the 'A' interacts. Another construct, namely 'state' (S), is the 'A' at a definite time step in the 'E.' The process that cannot be changed arbitrarily or randomly is a part of the 'E.' In simplification, action is any decision which the model wants it to learn, and the State is useful in choosing action. The 'A' cannot change the rewards arbitrarily. However, while designing the 'A,' it is assumed that as a part of its functions of taking action under the particular State, the agent knows how the computation of reward takes place in the environment. In some context, though the 'A' may have complete exposure or awareness of the 'E' yet, the 'A' finds it hard to maximize the 'R.' Therefore, the AE-R is the representation of the boundary or the limit of the 'A' control, not the knowledge of the 'A.'

D. Markov Property (MP)

The two intrinsic properties that define the Markov property are i) Transition (T): the process of changing one State to another state, ii) Transition probability (TP): The probability by which the agent can move from one State to another. Therefore, the Markov property states that "Future is independent of the past if present is given." The MP is expressed as in (1).

$$P[S_{t+1} \mid S_t]=P[S_{t+1}|S_1,\ldots,S_t] \quad (1)$$

Where, S[t] =current state of ‘A’ and S[t+1] = next state. The intuitively meaning is that the current state includes information of the past states. Whereas, the state transition probability (STP) is expressed as in (2).

$$STP = P_{s \rightarrow s'} = P\ [S_{t+1\,=}s' \mid S_t =s] \quad (2)$$

The STP formulates a STP matrix (STPM) where, ∀ Row ∈ STPM, represents the probability of moving next state.

$$\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}$$

The sum of each row ∈ STPM, $\sum R = 1$.

E. Markov Process and Markov Chain (MP-MC)

MP is a memoryless random process such that the sequence of the random states {S[1], S[2],…S[n]} with a Markov property, so the environment, E:{State(S), STPM}.

F. Markov Reward Process (MRP)

To understand the MRP, it is essential to understand the rewards concept and the different nature of the task. The agents receive a +ve or a -ve numerical value on acting as some state(S) in the 'E,' whereas the sum of such rewards is called return(G) which is expressed in the equation below as in (3):

$$G_t = \sum_{i=t+1}^{T} r_i \quad (3)$$

Another essential aspect is the type of task. It is either episodic or continuous. In the episodic task, every time the process initiates from the start-state (S_s) and ends at the terminating-state (S_t), and once it researches the $S_{t,}$ it is said to be completing one episode (E_t). Again, the process restarts from the S_s to S_t, where every E_t is independent of others. Once the agent research the destination node from the source node in the routing case, it completes one E_t. In contrast to the episodic task, the continuous task (C_t) does not have any terminating state (S_t). Therefore, the return (G) can be easily calculated, whereas the value of 'G' in Ct yields infinity. Thus, to resolve the problem of infinity return (G), the concept of discount

factor (γ) comes into the picture.

The factor 'γ' basically regulates the immediate reward (Ri) and the future rewards (Rf) and the range of 'γ': {0,1}to avoid $R\infty$ in the Ct. The value '0' of 'γ' indicates higher importance to the Ri, whereas the value '1' indicates higher importance. Therefore γ = 0, in practice no learning condition and γ = 1, will keep hunting infinitely the Rf, so for all practical purpose 'γ' is taken as: 0.2≤γ≤0.8. Therefore, the equation (3) with discount factor(γ) is normalized as in equation (4).

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (4)$$

The Agent (A) operates in the environment and gets rewards for each operation, and the prime goal of the 'A' is to maximize the cumulative reward by taking the most suitable action for the current observation. Therefore, the success rate of the 'A' is decided based on its reward, which shows how well the action being taken. In the existing literature, various research works have been conducted based on RL to solve network-related problems. However, none of the existing research works have introduced a suitable networking environment to evaluate the RL agent. In this paper, the evaluation of a customized environment, namely Net-AI Gym, is carried out with an RL agent algorithm designed based on the Q-Learning approach for network routing. The proposed study also considers a rule-based algorithm to be evaluated in the Net-AI Gym environment to carry out performance assessment in terms of reward VS episode and path length. The remaining sections of this paper are organized as follows: Section II provides a brief discussion on the types of networks and their characteristics. Section III presents the formulation of routing problems in the network. Section IV presents related work for analyzing the existing literature regarding the application of RL to network routing. Section V discusses the Environment Setup and processes involved. Section VI discusses the performance analysis and model validation, and finally, the overall contribution of the proposed work is concluded in Section VII.

II. Types of Network And Characteristics

Networks have become progressively ubiquitous. The network is a system of interconnected devices intended to share digital information. According to the characteristics, communication networks are usually divided into different categories: wired or wireless, energy limitation, network topology, and node mobility. These characteristics have a significant influence on RL-based routing optimization. Fig. 3 highlights the different types of networks based on wired and wireless network types.

In general, the typical characteristics of the network that usually affect the protocol design are based on several factors such as network is infrastructure-based or no infrastructure-based, centralized or distributed, node mobility, variation in network topology, node's energy consumption, link quality, bandwidth, the accuracy of data transmission, and application-specific requirements such as, time sensitivity and reliability-based requirements. The RL-based routing protocol has been gradually extended and improved as the network develops. Thus, l RL-based routing protocols have the capability of addressing various network issues.

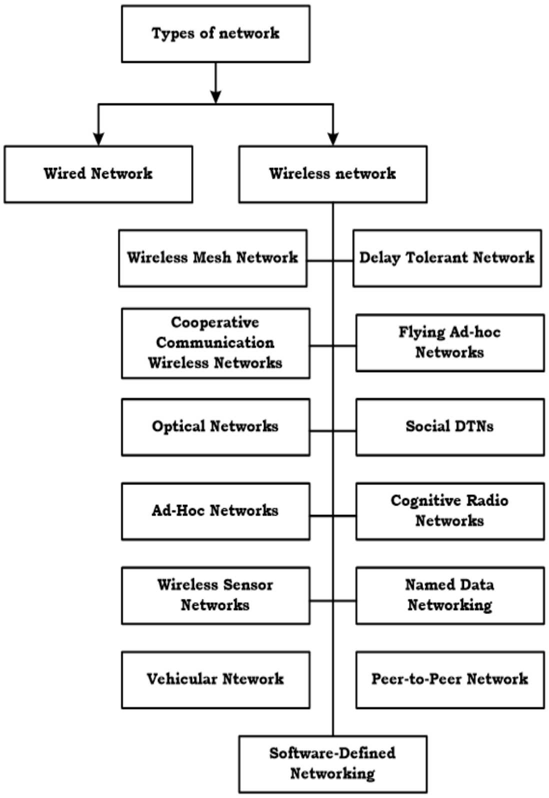

Fig. 3. Types of Networks.

III. Formulation of Routing Problem

The typical network 'NW' is a set of nodes(N), whereas a typical graph(G) is a set of vertices(V). In the case of NW, the 'N' is connected by links(L), whereas in the case of G, 'V' is connected by edges(E). Therefore, the mapping of a set: Network (NW) ={Node(N), Link(L)}→ Graph(G) ={Vertex(V), Edge(E)}, so NW (N, L) is mapped to G (V, E). Whereas the Route(R) in an NW and a Path (P) in G is defined as a way to move from the source node (Ns) to the destination node (Nd) in the NW and origin vertex (Vi) to another vertex (Vj), so {R (NW): Ns→Nd}→{P(G): Vi→Vj}. In the Network (NW), the link(L) is either one way or two ways, i.e., either Vi→Vj or Vj→ Vi. Fig. 4 illustrates a real-world network as a mathematical model of an NW.

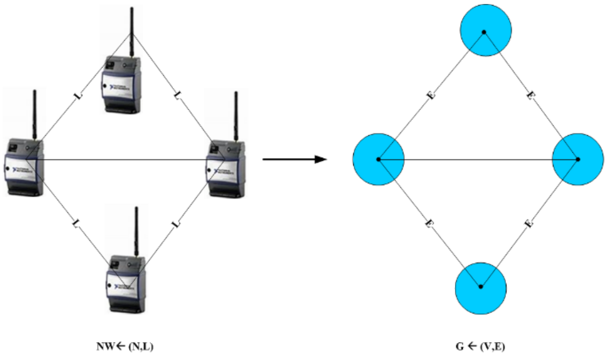

Fig. 4. Mapping Real-world Network (NW) to a Mathematical Model as Graph(G) of Machine Learning Models.

Therefore, in a nutshell, the network routing problem (NRP) is to obtain an optimal path (Po) between Vs. and Vd through intermediate nodes (Vk) under the constraints of weight (Wij) assigned to the link between Vi and Vj. The numerical value of Wij depends upon a set of parameters (Ps) with explicit and implicit relation between various properties of the chosen network such that Ps={Time(T), Cost (C), distance(D), Energy(E), Bandwidth (BW), Signal Strength (SS)}. Generally, in any network delay is induced or experienced due to the traffic condition of congestion, along with it if the nodes and links fail, the network topology dynamically changes; therefore, finding Po in such changing dynamics is a computationally expensive task because the complexity O(N2) of computation is exponential with the number of nodes(N/V).

IV. Review of Literature

This section discusses the existing literature regarding the application of RL to network routing. A recent work carried out by Guo et al. [5] developed a quality-of-service (QoS) aware secure routing technique in software-defined Internet of Things (SD-IOT). This scheme exploits SDN's exclusive characteristics and considers the routing path planning under malevolent attacks to achieve QoS. A few research literature on underwater sensor network (UWSN) specific RL-based routing congestion control techniques [6]. In this approach, the next forwarder nodes are selected based on the current buffer state, location, and remaining energy of the one-hop neighbor nodes. The adoption of RL based routing scheme is considered for magnetic induction communication in UWSN [7]. In this study, the authors have derived the iterative formula of the Q-table by considering distance and energy path metrics. The RL-based routing scheme to the Mobile Ad-hoc Network (MANET) is considered by [8]. A routing algorithm is designed to select the next hop for packet forwarding based on the joint mechanism of stochastic approximation, function approximation, and RL. The authors in the study of [9] used RL's Q-learning approach to design an efficient and enhanced gradient-oriented routing strategy for balancing energy consumption and QoS in Wireless Sensor Network (WSN). RL-based opportunistic routing scheme is suggested by [10] to support high-dimension data streaming in the application of multi-hop wireless networks. The application of Q-learning for Unstructured Peer-to-Peer Networks is considered by [11]. A load-balancing intelligent routing is devised to improvise query search efficiency for both intragroup and intergroup peers and reduce loads of queries under higher churns and heavy network workloads. A concept of a multi-agent RL scheme is introduced by [12] to devise an optimal routing scheme for the underwater optical WSN. In this approach, nodes' link quality and remnant energy are considered to develop an efficient routing algorithm suitable to dynamic communication environment and prolong network service duration. Also, Q-value initialization and the variant learning rate are devised to boost the routing algorithm's convergence. Some existing studies have implemented an RL-based routing scheme in Cognitive Radio Networks (CRNs). The authors in [13] examined the routing issue in energy harvesting in a multi-hop CRN communication scenario. RL-based route selection mechanism is developed considering the various factors affecting routings, such as the number of hops, the node's distance, energy consumption during the communication process, and remnant energy. Various routing schemes have been introduced in the context of unmanned aerial vehicle (UAV) communications in complex network environments [14-16]. However, such schemes are associated with bottleneck issues. In this regard, the use of RL for routing algorithms in UAV applications has gained wide attention. The researchers in the study of [17] suggested a Q-learning-based load balancing routing technique to handle relay traffic in UAV communication. The presented technique estimates network load through the queue status obtained from the ground-vehicular nodes. A reward function control is also implemented for quick learning feedback of the reward values under a dynamic communication environment. In [18], a global routing scheme using where each mobile node participates in the route discovery process. The presented global routing protocol is compared with the local routing protocol, where each intermediate node performs routing then based on its energy profile. Both these approaches are designed based on the Q-learning approach of RL. The concept of RL is adopted in routing optimization at the network level for minimizing interferences and delay in channel switching [19]. The routing decision is carried out based on past events and predicted routing decisions of primary users. An RL agent is devised into the cross-layer approach towards assisting the transfer of channel information to the network layer. In [20], adopted deep RL mechanism to address sampling problems and optimal route selection in the highly complex and dynamic network communication scenario. A hierarchical routing scheme based on Q-learning is presented by [21] to enhance message delivery rate performance with less delay and hops in Vehicular Ad-hoc networks (VANET). Here, a network region is divided into different grids, and the presented routing scheme discovers the next optimal grid towards the end-point or target point. It also finds a vehicle moving towards the next optimal grid for communicating data transmission. The authors in [22] presented a collaborative RL model to design a tree-based routing scheme that captures the network's dynamic characteristics, such as several nodes and uncertain traffic, to provide QoS-aware services in Cloud Content Delivery Networks. In [23], the authors have used RL for network traffic engineering. The presented techniques learn to select critical traffic flows in the matrix and reconstruct optimal routes based on flow information in the matrix to balance link utilization in the network. In [24], RL-based intelligent routing is developed to provide a complete view of the network and fast data forwarding process in SDN-enabled networks. In [25], Cluster-oriented cooperative Scheduling scheme using RL to improve vehicular networks' communication efficiency and reliability.

V. Open-AI: Gym and Network Environment Setup

With the appropriate function approximation, the methods like Q-Learning and policy gradients may provide better performance even under challenging environments. The popular benchmark for RL includes 1) Arcade Learning Environment (ALE), and 2) RL Lab (RLL).

Recently, Open-AI-Gym is the most popular benchmark with the following essentials into it:

- Combines best elements of ALE and RLL and having a diverse collection of tasks (Environment) with a standard interface.
- OpenAI-Gym provides the episodic setting of RL, where the agent experiences are broken down into a series of episodes.
- The agents' initial State is randomly sampled from distribution in each episode. The interaction proceeds until the environment reaches a terminal state.
- The goal in episodic RL is to maximize the expectation of total reward per episode and achieve a high level of performance in a few episodes as possible.
- OpenAI does not include an agent class.

Process: Function of -Open-AI Gym

```
1. Sample environment state [ return first observation]:
 Ob0 = env.reset()
2. Agent chooses first action:
 A0=agent.act(Ob0)
3. Environment returns observations:
 Info =env.step(A0)
 Ob1, Rew0, Done0
4. Reward and a Boolean flag indicates
- if the episode is complete
 A1=agent.act(Ob1)
 Env.step(A1)
 Ob2, Rew1, Done1
 [A99 =agent.act(O99)]
 Info= env.step(A99)
Ob100, rew 99, Done99
5. Done 99 = = True →Terminal
```

Design assumptions in OpenAI-Gym include i) the Only environment but no agent, ii) Emphasis on the sample, not just final performance, iii) Encourage peer review, not competition, iv) Strict versioning of environment, v) Monitoring by default.

OpenAI-Gym is a collection of partially observable Markov decision processes (POMDP), and the current environment consists of i) Algorithms, ii) Atari, iii) Box2D, iv Classic Control, iv) Mujoco, v) Robotics, and vi) Toy text. It does not consider any environment explicitly for a network; therefore, a custom environment for the network is created as SimpleNetwork with a different number of nodes.

VI. Q-ROUTING MODEL VALIDATION WITH RULE-BASED ROUTING

This section presents outcome analysis and performance assessment of the proposed Q-routing on the customized Open-AI Gym environment, namely, Net-AI Gym [26]. The development and design of the proposed system are carried out on the numerical computing platform, and scripting of the proposed technique is done in Python. Different case studies have been considered to evaluate the stability and consistency of both Net-AI gym and Q-routing scheme in the performance analysis.

A. *Case:1 Network with 6-Nodes*

In this scenario, a network with six nodes is being considered.

Table I presents test case one with a network scenario with six nodes.

TABLE I. NETWORK SCENARIO-1: 6-NODES

Observation space	Discrete (6)
Action space	Discrete (6)

In this scenario, a network with six nodes is being considered. The agent will be examined with various networks containing various nodes. Both rule-based algorithms, as well as Q learning algorithms will be tested. The networks are designed in such a way that they simulate the actual internet.

Fig. 5 shows the network deployment scenario. The network contains six nodes. Randomly two nodes are selected and made into source and destination. The Net-AI gym platform has a unique reward system where if the agent tries to transfer the packet from one node to another, the packet will be dropped if the nodes are not connected. Every transfer agent will get a negative reward. However, the agent will get a positive reward when the packet reaches its destination. So in a way, the reward shows the throughput of the system.

Fig. 6 shows how the throughput is increasing over some time. After many episodes, the throughput starts increasing. The graph clearly shows the increase in the throughput and hence the reward. The agent initially takes some random moves to explore the network.

In Fig. 7 shows an analysis of the proposed Q-routing scheme concerning Epsilon Vs; the episode with six nodes network. The epsilon represents the probability of the agent taking up random moves. As can see, as the episodes progress, epsilon decays down. This means, in the initial episodes, the agent may take more random moves; the probability of it reduces as the episodes progress.

Fig. 8 shows an analysis of the proposed Q-routing scheme concerning Episodes Vs. path length. The path length of 0 represents that the packet is lost. The path length of 2 is optimal. If we can observe the epsilon plot and the path length plot together, path length reduces to optimal length when epsilon decay below 1% which means the agent stops exploring.

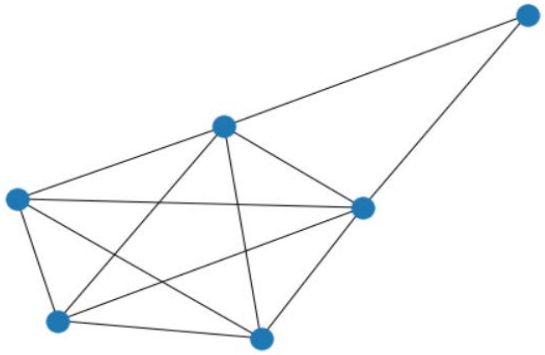

Fig. 5. An Environment with 6- Nodes Network.

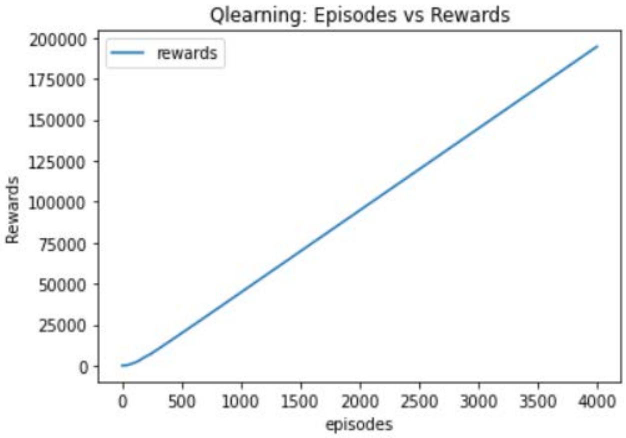

Fig. 6. Network with Node=6, Episodes vs. Rewards.

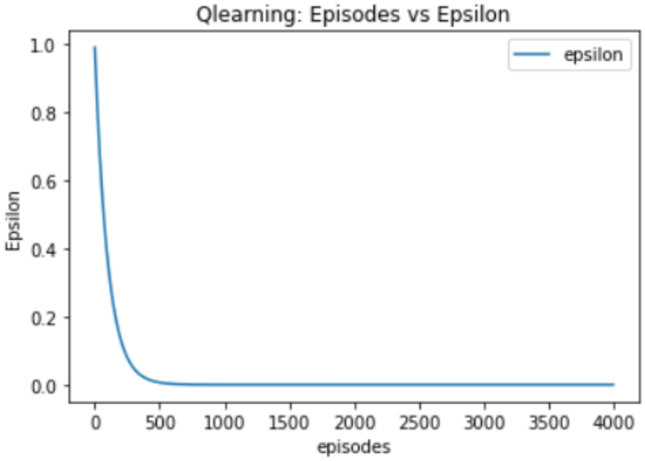

Fig. 7. Network with node=6, Epsilon Vs. Episode Define Episode.

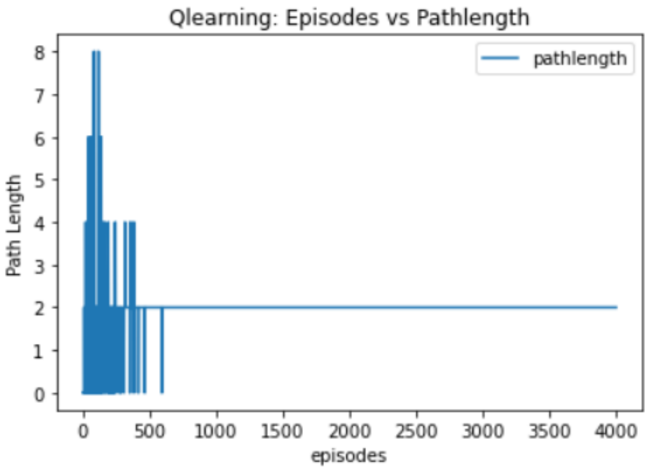

Fig. 8. Network with Node=5, Episode vs. Pathlength.

B. *Case:2 Network with 50-Nodes*

In this scenario, a network with fifty nodes is being considered. Table II presents test case one with a network scenario with six nodes.

TABLE II. NETWORK SCENARIO-2: 50-NODES

Observation space	Discrete (50)
Action space	Discrete (50)

However, when the network is with higher nodes, as shown in Fig. 9, the study comes across a different type of result. The probability of dropping is much more compared to a smaller network. As it can be observed, the agent will start delivering packets only after 1000 episodes.

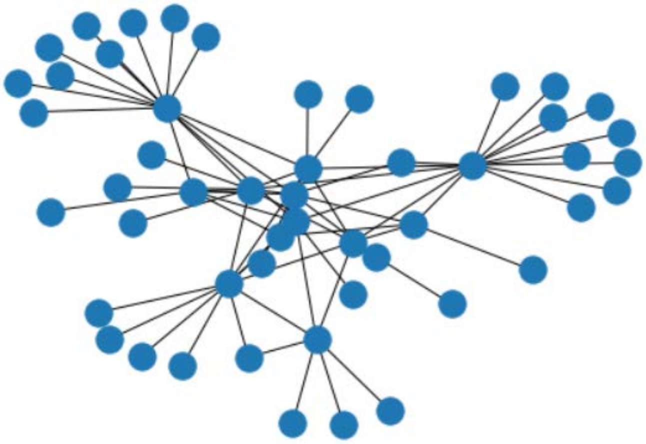

Fig. 9. An environment with 50- Nodes Network

This indicates that the more significant nodes, the more time the agent will take to learn the correct path. This is as expected.

As shown in Fig. 10, 11, and 12, even though the epsilon decay is much similar to that of the lower node network, if we observe the path length graph, the agent takes a very long time to find any let alone the longer path. This is because the network contains many gateway nodes. Due to which there will be many nodes connected to a common node. This exactly happens in a real-world network scenario. The gateways are usually connected to a high number of other nodes and national links. Very rarely will any node be connected directly to a significant network. As it can be observed, the optimal path length here is 4. Even with 50 nodes, the optimal path length is only 4. This is true even in a real-world scenario.

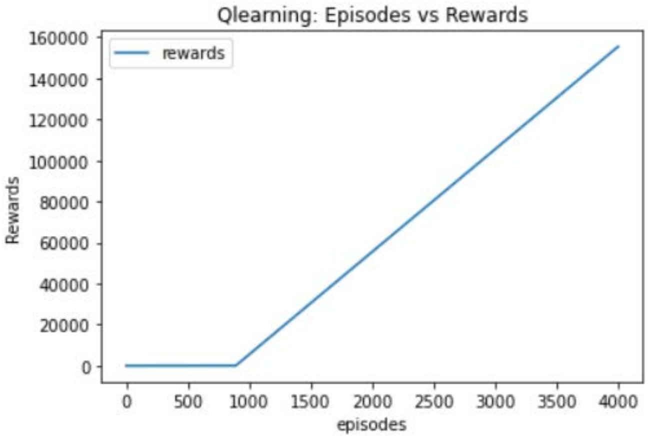

Fig. 10. Network with node=50, Episode vs. Rewards.

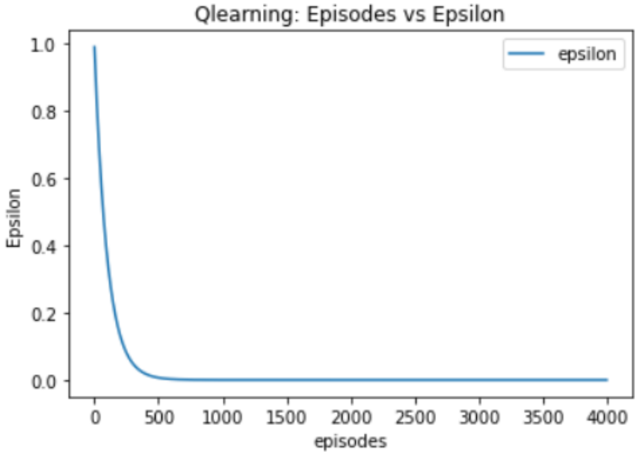

Fig. 11. Network with node=50, Episode vs. Epsilon.

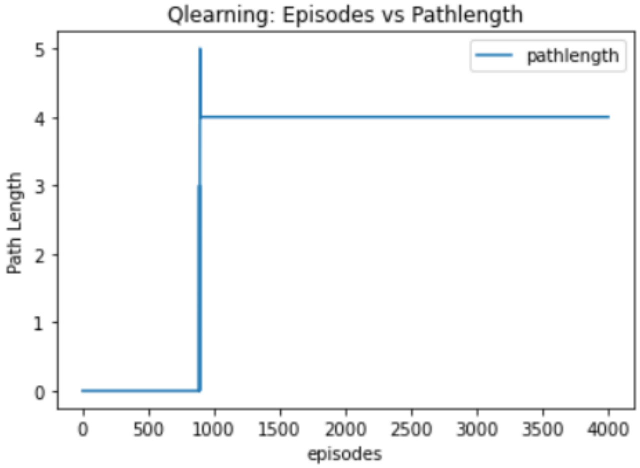

Fig. 12. Network with node=50, Episode vs. Pathlength.

C. *Case:3 Network with 100-Nodes*

In this scenario, a network with fifty nodes is being considered. Table III presents test case one with a network scenario with six nodes.

TABLE III. NETWORK SCENARIO-3: 100-NODES

Observation space	Discrete (100)
Action space	Discrete (100)

Fig. 13 shows a network environment with 100 nodes. The study has adopted the monte Carlo method here to evaluate the algorithm. A different case study has been considered to see how the algorithm performs with an increasing number of nodes. The scalability of the network routing algorithm is an essential aspect.

In Fig. 14, the 100-node network takes around 2000 episodes to learn the optimal path. As shown from graph trend, till episode 2000, there is no reward at all. Only after the algorithm learns the route, it starts to perform.

Fig. 15 shows an analysis of the proposed Q-routing scheme concerning Epsilon Vs; an episode with a 100 nodes network. The epsilon represents the probability of the agent taking up random moves. As can see as the episodes progress, Epsilon decay is the same as usual. The same rule works here as well.

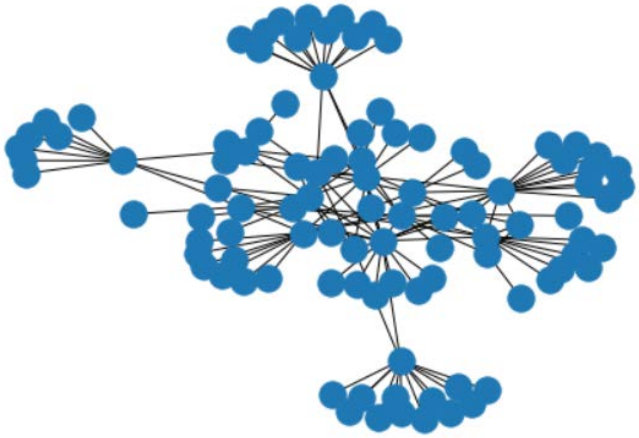

Fig. 13. An Environment with 100- Nodes Network.

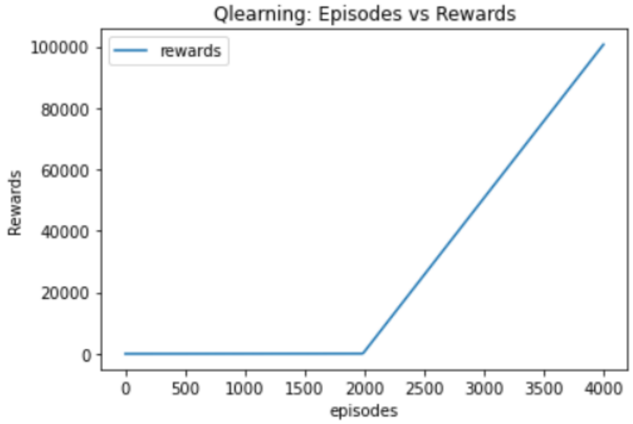

Fig. 14. Network with node=100, Episode vs. Rewards.

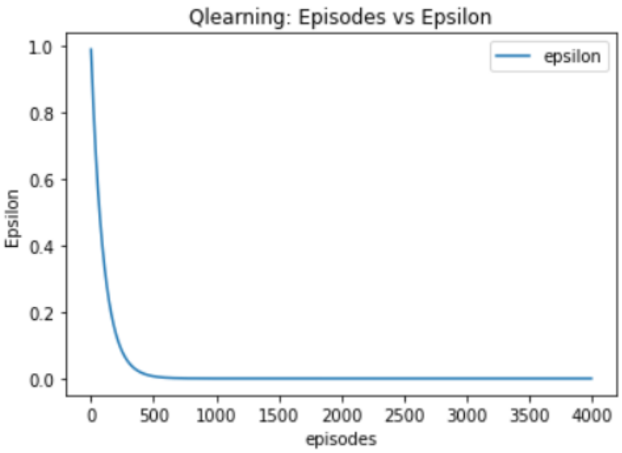

Fig. 15. Network with node=100, Episode vs. Epsilon.

Fig. 16 shows an analysis of the proposed Q-routing scheme concerning Episodes Vs. path length. As shown from the path length plot, the algorithm finds a suitable path after 2000 episodes. Even after epsilon decay fall below 1%, the algorithm has not found a path yet. Even if the algorithm does not explore, the algorithm finds a path in the network. Also, it can be analyzed based on the closer analysis that the optimal path length here is 3, and it has been earlier; there is no relationship between optical path length and the number of nodes.

Fig. 17 demonstrates a comparative analysis for all three case studies of different network sizes. As it can be observed, once the algorithm learns, then the performance is the same on all the networks. However same is not true with rule-based methods.

As it can be seen in Fig. 18, the rule-based method never settles for a single path. This is due to the dynamic nature of the network. Connections and the weights keep changing. Hence, the path length also keeps varying.

Fig. 19 shows an analysis of Episodes vs. Pathlength for the proposed Q-learning scheme. In contrast to the path length plot of the rule-based method, the Q learning method always follows the optimal path after some time since it can predict the changes in the network before they can occur.

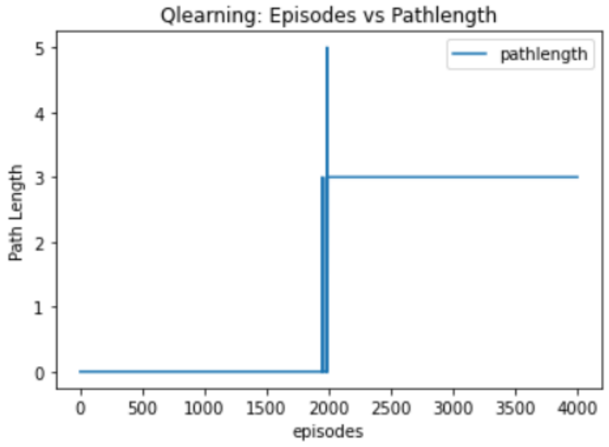

Fig. 16. Network with node=100, Episode vs. Pathlength.

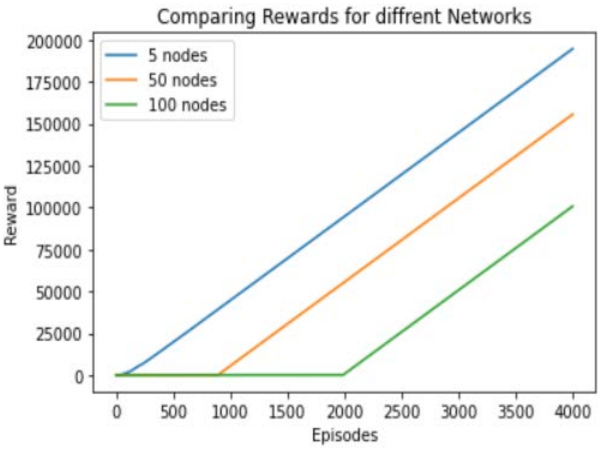

Fig. 17. Comparison Scenario-1, Two and, 3.

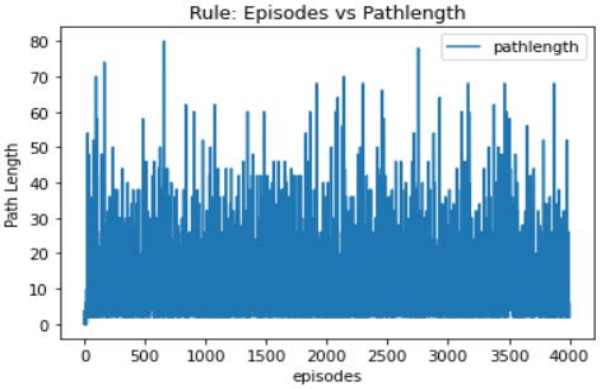

Fig. 18. Rule-Based: Episodes vs. Pathlength.

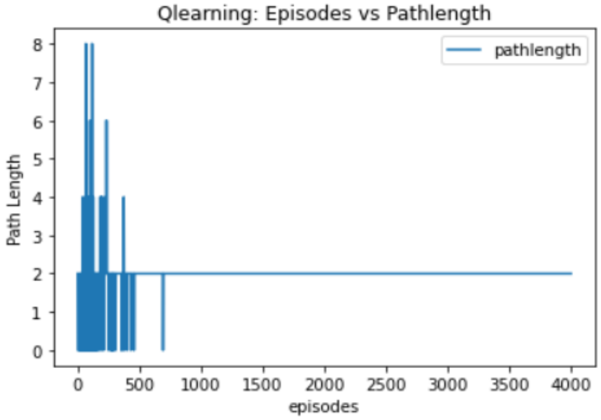

Fig. 19. Q-Learning Based: Episodes vs. Pathlength.

From the comparative analysis in Fig. 20, it can be analyzed that the rule-based method exhibited higher throughput. However, the Q-routing method is highly efficient once it learns the policy. Hence, in the long run, Q learning produces higher throughput compared to the rule-based method. Based on the overall analysis it can be observed that the proposed system offers a good scope for solving network related problem with RL agent algorithm, which can be well evaluated in the proposed customized Net-AI-Gym environment. The proposed system is found to be scalable to both small network and large network as it perform well in all the three cases of network with different number of nodes. Based on the comparative analysis the proposed Q-learning algorithm outperform the rule-based algorithm in terms of episode vs. reward which shows the stability and efficiency of proposed system to address routing related problem.

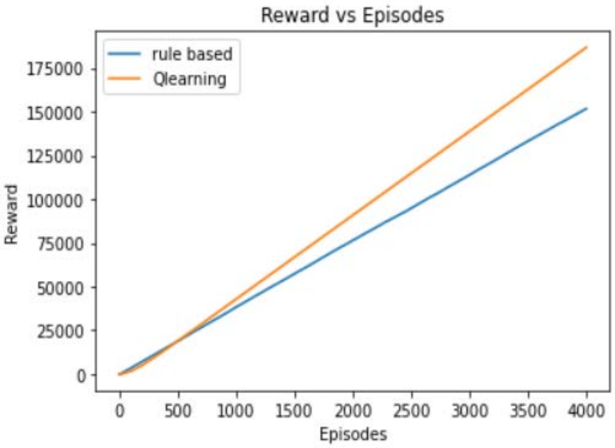

Fig. 20. Rule-Based vs. Q-Learning (Episodes vs. Reward).

VII. Conclusion

In this paper, the evaluation of the newly designed environment on the Open-AI Gym, namely Net-AI-Gym, is performed for agent algorithm based on the Q-learning is evaluated. The scalability limitation of network routing protocols using rule-based methods is being overcome with the RL-based Q-routing protocol. The performance validations between rule-based and RL-based Q-learning are carried out, and it is being found that the Q-learning performs better than rule-based routing protocols concerning episodes Vs. Rewards. The self-validation test for scalability for varying nodes provides a consistent result. However, the constraints of space optimization are planned as future research work. This is the first of its kind of work, custom-developed for routing protocol on the newly designed environment. In the future work, the proposed work can be extended towards improving the performance of the system and evaluation of RL algorithm on the customized Net-AI Gym environment with different technique dealing with a complex, high-dimensional state space.

References

[1] Liu, K.R., Sadek, A.K., Su, W. and Kwasinski, A., 2009. Cooperative communications and networking. Cambridge university press.

[2] Balasubramaniam, Deepa. (2015). Computer Networking: A Survey. International Journal of Trend in Research and Development, 2.

[3] Jaffali S., Jamoussi S., Khelifi N., Hamadou A.B. (2020) Survey on Social Networks Data Analysis. In: Rautaray S., Eichler G., Erfurth C., Fahrnberger G. (eds) Innovations for Community Services. I4CS 2020. Communications in Computer and Information Science, vol 1139. Springer, Cham.

[4] Triantafyllou, Anna, Panagiotis Sarigiannidis, and Thomas D. Lagkas. "Network protocols, schemes, and mechanisms for the internet of things (iot): Features, open challenges, and trends." Wireless communications and mobile computing 2018 (2018).

[5] X. Guo, H. Lin, Z. Li and M. Peng, "Deep-Reinforcement-Learning-Based QoS-Aware Secure Routing for SDN-IoT," in IEEE Internet of

Things Journal, vol. 7, no. 7, pp. 6242-6251, July 2020, doi: 10.1109/JIOT.2019.2960033.

[6] Z. Jin, Q. Zhao and Y. Su, "RCAR: A Reinforcement-Learning-Based Routing Protocol for Congestion-Avoided Underwater Acoustic Sensor Networks," IEEE Sensors Journal, vol. 19, no. 22, pp. 10881-10891, 15 Nov.15, 2019, doi: 10.1109/JSEN.2019.2932126.

[7] S. Wang and Y. Shin, "Efficient Routing Protocol Based on Reinforcement Learning for Magnetic Induction Underwater Sensor Networks," in IEEE Access, vol. 7, pp. 82027-82037, 2019, doi: 10.1109/ACCESS.2019.2923425.

[8] P. Nurmi, "Reinforcement learning for routing in ad hoc networks," in Proc. 5th Int. Symp. Modeling Optim. Mobile, Ad Hoc Wireless Netw., Limassol, Cyprus, 2007, pp. 1–8.

[9] B. Debowski, P. Spachos, and S. Areibi, "Q-learning enhanced gradient-based routing for balancing energy consumption in WSNs," in Proc. 21st IEEE Int. Workshop Comput. Aided Modelling Design Commun. Links Netw. (CAMAD), Oct. 2016, pp. 18–23.

[10] K. Tang, C. Li, H. Xiong, J. Zou, and P. Frossard, "Reinforcement learning-based opportunistic routing for live video streaming over multi-hop wireless networks," in Proc. IEEE 19th Int. Workshop Multimedia Signal Process., Oct. 2017, pp. 1–6.

[11] X. Shen, Q. Chang, L. Liu, J. Panneerselvam and Z. Zha, "CCLBR: Congestion Control-Based Load Balanced Routing in Unstructured P2P Systems," in IEEE Systems Journal, vol. 12, no. 1, pp. 802-813, March 2018, doi:

[12] X. Li, X. Hu, R. Zhang, and L. Yang, "Routing Protocol Design for Underwater Optical Wireless Sensor Networks: A Multiagent Reinforcement Learning Approach," in IEEE Internet of Things Journal, vol. 7, no. 10, pp. 9805-9818, Oct. 2020, doi: 10.1109/JIOT.2020.2989924.

[13] X. He, H. Jiang, Y. Song, C. He, and H. Xiao, "Routing Selection With Reinforcement Learning for Energy Harvesting Multi-Hop CRN," in IEEE Access, vol. 7, pp. 54435-54448, 2019, doi: 10.1109/ACCESS.2019.2912996.

[14] Kashyap, A.; Ghose, D.; Menon, P.P.; Sujit, P.; Das, K. UAV aided dynamic routing of resources in a flood scenario. In Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019; pp. 328–335. 2. Zeng, F.;

[15] Zhang, R.; Cheng, X.; Yang, L. UAV-assisted data dissemination scheduling in VANETs. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.

[16] Yang, L.; Yao, H.; Wang, J.; Jiang, C.; Benslimane, A.; Liu, Y. Multi-UAV Enabled Load-Balance Mobile Edge Computing for IoT Networks. IEEE Internet Things J. 2020, 7, 1

[17] Roh, B.S., Han, M.H., Ham, J.H. and Kim, K.I., 2020. Q-LBR: Q-Learning Based Load Balancing Routing for UAV-Assisted VANET. Sensors, 20(19), p.5685.

[18] Mili R., Chikhi S. (2019) Reinforcement Learning Based Routing Protocols Analysis for Mobile Ad-Hoc Networks. In: Renault É., Mühlethaler P., Boumerdassi S. (eds) Machine Learning for Networking. MLN 2018. Lecture Notes in Computer Science, vol 11407. Springer, Cham.

[19] Safdar Malik, T. and Hasan, M.H., 2020. Reinforcement Learning-Based Routing Protocol to Minimize Channel Switching and Interference for Cognitive Radio Networks. Complexity, 2020.

[20] Y. Shao, A. Rezaee, S. C. Liew and V. W. S. Chan, "Significant Sampling for Shortest Path Routing: A Deep Reinforcement Learning Solution," in IEEE Journal on Selected Areas in Communications, vol. 38, no. 10, pp. 2234-2248, Oct. 2020.

[21] F. Li, X. Song, H. Chen, X. Li and Y. Wang, "Hierarchical Routing for Vehicular Ad Hoc Networks via Reinforcement Learning," in IEEE Transactions on Vehicular Technology, vol. 68, no. 2, pp. 1852-1865, Feb. 2019, doi: 10.1109/TVT.2018.2887282.

[22] M. He, D. Lu, J. Tian, and G. Zhang, "Collaborative Reinforcement Learning Based Route Planning for Cloud Content Delivery Networks," in IEEE Access, vol. 9, pp. 30868-30880, 2021, doi: 10.1109/ACCESS.2021.3060440.

[23] J. Zhang, M. Ye, Z. Guo, C. -Y. Yen and H. J. Chao, "CFR-RL: Traffic Engineering With Reinforcement Learning in SDN," in IEEE Journal on Selected Areas in Communications, vol. 38, no. 10, pp. 2249-2259, Oct. 2020, doi: 10.1109/JSAC.2020.3000371.

[24] D. M. Casas-Velasco, O. M. C. Rendon and N. L. S. da Fonseca, "Intelligent Routing Based on Reinforcement Learning for Software-Defined Networking," in IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 870-881, March 2021, doi: 10.1109/TNSM.2020.3036911.

[25] Xia, Y., Wu, L., Wang, Z., Zheng, X. and Jin, J., 2020. Cluster-Enabled Cooperative Scheduling Based on Reinforcement Learning for High-Mobility Vehicular Networks. IEEE Transactions on Vehicular Technology, 69(11), pp.12664-12678.

[26] Varshini Vidyadhar, Nagaraj R, D.V. Ashoka, "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), Vol. 12, No.4, pp. 169-176, 2021.