2024 13th International Conference on System Modeling & Advancement in Research Trends (SMART) | 979-8-3503-8058-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/SMART63812.2024.10882526

Comparative Analysis of Machine Learning Application on Routing Protocols in FANET

Khushbu Jaiswal[1], Sudesh Kumar[2] and Neeraj Kumar Rathore[3]
[1,2,3]Department of Computer Science, Indira Gandhi National Tribal University, Amarkantak, Madhya Pradesh–484887, Madhya Pradesh, India
E-mail: [1]khushbujaiswal963@gmail.com, [2]sudesh.kumar@igntu.ac.in, [3]neeraj.rathore@igntu.ac.in

***Abstract*—Flying ad hoc networks (FANET) are defined by their energetic topology and the absence of fixed infrastructure, which makes efficient routing an important challenge. Traditional routing protocols faced an indeed imposing problem, namely they could not at all efficiently react to network speed fluctuations that often cause topology changing that in turn results into low throughput, high delay, packet loss, and disconnection. This is helped by emerging methods like Machine Learning (ML), which are able to learn by maximizing learning and self-learn. In this paper, we review quite a number of powerful ML techniques and their application to routing, documenting the key strengths and weaknesses of all. We also review how the above stated ML approaches have been applied to distinct current work.**
Keywords: FANET, UAVs, ML, Routing Protocol, Reinforcement Learning

I. Introduction

Swarms of Unmanned Aerial Vehicles (UAVs) are becoming commonplace, and are comprised of flying ad hoc network (FANET). It provides flexible and scalable communication solutions to precision farming, remote sensing, surveillance, search and destroy, search and rescue [1]. Owing to its unique characteristics, such as high mobility, fast topological changes and constrained on board resources, FANET presents a challenging system that greatly differs from traditional ad hoc networks. Reliable communication depends only on routing, and it lies in the base of FANET system. This leads to the network's flexibility requiring adaptive routing, which is often crucial for FANET systems, which are inherently dynamic by nature [2].

FANET system is highly reliant on the routing for reliable communication and also effective. There are many existing routing protocols but their effectiveness in terms of coping with features specific to FANET is not much [3]. There are two widely used classical routing, such as topological and positional based routing. High overhead from the first method since established connections between nodes occur before transmission is required. This difficulty is overcome by a new technique that sets up a connection to be used as soon as a node wants to transmit data. On the other hand, the strategy has end to end latency. In order to minimize the amount of overheard by the least amount of time, position-based routing was developed; routing tables were maintained using a GPS based method. Although, these protocols have drawbacks like traffic congestion taking place in networks with high traffic density [4–7].

More and more Routing protocols are being improved by Machine Learning (ML) for FANET. The varied ML approach can better apply to the dynamic and unpredictable FANET nature of these protocols, leading to more reliable and effective communication. These protocols can use ML approaches to gain improved route identification and maintenance; to dynamically adapt to changing network conditions; and maximize network performance [8]. With ML based routing, routing protocols learn from the network conditions and take intelligent routing decisions, and it provides a workable solution that existing routing protocols can learn from. The goals of this research article are motivated by the routing problem in the FANET where the aerial node's high mobility and dynamic topology frequently make the established solutions no longer work. Since there will be a comparative analysis of these models, one will gain a thorough understanding of the advantages and disadvantages of ML techniques on routing in FANET. This will not only enhance further study in ML applications to aerial networks, but also help produce more effective routing protocols.

The remainder of the paper is sectioned as: Section 2 discusses the fundamentals of ML as well as several routing-efficient ML algorithms. The comparative analysis of various ML techniques on the basis of routing are analysed in section 3. In Section 4 applications of ML technique on routing in FANET are covered. Section 5 conclude the paper.

II. ML Techniques

ML includes a variety of methods, such as neural networks, supervised, unsupervised, and Reinforcement Learning (RL) [9]. By enhancing decision-making processes and offering insights into network activity, ML can improve routing algorithms in the context of FANET. Supervised and unsupervised learning methods

Copyright © IEEE–2024 ISBN: 979-8-3503-8058-3

Authorized licensed use limited to: University of Saskatchewan. Downloaded on May 20,2025 at 21:31:22 UTC from IEEE Xplore. Restrictions apply.

are not well adapted to dynamic situations since they are primarily concerned with classification and regression. On the contrary however, these kinds of settings are better handled by RL algorithms that learn and find the optimal sequence of actions while maximizing some defined objective function. Thus, algorithms for RL are employed for determining optimal routing policies using learning from interactions with the environment [10]. In FANET these algorithms are able to maximize routing choices based on the network variables such as node mobility, link quality and energy constraints in order to manage the network dynamic and unpredictable character. Listed below are a few of the principal methods:

A. Reinforcement Learning (RL)

Optimal routing can be achieved via reinforcement learning which, rather than inputting discrete patterns, permitting rewards and punishments to determine the consequences of its actions, is trained to do different actions, learning from the results. The reward and punishment strategy, however, uses the maximum reward available in the environment, in order to improve state prediction [11]. This problem is a version of Markov decision process problem that is based on four parameters: S, A, P, and R, where S is finite states, A is finite action space, and R is reward values from the environment, and P is the transition function from current s to s' after taking an action a and receive immediate reward (r) of environment. The present state alone determines which node will be selected next. To determine the appropriate course of action for making decisions, the value iteration function is employed; the value iteration function V*(s) is computed as follows [12]:

$$V^{*} \leftarrow max_{a \in A(s)}[R(s,a) + \tau \Sigma_{s\prime \in S} P(s'|s,a) V^{*}(s')] \quad (1)$$

where V* is the ideal value function for the current state and the action made to maximize the expected return by beginning in state s and adhering to the best policy. R(s,a) represents the immediate reward following an action, while τ stands for the discount factor, which indicates the significance of the reward in the future. $P(s' \mid s,a)\, V^{*}(s')$ is the probability function gives the expected value of being in state s and taking action a, considering all possible next states s′. Based on this equation MDP update its value function for each current state until it convergence to the optimal policy.

B. Q-Learning

A well-liked RL technique called Q-learning is model-free, off-policy, and operates on the tenet that every node is a learner that absorbs information from the surroundings and keeps track of a Q-table to facilitate the best possible decisions. The agent explores the environment, selects actions based on a policy, and receives rewards while transitioning from the current state to the next state. Here's a detailed explanation of how Q-learning updates its Q-table [13]:

$$Q(S_{t+1}, a_{t+1}) \leftarrow Q(s_t, a_t) + \alpha[r_t + \tau max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)] \quad (1)$$

Where, the agent obtains reward r_t and moves into a new state s_t', and a represents an action taken in state s_t. The significance of future reward in relation to immediate gains is determined by the discount factor τ. The learning rate, or α, indicates how much newly learned information replaces previously acquired knowledge. $\tau max_{a'} Q(s_{t+1}, a')$ represents the maximum Q-value for the next stage, which indicates the best potential rewards in the future.

In the FANET environment, each node represents the state denoted by s, and the neighbor table maintains the node in the state of transition produced by action a. The next state is determined by a reward function based on the Q-learning algorithm, which estimates and rewards the information in the neighbor table. When there is no environment model available, Q-learning works by determining the best course of action. Moreover, it is adaptive enough to work in a stochastic environment. In order to minimize the computing cost, it also looks at a limited set of basic mathematical operations when updating the routing table.

C. Deep Q-Network (DQN)

Large state spaces make Q-Learning useless because large Q-table entries make reading from and saving to the table less efficient. This is because table lookups are used for decision-making in small state spaces, which is where Q-Learning performs best. Deep RL can be used to improve it for restricted environment utilization. Deep Q network (DQN) constructs with the merger of the RL and the deep learning by using a deep neural network (DNN), instead of a Q table. Generalizing the state space over RL enable to make better decisions on states that were not seen during training. DQN is particularly helpful for FANET, where the routing problem is discrete actions, and the state space is complex [14].

In this model, Q-evaluation network and Q-target network are used in order to minimize the sample correlation. To update the parameters of the present network model from the Q-evaluation network the probability of the cumulative reward for each action under a particular condition is predicted. Q-target network offers training with stable target Q-values. It is used as a guide to determine the desired Q-values that are needed to update the Q-evaluation network In the training phase, the update function shows as follows:

$$Q_{Eval}(S_t, a_t) \leftarrow (1-\alpha) Q_{Eval}(S_t, a_t) + \alpha[R_t + \tau max_{a'} Q_{Tar}(S_{t+1}, a')] \quad (2)$$

Authorized licensed use limited to: University of Saskatchewan. Downloaded on May 20,2025 at 21:31:22 UTC from IEEE Xplore. Restrictions apply.

The equation is used to update the Q-value estimate $Q_{Eval}(St,a_t)$ for taking action a_t in state s_t. Here, α symbolizes the learning rate, which determines the degree to which the current estimate is influenced by new information and the discount factor is represented by τ. The target value is denoted by the $R_t + \tau max_{a'}\ Q_{Tar}\ (S_{t+1},a')$ which combines the target network's projected discounted maximum future reward, Q_{Tar}, with the immediate reward, R_t. The update rule blends the old Q-value with this target value to iteratively refine the evaluation network's estimates, guiding it to approximate the true Q-values more accurately and improving the policy derived from these values.

D. Deep Deterministic Policy Gradient (DDPG)

When it comes to solving infinite state space problems using the value function approximation, the Deep Deterministic Policy Gradient (DDPG) approach is highly effective. The approach determines action values in continuous space. Target actor-network, target critic network, actor-network, and critic network are the four neural network parameters that the model-free DDPG applies to. Each of these networks has a unique function for routing decision optimization [15].

Firstly UAV nodes interacts with environment and stores the obtains tuples [s,a,r,s'] in replay buffer. Given a state-action pair and [s,a], the critic network calculates the Q-value to evaluate the caliber of the acts carried out by the actor network. The estimated Q-value for a state-action combination is as follows [s,a]:

$$Q(S_t,a_t \mid \theta^Q) \tag{3}$$

Here, the state at time t is denoted by s_t, while action a which made to select a route is denoted by θ^Q. The Critic Network's parameters are utilized to compute the Q-value for both state s_t and action a_t.

The target actor is employed to ensure the stability of the training process. It generates actions that are used in the target Q-value calculations for the Critic Network. It takes s' as input and generate a''. A copy of the Critic Network, the target critic is used to provide stable Q-value targets throughout the training phase. The network generates the Q-value function Q(s', a'') from two inputs, s' and a''. The following formula determines the desired Q-value, which is used to update the Critic Network during training:

$$Y_t \leftarrow r_t + \tau Q'\ (S_{t+1}\ , \mu'(S_{t+1}) | \theta^{Q'} \tag{4}$$

Here, the instant reward that is received following an action in a particular state s_t is represented by the variable r_t. The discount factor, represented by τ, determines the relative value of future rewards compared to immediate gains. The action $\mu'\ (S_t+1)$ that the Target Actor Network recommends for the subsequent state $\theta^{Q'}$ is the parameters of the Target Critic Network, while the Q-value predicted for the next state S_{t+1} is represented by $Q'\ (S_{t+1})\ , \mu'\ (S_{t+1})|\theta^{Q'}$. With this approach, DDPG can continuously improve routing policies given dynamic network conditions, optimizing, for instance, for latency minimization and throughput maximization.

E. Graph Neural Network (GNN)

This technique is used to optimize graph-structured data, which makes it able to solve optimization problems. A Graph Neural Network (GNN) is a graph in which each node assigns some features, and each edge, or relationship, between two nodes has weights or some form of data. The model can solve optimization problems such as improving scalability, path selection optimization, how to control congestion with network topology information based on traffic congestion or link failure, and how to adapt to network topology changes frequently under historical data [16].

In order to communicate optimally, each node updated its representation based on its own feature, as well as the features of those nearby nodes, according to the update rule used. This is the representation of the node update rule:

$$h_i^{[L]} = \sigma\left(W.AGGREGATE\ \left(\left\{h_i^{[L-1]} : j \in n(I)\right\}\right) + b\right) \tag{5}$$

Here, node i representation at layer L is represented by $h_i^{[L]}$, the aggregation function aggregates the features of nearby nodes, W denotes weight matrix, and the activation function is represented by σ. N (i) indicates node i's neighbors.

The GNN application can improve routing performance in an ad hoc network by leveraging its distinctive neural network-based graph-structured model along with other ML approaches. The use of graph-based nearby node selection by GNN can improve UAV reliable communication performance.

III. Comparision

A comparative examination of different ML algorithms for FANET routing is presented in Table I. It assesses the feasibility of several ML techniques, including MDP, Q-Learning, DQN, DDPG, and GNN based on its type, key features, and limitation. The table evaluates each technique's efficacy in routing protocol optimization and overall network performance, highlighting how it addresses the particular issues faced by FANET, such as node mobility and fast changes in network topology.

Table I: Comparative Analysis of ML Techniques Suitability on Routing In Fanet

ML Techniques	Type	Advantages	Limitations
MDP	Model based	Provides a clear and rigorous framework for decision-making. Discrete; manageable in size.	Less scalable. Inefficient in large or continuous state/action spaces.
Q-Learning	Model free	Efficient and Stable for small, discrete problems. Intuitive principle logic. Low computational complexity for small state-action spaces.	less scalable. Infeasible for large spaces.

Copyright © IEEE–2024 ISBN: 979-8-3503-8058-3

Authorized licensed use limited to: University of Saskatchewan. Downloaded on May 20,2025 at 21:31:22 UTC from IEEE Xplore. Restrictions apply.

DQN	Model free	Performs well in large state and action space. balanced exploration and exploitation strategy.	High computational complexity due to neural network training and experience replay. Limited to discrete action spaces.
DDPG	Model free	Well-suited for complex routing. Highly adaptable to continuous and dynamic environments.	High computational complexity due to actor and critic network and continuous action space handling.
GNN	Model-based	Can handle large network. Effectively adapt quality of changing links.	Overfitting risks. Complex to implement. Required high computation.

The MDP is good for discrete problems and provides a clear, model-based framework; nevertheless, it has issues with scalability and efficiency in larger or continuous state/action spaces. In contrast, the model-free, table-based Q-Learning technique is unscalable and unworkable for bigger regions, but it is effective for discrete, small problems with low computational cost. DQN Best suited for extensive state and action domains that adopt a well-rounded strategy for investigation and utilization; nonetheless, its neural network training makes it computationally demanding and restricted to distinct actions. Ultimately, dynamic, complex systems with continuous action spaces and high flexibility are most suited for DDPG, while its actor-critic framework places heavy computing demands on it. In vast state spaces, the GNN is effective for network topology learning and route optimization, and it can operate in both discrete and continuous contexts. In addition to being difficult to use and requiring a lot of processing resources, the model may experience overfitting.

IV. Applications

Nowadays ML becomes great support in the area of routing in FANET. The essential application of ML techniques in routing shown in "Fig. 1". In this section applications of mentioned ML techniques in routing of FANET has been discussed.

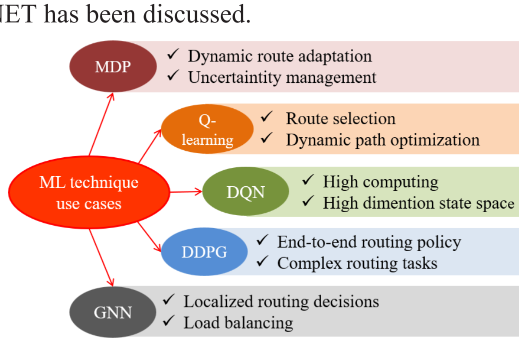

Fig. 1: ML Technique Use Cases on Routing in FANET

A. RL

The traditional routing protocol was unable to adjust to the dynamic characteristics of FANET. As a result, a reinforcement learning-based self-learning routing protocol (RLSRP) and a position prediction-based directional MAC (PPMAC) protocol were introduced [17]. These protocols utilized a partially observable MDP with a greedy policy to monitor changes in network topology and make optimal decisions for route selection. On the other hand, the protocol took few routing characteristics into account and employed preset learning parameters. Similar to this adaptive routing protocol using generative adversarial imitation learning (AR-GAL) [18] utilizes the MDP technique in order to get improved routing decision-making with less delay and improved packet loss, However, the protocol is not effective at low flight densities. A multi-objective Markov Decision Process (MOBMDP) protocol was put out by the author in [19] for use in particular search-based activities. In order to create a better routing protocol for optimal decision-making and to provide average delay with high PDR, the author used the MDP and Q-learning models while taking into account a variety of crucial parameters, including energy constraints, power distance ratio, delay, message delivery ratio, and network lifetime.

B. Q-Learning

The Q-Learning technique works best in small state space and can be applied to find optimal route discovery and link stability in FANET. The Q-learning technique is used in the Modified Q-learning AODV (QLAODV) [20] protocol to forecast mobility and balance the load on the AODV routing protocol, resulting in improved path selection. They have enhanced connection stability by anticipating nodes' motions through the application of the Gaussian filter method and the Kalman filter. In terms of balancing routing overhead and delay, this protocol falls short. Another protocol, the Q-Learning-based stepwise routing protocol (QSRP) [21], takes into account a central controller to determine the minimum hop count and link quality, which are utilized by the Q-Learning algorithm to determine the best routing. They have used adaptively adjusted hello message intervals but weren't able to reduce network overhead. In order to reduce packet loss and network overhead, Q-learning-based Geographic FANET Routing (QLGR) [22] method based on multiagent RL was presented as a solution to this issue. The protocol offers a distributed method for evaluating neighboring nodes based on local information, such as the nodes' energy, packet queue length, and network quality. However, because it employs preset learning parameters, the protocol is unable to adjust to the dynamic nature of FANET. Q-Learning enabled highly dynamic and latency-aware routing (QEHLR) [23] encounters the same problem in a similar way. Dynamic learning parameters are needed to make

Copyright © IEEE–2024 ISBN: 979-8-3503-8058-3

Authorized licensed use limited to: University of Saskatchewan. Downloaded on May 20,2025 at 21:31:22 UTC from IEEE Xplore. Restrictions apply.

routing algorithms more efficient. Q-Learning-based smart clustering routing (QSCR) [24] used adaptive learning rate as a result, which allowed the protocol to successfully adapt to the rapidly changing topology.

C. DQN

Q-Learning lacks in large state space where DQN applied very efficiently. The Deep Q-network based vertical routing (DQN-VR) [25] used DQN technique in order to achieve scalability and stability in the network by reducing state and action space in routing process. The protocol is suitable both for centralized and decentralized routing, but it lacks in the consideration to adjust the hello interval which leads to routing overhead. Greedy Perimeter Stateless Routing protocol suffers from routing congestion therefore, Traffic-aware Q-learning-based GPSR routing protocol (TQNGPSR) [26] introduced for UAV networks. The protocol estimated each link's quality based on congestion data from nearby nodes in order to make efficient routing decisions. This way, the estimated Q-values will be more accurate when it comes to the actual quality of links during congestion. However, the protocol didn't consider energy constraints and used the fixed learning parameters. Therefore, DQN based OLSR (DQN-OLSR) [27] used the DQN approach to dynamically alter the TC interval, resulting in a correct routing table. This technique used dynamic learning parameters. When a node calculates how long it will take for a neighboring node to move outside of the communication radius, DQN can detect this change in topology with ease. Based on this information, the node determines the shortest duration for the Hello message cycle. However, the protocol only suitable for centralized network.

D. DDPG

Q-Learning and DQN are not suitable for situations with extensive and continuous state and action spaces, while DDPG approaches are well-suited for such scenarios. Topology-aware resilient routing approach based on adaptive Q-learning (TARRAQ) [28] protocol uses the DDPG algorithm to minimize packet loss, latency, and network overhead while taking topology dynamicity into account. UAVs' dynamic behaviors are examined using the queuing method. Nevertheless, the model performs poorly for cluster-based routing and not able to reduce routing overhead. A routing protocol that focuses on lowering network overhead in order to improve network performance, including energy and bandwidth, is suggested by the author, M. S. Ayub et al. [29], as a solution to the problem of high message broadcasts created for route maintenance. The hello interval action was carried out and a new AI-based AI-hello model was developed. Furthermore, for UAV networks, multi-agent independent Deep Deterministic Policy Gradient Optimized link state routing (MA-IDDPG-OLSR) was presented by Y. Zeng in [30]. Because the MA-IDDPG-OLSR focuses on local data rather than centralized decision-making, it provides a robust model for distributed UAV networks that may experience central node failure. Depending on its local state and the nodes nearby, each node autonomously modifies the interval between sending a hello packet and a TC message. They have employed a reward function with a penalty for excessive routing overhead in order to cut down on overhead. As a result, UAV nodes are encouraged to maximize the time between messages sent.

E. GNN

The FANET system's performance is enhanced by a neural network-based unique GNN model that optimizes path selection based on link quality. In device-to-device communication, a unique method for optimizing power control and position dispatch for numerous UAVs was introduced in [31]. The GNN-based model maps the communication system into a graph structure with transmission links represented as vertices and interference links as edges. This makes it possible to represent intricate network interactions effectively. But because the model solely considers stationary UAVs, it might not adequately capture real-world scenarios when UAVs are frequently in motion In order to choose the optimum relay paths and maximize communication and coverage efficiency, two GNN-based techniques are utilized in [32], Relay GNN (RGNN) and Location GNN (LGNN). Whereas the LGNN employed the unsupervised technique to optimize UAV placements, the RGNN used the RL technique to choose relay paths. But there is a lack of optimization of UAV trajectories in this work. A different protocol [33] improved UAV data sharing by using a distributed cooperative data distribution technique based on GNN. Based on local data UAVs can learn a decision policy and train the model with a reward function using the RL method, which makes use of the interaction between network topology. A constraint in high-noise or high-interference environments could be the signal-to-interference-plus-noise ratio (SINR), which is essential to the idea.

V. Conclusion

This research presents a comprehensive review of ML techniques and their applications for FANET routing. It finds that by adequately addressing the dynamic and complex character of aerial networks, ML approaches offer significant improvements over traditional routing methods. These methods can be applied in a variety of situations. The various routing algorithms MDP, Q-learning, DQN, DDPG, and GNNs offer distinct advantages over the others in terms of enhanced decision-making, resource management, and adaptability to quickly changing topologies. In scenarios where the agent may learn from its interactions with the network to optimize routing policies, MDP and Q-learning perform exceptionally well.

Copyright © IEEE–2024 ISBN: 979-8-3503-8058-3

Authorized licensed use limited to: University of Saskatchewan. Downloaded on May 20,2025 at 21:31:22 UTC from IEEE Xplore. Restrictions apply.

Through function approximation, deep learning based DRL technique offers more sophisticated handling of huge state-action spaces and intricate network dynamics. On the other hand, by modeling the interactions and linkages between nodes, the GNN technique works well for route prediction. Overall, ML techniques offer a lot of potential in increasing the scalability, reliability, and efficiency of FANET routing; nevertheless, there are still concerns with computational complexity, data requirements, and convergence. Subsequent investigations ought to concentrate on merging these ML methods with hybrid strategies and investigating real-time implementations to enhance routing efficiency in progressively more dynamic FANET settings.

References

[1] İ. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying Ad-Hoc Networks (FANETs): A survey," Ad Hoc Netw., vol. 11, no. 3, pp. 1254–1270, May 2013, doi: 10.1016/j.adhoc.2012.12.004.

[2] M. A. Khan, A. Safi, I. M. Qureshi, and I. U. Khan, "Flying ad-hoc networks (FANETs): A review of communication architectures, and routing protocols," 2017 First International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT), pp. 1-9, November 2017, doi: 10.1109/intellect.2017.8277614.

[3] M. F. Khan, K.-L. A. Yau, R. M. Noor, and M. A. Imran, "Routing Schemes in FANETs: A Survey," Sensors, vol. 20, no. 1, pp. 38, December 2019, doi: 10.3390/s20010038.

[4] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, Md. H. Tareque, and Md. S. Hossain, "Routing in Flying Ad Hoc Networks: Survey, Constraints, and Future Challenge Perspectives," IEEE Access, vol. 7, pp. 81057–81105, January 2019, doi: 10.1109/access.2019.2923840.

[5] S. Kumar, R. S. Raw, and A. Bansal, "LoCaL: Link-optimized cone-assisted location routing in flying ad hoc networks," Int. J. Commun. Syst., vol. 36, no. 2, October 2022, doi: 10.1002/dac.5375.

[6] S. Kumar, R. S. Raw, A. Bansal, M. A. Mohammed, P. Khuwuthyakorn, and O. Thinnukool, "3D Location Oriented Routing in Flying Ad-Hoc Networks for Information Dissemination," IEEE Access, vol. 9, pp. 137083–137098, January 2021, doi: 10.1109/access.2021.3115000.

[7] S. Kumar, R. S. Raw, A. Bansal, and P. Singh, "UF-GPSR: Modified geographical routing protocol for flying ad-hoc networks," Trans. Emerg. Telecommun., vol. 34, no. 8, June 2023, doi: 10.1002/ett.4813. Available: https://doi.org/10.1002/ett.4813.

[8] H. M. F. Noman, E. Hanafi, K. A. Noordin, K. Dimyati, M. N. Hindia, A. Abdrabou, and F. Qamar, "Machine Learning Empowered Emerging Wireless Networks in 6G: Recent Advancements, Challenges and Future Trends," IEEE Access, vol. 11, pp. 83017–83051, January 2023, doi: 10.1109/access.2023.3302250.

[9] S. Ray, "A Quick Review of ML Algorithms," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), February 2019, doi: 10.1109/comitcon.2019.8862451.

[10] A. Rovira-Sugranes, A. Razi, F. Afghah, and J. Chakareski, "A review of AI-enabled routing protocols for UAV networks: Trends, challenges, and future outlook," Ad Hoc Networks, vol. 130, p. 102790, May 2022, doi: 10.1016/j.adhoc.2022.102790.

[11] S. Padakandla, "A Survey of Reinforcement Learning Algorithms for Dynamically Varying Environments," ACM Comput. Surv., vol. 54, no. 6, pp. 1–25, July 2021, doi: 10.1145/3459991.

[12] W. Qiang and Z. Zhongli, "Reinforcement learning model, algorithms and its application," 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), pp. 1143-1146, August 2011, doi: 10.1109/mec.2011.6025669.

[13] M. M. Alam and S. Moh, "Survey on Q-Learning-Based Position-Aware Routing Protocols in Flying Ad Hoc Networks," Electronics, vol. 11, no. 7, p. 1099, March 2022, doi: 10.3390/electronics11071099.

[14] F. Agostinelli, G. Hocquet, S. Singh, and P. Baldi, "From Reinforcement Learning to Deep Reinforcement Learning: An Overview," in Lect. notes comput. sci., 2018, pp. 298–328. doi: 10.1007/978-3-319-99492-5_13.

[15] A. Zhang, M. Sun, J. Wang, Z. Li, Y. Cheng, and C. Wang, "Deep Reinforcement Learning-Based Multi-Hop State-Aware Routing Strategy for Wireless Sensor Networks," Appl. Sci., vol. 11, no. 10, p. 4436, May 2021, doi: 10.3390/app11104436.

[16] Z. Zheng, A. K. Sangaiah, and T. Wang, "Adaptive Communication Protocols in Flying Ad Hoc Network," IEEE Commun. Mag., vol. 56, no. 1, pp. 136–142, January 2018, doi: 10.1109/mcom.2017.1700323.

[17] J. Liu, Q. Wang, and Y. Xu, "AR-GAIL: Adaptive routing protocol for FANETs using generative adversarial imitation learning," Comput. Netw., vol. 218, p. 109382, December 2022, doi: 10.1016/j.comnet.2022.109382.

[18] P. Mahajan, B. P, A. Kumar, G. S. S. Chalapathi, V. Chamola, and M. Khabbaz, "Multi-Objective MDP-based Routing In UAV Networks For Search-based Operations," IEEE Trans. Veh. Technol., pp. 1–14, January 2024, doi: 10.1109/tvt.2024.3395840.

[19] C. Li and J. Liu, "A Modified Multi-Agent Reinforcement Learning Protocol Based on Prediction for UAANETs," 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), 2020, pp. 1-5, doi: 10.1109/VTC2020-Fall49728.2020.9348732.

[20] J. W. Lim and Y. B. Ko, "Q-learning based Stepwise Routing Protocol for Multi-UAV Networks," 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), April 2021, doi: 10.1109/icaiic51459.2021.9415265.

[21] X. Qiu, Y. Xie, Y. Wang, L. Ye, and Y. Yang, "QLGR: A Q-learning-based Geographic FANET Routing Algorithm Based on Multi-agent Reinforcement Learning," KSII T. Internet Info., vol. 15, no. 11, November 2021, doi: 10.3837/tiis.2021.11.020.

[22] Q. Xue, Y. Yang, J. Yang, X. Tan, J. Sun, G. Li, and Y. Chen, "QEHLR: A Q-Learning Empowered Highly Dynamic and Latency-Aware Routing Algorithm for Flying Ad-Hoc Networks," Drones, vol. 7, no. 7, p. 459, July 2023, doi: 10.3390/drones7070459.

[23] M. Hosseinzadeh, J. Tanveer, A. M. Rahmani, K. Aurangzeb, E. Yousefpoor, M. S. Yousefpoor, A. Darwesh, S.W. Lee, and M. Fazlali, "A Q-learning-based smart clustering routing method in flying Ad Hoc networks," J. King Saud Univ.-Comp. & Info., vol. 36, no. 1, p. 101894, January 2024, doi: 10.1016/j.jksuci.2023.101894.

[24] M. F. Khan, K. L. A. Yau, M. H. Ling, M. A. Imran, and Y. W. Chong, "An Intelligent Cluster-Based Routing Scheme in 5G Flying Ad Hoc Networks," Appl. Sci., vol. 12, no. 7, p. 3665, April 2022, doi: 10.3390/app12073665.

[25] Y.N. Chen, N. Q. Lyu, G. H. Song, B. W. Yang, and X. H. Jiang, "A traffic-aware Q-network enhanced routing protocol based on GPSR for unmanned aerial vehicle ad-hoc networks," Front. Inf. Technol. Electron. Eng., vol. 21, no. 9, pp. 1308–1320, September 2020, doi: 10.1631/fitee.1900401.

[26] Y. Zeng, J. Zhou, Y. Liu, T. Chao, D. Yang, Y. Liu, and X. Shi, "Stable routing protocol for unmanned aerial vehicle ad-hoc networks based on DQN-OLSR," IET Commun., vol. 17, no. 1, pp. 73–85, October 2022, doi: 10.1049/cmu2.12510.

[27] Y. Cui, Q. Zhang, Z. Feng, Z. Wei, C. Shi, and H. Yang, "Topology-Aware Resilient Routing Protocol for FANETs: An Adaptive Q-Learning Approach," IEEE Internet Things J., vol. 9, no. 19, pp. 18632–18649, October 2022, doi: 10.1109/jiot.2022.3162849.

[28] M. S. Ayub, P. Adasme, D. C. Melgarejo, R. L. Rosa, and D. Z. Rodriguez, "Intelligent Hello Dissemination Model for FANET Routing Protocols," IEEE Access, vol. 10, pp. 46513–46525, January 2022, doi: 10.1109/access.2022.3170066.

[29] Y. Zeng, J. Zhou, Y. Liu, T. Chao, D. Yang, Y. Liu, and X. Shi, "Distributed MA-IDDPG-OLSR based stable routing protocol for unmanned aerial vehicle ad-hoc network," IET Commun., vol. 18, no. 8, pp. 503–522, April 2024, doi: 10.1049/cmu2.12757.

[30] S. He, S. Xiong, Y. Ou, J. Zhang, J. Wang, Y. Huang, and Y. Zhang, "An Overview on the Application of Graph Neural Networks in Wireless Networks," IEEE open j. Commun. Soc., vol. 2, pp. 2547–2565, January 2021, doi: 10.1109/ojcoms.2021.3128637.

[31] P. Li, L. Wang, W. Wu, F. Zhou, B. Wang, and Q. Wu, "Graph neural network-based scheduling for multi-UAV-enabled communications in D2D networks," Digit. Commun. Netw., vol. 10, no. 1, pp. 45–52, February 2024, doi: 10.1016/j.dcan.2022.05.014.

[32] X. Wang, L. Fu, N. Cheng, R. Sun, T. Luan, W. Quan, and K. Aldubaikhy, "Joint Flying Relay Location and Routing Optimization for 6G UAV–IoT Networks: A Graph Neural Network-Based Approach," Remote Sens., vol. 14, no. 17, p. 4377, September 2022, doi: 10.3390/rs14174377.

[33] N. Xing, Y. Zhang, Y. Wang, and Y. Zhou, "Unmanned Aerial Vehicle Cooperative Data Dissemination Based on Graph Neural Networks," Sensors, vol. 24, no. 3, p. 887, January 2024, doi: 10.3390/s24030887.

Authorized licensed use limited to: University of Saskatchewan. Downloaded on May 20,2025 at 21:31:22 UTC from IEEE Xplore. Restrictions apply.