Hindawi
Mobile Information Systems
Volume 2022, Article ID 4560072, 18 pages
https://doi.org/10.1155/2022/4560072

Review Article

Comparative Analysis of Routing Schemes Based on Machine Learning

Shaoyu Yang,[1] Cong Tan,[2] Dag Øivind Madsen,[3] Haige Xiang,[4] Yun Li,[5] Imran Khan,[6] and Bong Jun Choi[7]

[1] *North China University of Water Resources and Electric Power, Zhengzhou 450046, China*
[2] *Henan University of Animal Husbandry and Economy, Zhengzhou 450046, China*
[3] *University of South-Eastern Norway, Bredalsveien 14, Hønefoss 3511, Norway*
[4] *Information Research Center, Beijing University, Beijing, China*
[5] *Department of Electronic Engineering, University of Illinois Urbana-Champaign, Champaign, USA*
[6] *Department of Electrical Engineering, University of Engineering and Technology, Peshawar 814, Pakistan*
[7] *School of Computer Science and Engineering, Soongsil University, Seoul, Republic of Korea*

Correspondence should be addressed to Shaoyu Yang; ysymagnet@126.com, Dag Øivind Madsen; dag.oivindmadsen@outlook.com, and Bong Jun Choi; davidchoi@soongsil.ac.kr

Received 25 April 2022; Revised 17 May 2022; Accepted 21 May 2022; Published 30 June 2022

Academic Editor: Yanyi Rao

Machine learning-based distributed routing algorithms, in contrast to traditional mathematical model-driven distributed routing algorithms, are typically data-driven, allowing them to adapt to dynamically changing network environments and various performance evaluation index optimization requirements. It is quite likely that it will become a key part of the next-generation Internet in the future. However, current intelligent routing research is still in its early stages. This article provides a comprehensive review of the state-of-the-art routing algorithms based on machine learning. First, important research on existing data-driven intelligent routing algorithms is presented with the key concepts and applications of these systems demonstrated. To enable intelligent routing algorithms to be deployed in real scenarios with cheap cost and high reliability, two appropriate training deployment frameworks and intelligent routing algorithm training and deployment strategies are given. Finally, the future development of machine learning-based intelligent routing systems is examined. The opportunities and problems that have been encountered, as well as prospective research directions, are discussed.

1. Introduction

In recent years, with the rapid development of the Internet, many emerging applications including industrial Internet, 4K + video and holographic communication, online games, and remote cloud services have emerged in large numbers. These emerging network applications bring highly differentiated service quality requirements. However, in the past, the method of improving network service quality by simply increasing the speed and capacity of equipment has gradually reached the ceiling and further improving the performance requires a high cost. Therefore, better optimization and utilization of existing network resources has become an important way to improve user service experience.

In a classical computer network design, the network layer uses best-effort packet forwarding, and the routing algorithm's focus is on the data packet's reachability, as well as the algorithm's performance and scalability. In recent years, with the rapid development of computer networks, the scale of the network has drastically increased, and the number of application service types on the upper layer of the network has also increased rapidly. As the number of service kinds grows, so do the goals for service performance improvement, which include latency, bandwidth, throughput, packet loss rate, and network stability. The traditional best-effort routing algorithm makes the existing computer network architecture have certain limitations in optimizing these performance evaluation indicators. Figure 1 shows an example of the limitations of the traditional routing algorithm. In this example, the network flow load requires a bandwidth of 500 Mbps. The traditional shortest path-based routing algorithm directs all traffic to the bottleneck link, and the selected path has an available bandwidth (100 Mbps), which is much smaller than the service demand bandwidth. This not only will greatly reduce the user experience, but also may bring serious network congestion and cause a huge waste of network resources. Appropriate routing and offloading of the above traffic can well avoid the problem in this example. However, because the available bandwidth of the path changes dynamically with time in the real network environment, it is difficult for traditional routing algorithms to accurately perceive the current network status and perform appropriate actions accordingly.

In addition, the emergence of emerging network application scenarios such as data center networks has brought new challenges to the field of routing optimization and traffic engineering [1]. Compared with the traditional network, the network bandwidth of the data center is larger, and there are larger flows and long flows at the same time, and the demand and difficulty for traffic scheduling are also higher. Although there have been some routing and traffic engineering methods to try to solve the network optimization problem in various data center scenarios, in the data center network scenario, the existing routing and traffic scheduling optimization methods are still difficult to meet the requirements of efficient utilization of links and loads [2]. In order to meet complex network application scenarios and diverse service quality requirements, many network layer optimization schemes based on mathematical models have been proposed [3–6]. These routing optimization or traffic engineering schemes usually make some assumptions for the application scenario to simplify the problem, so that the optimization problem can be efficiently solved by using the existing mathematical methods. However, real network application scenarios are often difficult to fully meet these idealized assumptions, which makes routing optimization algorithms based on mathematical models unable to guarantee their deployment effects in real scenarios. In fact, many routing optimization problems can be solved even under hypothetically simplified scenarios.

It is still very complex, and there is no general model that can solve different types of routing optimization problems at the same time [7]. Since traditional routing optimization tasks need to be modeled separately for each specific scenario and specific optimization objectives, deploying these methods in a real network environment may have an impact on the scalability of network facilities. Therefore, traditional mathematical models have still difficulty to deploy large-scale routing optimization schemes in practical scenarios.

In recent years, artificial intelligence (AI) technology based on deep learning has developed rapidly and has been widely used in natural language processing [8], image recognition [9], game strategy calculation [10], and other fields. The research on deep learning models and the development of computer hardware such as central processing unit (CPU) and graphical processing unit (GPU) have made the strategies that can be learned by AI models more complex, and the training and execution efficiency is getting higher. The improvement of equipment computing power and model expression ability makes the AI model have strong learning ability and good generalization. It is gradually possible to use AI model to solve routing optimization problems and to endow the network layer with intelligence. Compared with the traditional model-driven routing optimization algorithm, the data-driven intelligent routing optimization algorithm has three advantages:

(1) *Accuracy*. Using real data to train machine learning algorithm models does not require complex assumptions and modeling of the network environment;

(2) *Efficiency*. In polynomial time, the optimized routing decision can be obtained by fast reasoning according to the input data;

(3) *Universality*. The same machine learning model can be used to solve different.

The above three advantages make the data-driven intelligent routing method better adapt to different network application scenarios and routing optimization goals than the traditional routing method and have better scalability in the process of deployment.

In addition to the rapid development of AI technology, the related research on software-defined networking (SDN) [11] and programmable routing devices [12, 13] that have emerged in recent years also provides the possibility of deploying intelligent routing algorithms. These works enable the routing layer to complete more complex tasks. The emergence of the SDN architecture enables the intelligent routing algorithm based on machine learning to run as an application in the SDN server with powerful computing power and effectively control the routing and traffic [14]. However, the existing research on the intelligent routing scheme based on machine learning is still in a relatively preliminary stage as it mainly focuses on the correctness and convergence of the intelligent routing algorithm. The training and deployment scheme of the intelligent routing algorithm in the real scenario is still not perfect. In addition, the computing power of current routing equipment is still far from the large-scale deployment of intelligent routing algorithms [15].

In order to provide a detailed evaluation of the relevant state-of-the-art machine learning-based routing methods, this article proposed a comparative study and has the following contributions:

(i) Introduces the related work of existing data-driven intelligent routing algorithms based on machine

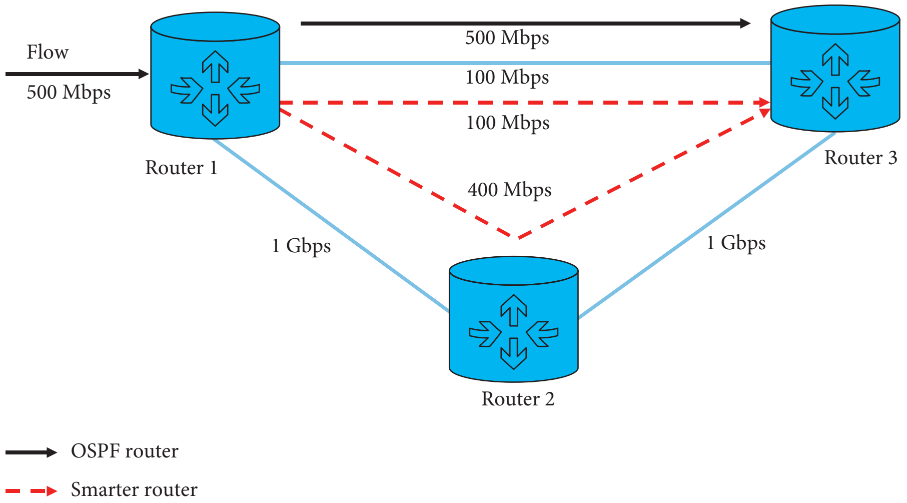

Figure 1: Illustration of flow decision in the open shortest path first algorithm.

learning from the perspectives of methods and application scenarios, and analyzes the advantages and disadvantages of different intelligent routing methods.

(ii) Further analyzes the training and deployment methods of existing intelligent routing algorithms.

(iii) Two intelligent routing algorithm training and deployment frameworks suitable for different application scenarios are proposed.

(iv) Analyzes the opportunities and challenges in the future development of intelligent routing algorithms based on machine learning, and gives the future research directions of intelligent routing algorithms.

2. Overview of Intelligent Routing Algorithms

The authors devised an intelligent routing algorithm Q-routing based on Q-learning and deployed in communication networks as early as 1994 [16]. The experiments demonstrated that the Q-routing strategy can efficiently avoid the network congestion and minimize the packet transmission time when compared to standard shortest path routing. However, although many subsequent works have perfected and optimized the method [17, 18], limited by the computing power of the router and the design of the network layer structure, the intelligent routing algorithm is difficult to be deployed in real network scenarios.

Reference [19] proposed the Q-learning-based energy-efficient and lifetime-aware routing (QELAR) method, which applied the idea of Q-Learning to wireless sensor networks (WSN) to optimize the energy consumption and lifetime of wireless sensor networks. Compared with the traditional network, the WSN is located in a complex and changeable environment and the demand for routing service quality is diverse. The traditional routing algorithm is often difficult to achieve satisfactory results in this application scenario. In addition, the structure of WSN is relatively independent compared with the traditional network, so the deployment of the intelligent routing method based on Q-learning is less difficult. Subsequent literature [20, 21] further applied the Q-learning method to the reliable transmission and accelerated forwarding of WSNs and achieved good results.

Deep learning has made great progress in the network area in recent years. It has been employed in transport layer congestion management [22], network vulnerability detection [23], video streaming optimization [24], and other domains. It has also gotten increased attention for solving routing optimization issues, and certain routing algorithms based on deep learning and deep reinforcement learning have been developed [25]. These intelligent routing algorithms not only use deep learning to improve the traditional routing algorithms [26], but also optimize the global performance for new network application scenarios such as data center network traffic scheduling and backbone network traffic engineering in recent years.

As more intelligent routing algorithms are proposed, how to deploy data-driven intelligent routing algorithms in real environments has also become a problem that has attracted much attention. The work of reference explored the prospect of deploying deep learning-based intelligent routing algorithms in real-world scenarios and proposed a way to deploy deep learning-based intelligent routing using a software-defined router (SDR) equipped with GPUs algorithm framework assumptions. However, according to our research, the existing research work still does not provide a feasible solution to deploy the intelligent routing algorithm in the existing computer network architecture.

According to the types of machine learning methods used, the data-driven intelligent routing algorithms are largely separated into intelligent routing algorithms based on supervised learning and reinforcement learning in recent years.

3. Supervised Learning-Based Routing Schemes

3.1. An Overview of Supervised Learning Methods Applied in Intelligent Routing. Supervised learning refers to the use of known input and output samples to train a model, so that the model can accurately complete a class of machine learning tasks from input to output mapping [27, 28]. In recent years, intelligent routing systems based on supervised learning have mostly relied on deep learning models. Compared with traditional supervised learning methods, deep learning models can learn more complex strategies through labeled data, which provides the possibility to implement intelligent routing methods in complex network environments. In this section, we will briefly introduce the deep learning methods commonly used in existing intelligent routing methods.

The most common deep learning model is deep neural network (DNN), whose model design simulates the working principle of biological neurons, and the working process includes the feedforward process and the feedback process. Figure 2 shows its model structure and working process. In the feedforward process of DNN, the model passes the input vector forward layer by layer by combining linear weighting and activation function, and finally realizes the mapping from input to output. In the feedback process of DNN, the model transmits the deviation between the actual output result and the expected result in reverse layer by layer to complete the adjustment process of model parameters and achieve the effect of automatic learning. As an improvement to the DNN model, Ref. [29] proposed deep belief network (DBN). The DBN model combines the traditional DNN model with the restricted Boltzmann machine (RBM). The training process can be regarded as using the RBM to initialize the parameters of the DBN model and using the gradient reverse transfer process to fine-tune the parameters of the DBN model according to the task. As a basic deep learning model, the DBN model can be used in various tasks including routing optimization.

In the intelligent routing scheme, it is often necessary to process serialized information with variable dimensions, such as path information extraction [30] and traffic prediction at the next moment based on past traffic information [31]. In these tasks, it is difficult to achieve the desired effect only through the DNN model, and the recurrent neural network (RNN) is often used. The RNN can handle serialized input of indeterminate length well and has a good guarantee for the timing of network traffic information and the ordering of path features. Figure 3 shows the model structure of the RNN network. As an improvement of the RNN model, the long short-term memory unit (LSTM) [32] and the gated recurrent unit (GRU) [33] has better performance in existing works and is widely used.

In the intelligent routing scheme, the local or global topology information of the current network is an important basis for completing the intelligent routing decisions. However, due to the dynamic variability of the network topology, the traditional deep learning models are often difficult to handle this part of the information efficiently. The graph neural network (GNN) is a new type of neural network structure proposed in recent years, which is considered to be able to effectively deal with the problem of topological information extraction [34]. The GNN model vectorizes the characteristics of network nodes and edges, and performs several rounds of iterations. During each iteration, the vectorized representations of these nodes and edges are updated according to the topological dependencies using an update function based on the deep learning model. Finally, the vectorized representation of these nodes and edges will converge to a certain value, which means that the GNN model has transformed the topology information into vectorized representation information that can be used by the deep learning model. Studies have shown that the GNN model has good scalability and generalization, and has been widely used in network topology information extraction tasks [35].

3.2. Intelligent Routing Algorithm Based on Deep Learning.

The most direct application of deep learning in routing optimization problem is to use deep learning model to replace the original routing algorithm based on a mathematical model. A general routing solution model is shown in Figure 4, which takes the network topology and network state information as input, and the deep learning model makes appropriate routing decisions according to the current network environment state according to the input information.

The authors in Ref. [15] proposed a routing decision scheme based on the DBN. Figure 5 shows the schematic diagram of the overall model of the scheme. The application scenario of this intelligent routing scheme is the backbone network. The scheme divides the routers into intradomain routers and border routers. When the data packets enter the backbone network through the border routers, the DBN model deployed on the border routers will calculate the data packets in the backbone network according to the current traffic status of each node in the network. The data packet is forwarded to the destination router through the intra-domain router and finally leaves the backbone network. In the above model, the interdomain routers are only responsible for route forwarding and network state information collection, thus avoiding the frequent exchange of network topology information in traditional distributed routing algorithms. The routing decision model of this scheme trains a DBN model separately for each routing node to each destination border router to output the appropriate next-hop node according to the network state information. The routing path calculation process adopts a hop-by-hop method to pass the corresponding DBN model generation. The work of Ref. [15] shows that the routing strategy based on the deep learning model can achieve 95% accuracy. At the same time, the deep learning model has the characteristics of making routing decisions based on part of the network state characteristics, which also makes the intelligent routing method based on deep learning. Compared with traditional routing methods, it has lower information exchange cost and faster routing convergence speed when the network environment changes. However, the deployment of the above scheme not only requires the backbone network routers to have strong model computing capabilities, but also needs to modify the existing routing protocols. Therefore, deploying the above scheme under the existing computer network architecture requires extremely high costs and will seriously affect the scalability of the network.

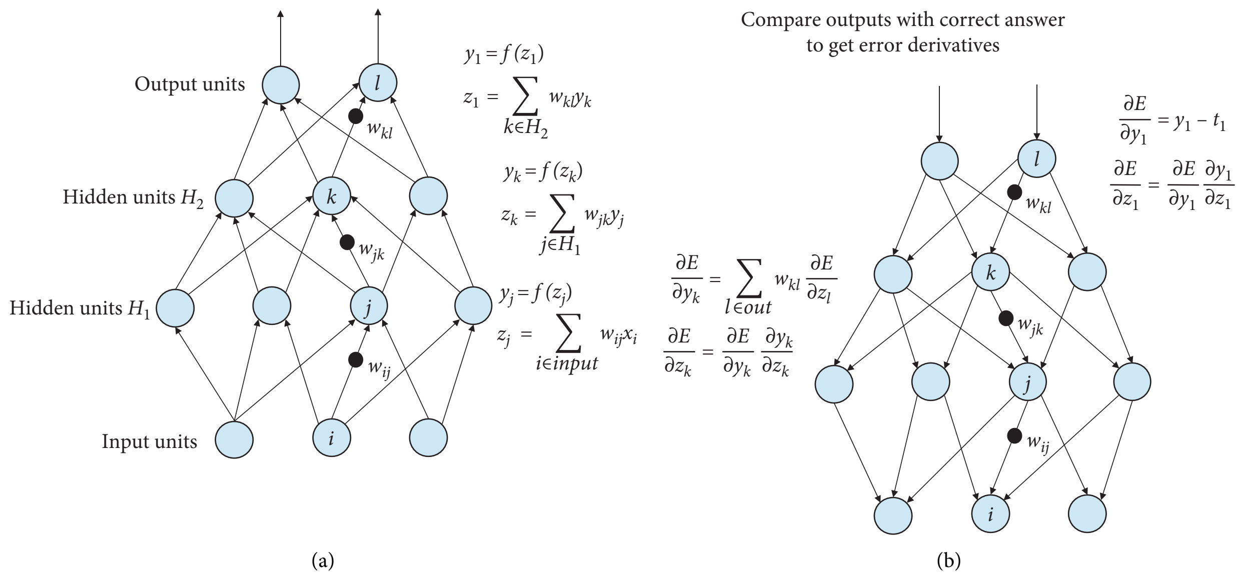
Output units
Hidden units H_2
Hidden units H_1
Input units
l
k
j
i
w_{kl}
w_{jk}
w_{ij}
$y_1 = f(z_1)$
$z_1 = \sum_{k \in H_2} w_{kl} y_k$
$y_k = f(z_k)$
$z_k = \sum_{j \in H_1} w_{jk} y_j$
$y_j = f(z_j)$
$z_j = \sum_{i \in input} w_{ij} x_i$
(a)
Compare outputs with correct answer to get error derivatives
$\frac{\partial E}{\partial y_1} = y_1 - t_1$
$\frac{\partial E}{\partial z_1} = \frac{\partial E}{\partial y_1} \frac{\partial y_1}{\partial z_1}$
$\frac{\partial E}{\partial y_k} = \sum_{l \in out} w_{kl} \frac{\partial E}{\partial z_l}$
$\frac{\partial E}{\partial z_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial z_k}$
(b)

Figure 2: Illustration of DNN. (a) Feed forward, (b) Back propagation.

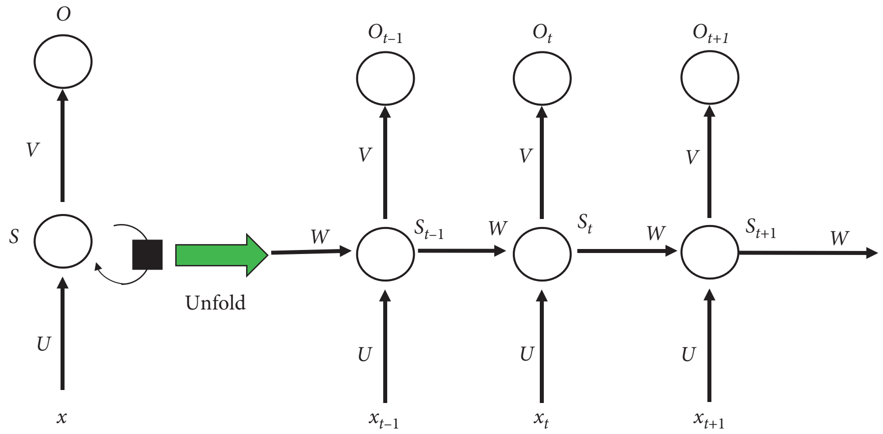

Figure 3: Process of RNN.

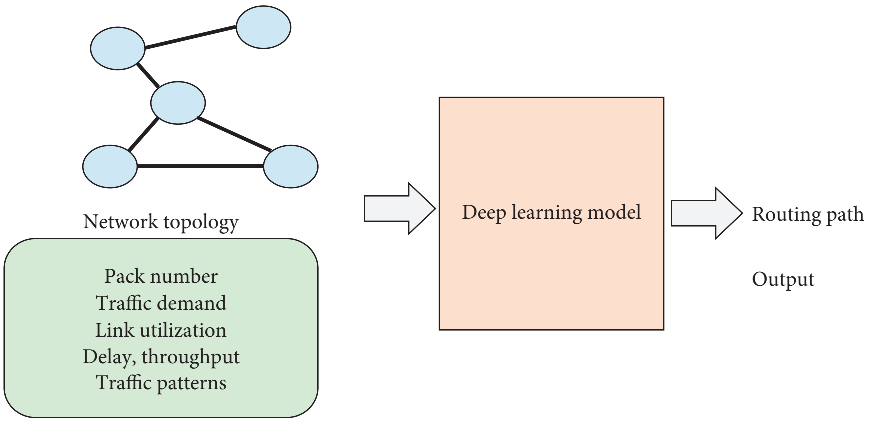

Figure 4: DL model for routing.

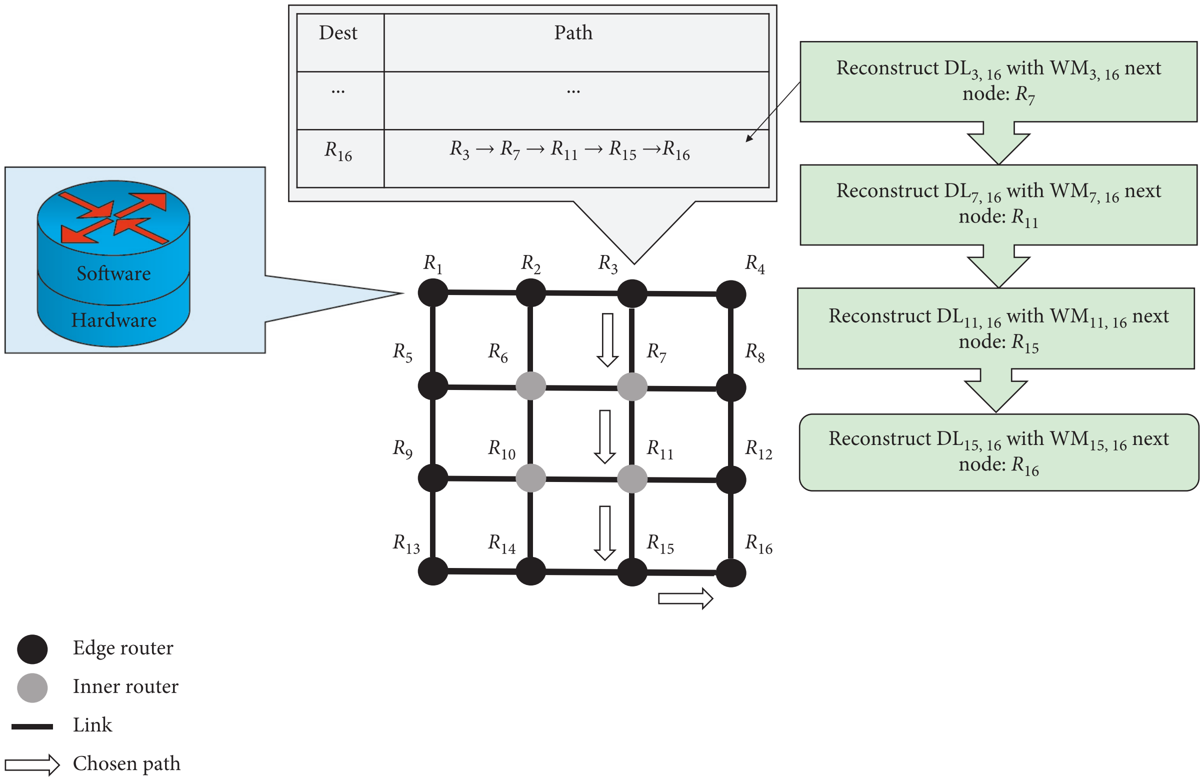
Dest
Path
...
...
R_{16}
$R_3 \rightarrow R_7 \rightarrow R_{11} \rightarrow R_{15} \rightarrow R_{16}$
Reconstruct $\mathrm{DL}_{3,16}$ with $\mathrm{WM}_{3,16}$ next node: R_7
Reconstruct $\mathrm{DL}_{7,16}$ with $\mathrm{WM}_{7,16}$ next node: R_{11}
Reconstruct $\mathrm{DL}_{11,16}$ with $\mathrm{WM}_{11,16}$ next node: R_{15}
Reconstruct $\mathrm{DL}_{15,16}$ with $\mathrm{WM}_{15,16}$ next node: R_{16}
Software
Hardware
R_1 R_2 R_3 R_4
R_5 R_6 R_7 R_8
R_9 R_{10} R_{11} R_{12}
R_{13} R_{14} R_{15} R_{16}
Edge router
Inner router
Link
Chosen path

FIGURE 5: Protocol flow process of the deep belief network.

In addition to the DBN model, other deep learning models have also been tried to apply to intelligent routing tasks. The work of Ref. [36] compared the effect of applying different deep learning models to learn routing decisions. In this work, the hop-by-hop intelligent routing decision-making process is formally expressed as

$$\mathrm{src}, \mathrm{dst}_{n+1} = F(\mathrm{src}, \mathrm{dst}_n, \mathrm{dst}, G). \tag{1}$$

Among them, src and dst represent the source and destination nodes, respectively, and $\mathrm{src}, \mathrm{dst}_n$ are the nth routing node numbers in the route from src to dst; $F(\cdot)$ is the routing decision function; G stands for topology information. Through experiments, it is found that the combination of the topology-based feature extraction method and the graph-aware deep learning (GADL) model can effectively improve the model test accuracy and reduce the model training time compared with the existing deep learning models such as DBN and CNN.

To further utilize the topology information, Shin and Kim [26] designed a distributed intelligent routing algorithm based on GRU and GNN. In order to make the GNN model better represent the structural characteristics of the routing network and make the network feature information modeled by the GNN more convenient for the routing decision-making process, the router interface is added to the graph model as an additional node. Figure 6 shows the schematic diagram of the graph model after adding the router interface as an additional node. After GNN completes the topology modeling, the node information corresponding to each router interface is vectorized and represented by $\mathbf{h}_v$. It not only contains its own information, but also contains the entire network structure and state information required for routing decisions due to the information transfer characteristics of GNN. Using the routing interface information $\mathbf{h}_v$, each router can locally calculate the router interface that should pass through to the corresponding destination node. Due to the model characteristics of GNN, the iterative process of the above GNN topology modeling can be done in a distributed manner by deploying the GNN parameter update function on each router, so this method naturally has good scalability and distributed routing decisions. The simulation experiments of this work show that the distributed intelligent routing algorithm based on GNN performs well in terms of routing convergence speed, accuracy, robustness, and fault adaptability. The accuracy rate of 98% is achieved within 15 rounds of iterations for the max-min fair routing algorithm [37].

Combining with the content in Table 1, it can be found that the existing intelligent routing schemes based on deep learning models mainly generate routing paths in a hop-by-hop manner. Another routing mode corresponding to the hop-by-hop routing generation method is to calculate all possible paths in advance and select the appropriate path according to the network state through the deep learning model. This method based on path selection can avoid routing loops caused by the path generation model. However, the number of optional paths in the network will increase exponentially with the increase in the network size, and its huge output dimension makes the learning difficulty of the deep learning model based on path selection and the number of model parameters in an unbearable order of

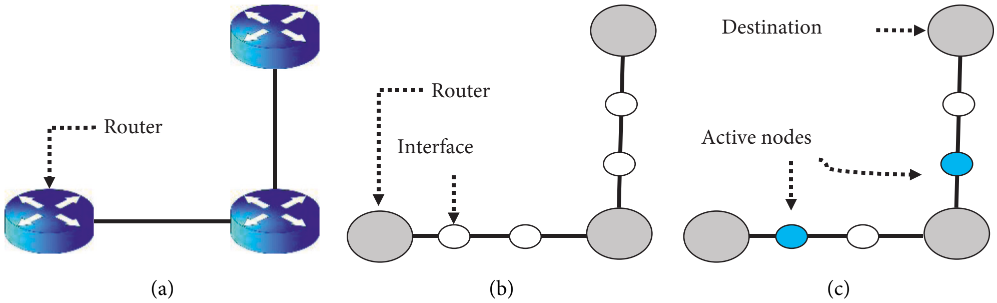

Figure 6: Inclusion of the router. (a) Example of network topology, (b) Associated graph model, (c) Example of output features.

Table 1: Comparison of various routing protocols based on machine learning.

Algorithm	Routing method	Controlling method	Learning method	Deployment	Training method
Ref. [15, 25, 39]	Path creation	Packet-based	Offline	Decentralized	Offline
Ref. [26]	Path creation				
Ref. [40]	Predicting congestion				
Ref. [30]	Predicting jitter and delay			Centralized	
Ref. [36]	Path creation	Flow-based		Centralized	
Ref. [16–21]	Path creation	Packet-based	Online	Decentralized	Online
Ref. [14]	Splitting ratio configuration	Epoch-based		Centralized	
Ref. [31]	Link weights	Epoch-based	Offline	Centralized	Offline
Ref. [38]					

magnitude [38]. In addition, due to the strong correlation between network path characteristics and topology, it is difficult for deep learning models based on path selection to have sufficient generality and generalization. Compared with the path selection method, the hop-by-hop path generation method can significantly reduce the output dimension and the difficulty of model decision-making, which can significantly improve the accuracy of routing decisions [39].

Existing work shows that the intelligent routing algorithm based on deep learning can quickly and accurately calculate the corresponding routing decision based on some network state information, and it shows certain advantages compared with traditional distributed routing in terms of information transmission cost and routing convergence speed. The distributed routing decision based on GNN has made some progress in the problems of topology information modeling, robustness, and fault adaptability that are difficult to be solved by intelligent routing solutions based on traditional deep learning models. However, the existing intelligent routing algorithm based on the deep learning model mainly learns the routing algorithm based on the shortest path, and whether it can learn more complex dynamic routing algorithms well is worth further discussion. In addition, the existing deep learning-based intelligent routing algorithms cannot guarantee their security and robustness in complex and changeable network environments, and require high deployment costs. Therefore, deep learning-based routing algorithms want to replace traditional routing algorithms and still a long way to go.

3.3. Utilize Intelligent Modules to Assist Routing Calculation.

Existing deep learning methods have achieved certain results in network modeling, traffic prediction, and congestion detection [41, 42]. Using the results of deep learning methods in these fields to assist routing calculation is to make routing algorithms more efficient. In routing optimization problems, traditional model-based optimization or heuristic methods often need to involve modules such as network environment modeling, traffic prediction, and congestion detection. Using deep learning methods to replace these modules sometimes achieves better results.

The work of Ref. [40] used a deep neural network predictor based on multitask learning to predict link congestion for each link based on the link historical state data and compared the predicted results with rule-based congestion avoidance and replay. The combination of routing schemes enables routing methods to actively adjust routing before congestion occurs, rather than passively make up for it after it occurs.

The authors in Ref. [30] combined GNN and LSTM model and used a deep learning model based on graph neural network to build the relationship between routing path delay and delay jitter and network topology, traffic matrix, and routing path model, and used the established model to assist the heuristic routing optimization algorithm to calculate the routing strategy. The research results show that the network modeling based on GNN can accurately predict the routing path delay and delay jitter according to the input information and shows good generalization for the topology that does not appear in the training and the dynamically changing routing path. The data-driven network modeling method provides an accurate and efficient routing strategy test environment for the exploration-based heuristic routing optimization algorithm, which enables the heuristic routing optimization algorithm to complete the routing optimization solution process at low cost, while avoiding the need for network optimization. The loss of routing strategy effect was caused by modeling and real environment.

The scheme of using the deep learning model to assist the traditional routing algorithm can effectively improve the performance of the traditional routing optimization algorithm, and at the same time, the traditional routing optimization algorithm ensures that the intelligent routing scheme has stronger reliability and interpretability. Therefore, combining traditional routing optimization algorithms with deep learning models may be a way to develop intelligent routing algorithms in the future.

4. Intelligent Routing Algorithm Based on Reinforcement Learning

4.1. Overview of Reinforcement Learning Methods Applied in Intelligent Routing. A standard reinforcement learning process can be viewed as a process in which a reinforcement learning unit interacts with the environment in discrete time steps. At each time point t, the reinforcement learning unit takes an action at according to the state s_t and receives a feedback reward r_t. The goal of reinforcement learning is to find a policy $\pi(s)$, the policy function is a mapping from state to action and can maximize the decreasing reward, and $\sum_{t=0}^{T} \gamma^t r_t, \gamma \in [0,1]$ is the reward discount factor.

The Q-learning method uses a Q-function to predict the maximum decreasing reward sum corresponding to the state s_t and the action at observed at time t. The Q-function is defined as

$$Q(s_t, a_t) = \overset{\max}{\pi} \{E[R_t | s_t, a_t, \pi]\}. \tag{2}$$

For the calculation of the Q-function, there are two methods: model-based and model-independent. The model-based method directly solves the Q-function through the correlation model between the states in the Markov decision-making process, which is formally expressed as

$$\begin{aligned} Q(s_t, a_t) &= r_t + \gamma \sum_{s_{t+1} \in S} P_{s_t s_{t+1}}^{a_t} V(s_{t+1}), \\ V(s) &= \overset{\max}{a} \{Q(s,a)\} \frac{1}{2}. \end{aligned} \tag{3}$$

Among them, the V function is the state value function, which represents the maximum decreasing reward sum that can be obtained in the corresponding state, and $P_{s_t s_{t+1}}^{a_t}$ represents the state transition probability of the reinforcement learning task corresponding to the Markov decision process. In reinforcement learning tasks, the state transition probability is not always easy to obtain, and the state-independent method can be used to estimate the Q-function:

$$Q(s_t, a_t) = (1-\alpha) Q(s_t, a_t) + \alpha [r_t + \gamma V(s_{t+1})], \tag{4}$$

where α is the model learning rate. Compared with the model-based Q-function calculation method, the model-independent Q-function calculation method usually requires a longer convergence time.

In the traditional Q-learning method, the Q-function is a mapping from the finite state decision space $S \times A$ to the real number space R. In order to deal with the reinforcement learning problem on the continuous high-dimensional state decision space, researchers introduce the deep learning model into the reinforcement learning framework, a variety of deep reinforcement learning (DRL) models have been designed.

The Google DeepMind institute proposed deep Q-learning (DQN) [43]. It uses a deep neural network (DNN) instead of the original Q-value table to approximate the Q-function and trains it through the squared error:

$$L(\theta^Q) = E\left[\left(y_t - Q(s_t, a_t | \theta^Q)\right)^2\right], \tag{5}$$

where θ^Q is the parameter of DQN and $(y_t$ is the target value, which can be calculated as

$$y_t = r_t + \gamma Q(s_{t+1}, \pi(s_{t+1}) | \theta^Q), \tag{6}$$

where $\pi(\cdot)$ is a policy function that can maximize the expected total return, and a commonly used asynchronous strategy is to choose actions in a greedy way:

$$\pi(s_t) = \overset{\text{argmax}}{a_t} Q(s_t, a_t). \tag{7}$$

Corresponding to the DQN method based on Q-function estimation is the policy gradient method [44]. The policy gradient method uses the deep learning model as the policy function $\pi_\theta(s,a)$ and directly optimizes the policy function by calculating the policy gradient.

In order to further improve the performance of the policy gradient method and accelerate the convergence speed of the reinforcement learning model, we can combine the Q-value learning and the policy gradient method, and use the value estimation function to predict the value that will be obtained after the action is taken in the current state, and use the prediction result. The policy model is trained, which is the Actor Evaluator (AC) framework for reinforcement learning.

A commonly used AC framework based on online strategy uses an action advantage function $A(s,a)$ to estimate the advantages and disadvantages of the strategy, and the policy gradient after introducing the advantage function is

$$\nabla J(\theta) = E_{\tau \sim p\theta(\tau)} [\nabla_\theta \log \pi_\theta(s,a) A(s,a)], \tag{8}$$

where τ represents the state-action tuple (s_t, a_t).

The reinforcement learning method based on the online strategy needs to synchronize the training process with the data collection and achieve parameter convergence through the iterative process of updating the parameters for multiple rounds of data collection. In order to decouple the data collection and model training process, an offline reinforcement learning method, a commonly used offline policy-based AC framework deep reinforcement learning model is deterministic policy gradient (DPG) [45]. The method directly uses the value network gradient backhaul to calculate the policy gradient and has achieved good results in the continuous action space reinforcement learning problem. An improved version of this method, deep deterministic policy gradient (DDPG) [46], has been widely used to solve routing optimization problems in continuous action spaces.

In recent year researches, in order to solve the problem of excessive policy update in the trust region policy optimization (TRPO) algorithm, extensive literature has been proposed [47]. Although the second-order method has better convergence guarantee than the first-order method, its high computational complexity limits its application scenarios. Based on the idea of TRPO, OpenAI and DeepMind proposed a proximal policy optimization (PPO) [48] algorithm, which combines the efficiency and ease of implementation of traditional first-order methods and the data efficiency and reliable performance of confidence region algorithms. It is one of the current mainstream reinforcement learning algorithms.

4.2. Intelligent Routing Algorithm Based on Q-Learning. The authors of Ref. [16] proposed Q-routing and for the first time applied Q-learning in routing algorithms. The Q-routing uses the Markov decision process (MDP) to represent the routing forwarding process, treating each routing node as a state in the MDP, the neighbor node picked by the routing next hop as the MDP action, and the routing node selected by each hop as the MDP action. The feedback value acquired by reinforcement learning an action is the time delay. In Q-routing, the Q-value function $Q_x(d, y)$ is used to predict the time it takes to use the next hop node y from the current node x to the target node d. Whenever node x sends a packet to neighbor node y, node y will immediately return the estimated remaining distance delay t to x, which is expressed as

$$t = \min_{z \in \text{neighbors of } y} Q_y(d, z). \tag{9}$$

At this time, using the model-based Q-Learning method, node x can dynamically update its corresponding Q-value function information, formally:

$$\Delta Q_x(d, y) = \eta\left(q + s + t - Q_x(d, y)\right), \tag{10}$$

where η is the learning rate of the algorithm, and q and s are the queue delay and transmission delay from x to y, respectively. According to the dynamically updated Q-value function, for each data packet, Q-routing can adapt to the dynamically changing network state and choose the routing path with the minimum latency. In contrast to the typical shortest path routing method, Q-routing measures the length of the path using time rather than routing hops, allowing it to efficiently avoid network congestion.

In order to achieve fast perception of congestion recovery, Ref. [17] modeled the relationship between the congestion recovery process and time in Q-routing and proposed to use the R function to estimate the rate of change of the Q-function with time and then estimate the rate of change of the Q-function over time. The R function is used to calculate the Q-value corresponding to each current neighbor node when making routing decisions. The experiments show that the Q-routing scheme based on the change of Q-value prediction is used in the situation of frequent network congestion. Compared with the original Q-routing scheme, it has better convergence speed and stability. In addition, Ref. [18] used dual reinforcement learning to improve the Q-routing and obtained better performance.

Reference [19] applied the Q-learning method to WSN and proposed the QELAR scheme. Due to the complex working environment of WSN and the frequent changes of network topology, traditional routing methods often fail to achieve good results in the WSN environment. The QELAR mainly solves the lifetime problem of WSN. Similar to Q-routing, the QELAR also uses the Markov process to model the process of data packet transmission in the network. Combining numbers as reinforcement learning, the feedback makes the routing algorithm able to make intelligent routing decisions according to the current state of the remaining energy of the system, so as to ensure the normal working time of the WSN network as long as possible.

After QELAR, Ref. [20, 21] proposed the MARLIN and MARLIN-Q models, and used MDP to model the packet sending and retransmission process of the WSN network. Figure 7 shows a schematic diagram of the state transition model of each routing node controlling the forwarding of data packets in the MARLIN-Q scheme. In the work of MARLIN and MARLIN-Q, the data packet p is defined in the state space S of each routing node according to the current data packet retransmission times as follows:

$$S = \{0, 1, \ldots, K - 1\} \cup \{\text{rcv}, \text{drop}\}. \tag{11}$$

The action space that each routing ith node can perform in the sth state includes the selected modem type and the next hop routing node that the corresponding modem can reach

$$A_i^M(s) = \{a = \, < j, m > | m \in M, j \in \text{Neighbor}_i^m\}, \tag{12}$$

where M is the set of modem types that the node has and Neighbor_i^m represents the set of neighbor nodes that the node can reach by using the modem type m. The MARLIN series algorithm cleverly designs the feedback function so that the feedback value obtained by the reinforcement learning model of each node is positively correlated with the data packet transmission delay and at the same time imposes a large penalty on a packet loss (*drop*) behavior, which can be used for reliable and low-latency data transmission of underwater sensor networks. In the real network scenario, through continuous trial and learning, the MARLIN series models can adaptively calculate the state transition probability $P_{i,s \longrightarrow rcv}^{(j,m)}$ through historical data and then ensure the route quality-of-service (QoS) of WSN network. In addition, by changing the maximum number of retransmissions K in the MDP process, The MARLIN-Q can support different types of QoS requirements, such as accelerated forwarding services requiring low latency and reliable transmission services requiring guaranteed reliability. The MARLIN-Q has tested the algorithm performance under different network parameters and loads in the simulation environment, and the results show that compared with the existing state-of-the-art underwater sensor network routing and transmission algorithm CARP [49] and QELAR optimized for

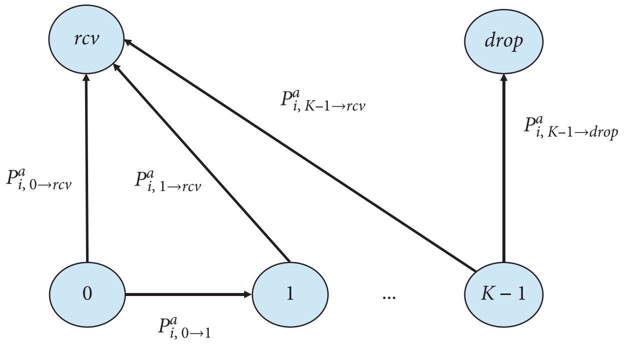

Figure 7: Illustration of MARLIN-Q transitions states.

network lifetime, the MARLIN-Q algorithm can effectively avoid the failure retransmission in the process of data packet transmission and have better performance in terms of effective throughput, delay, and energy consumption.

After investigation, most of the existing intelligent routing algorithms based on the Q-learning model the forwarding process of data packets in the network with MDP and then convert the routing optimization problem into a model-based Q-learning problem, and build on this basis. Due to the characteristics of MDP modeling and model-based Q-learning, its optimization objectives are mainly performance evaluation indicators that can be accumulated hop by hop, such as delay, throughput, and energy consumption. The intelligent routing algorithm designed using the model-based Q-Learning method can adapt itself to the dynamically changing network environment, and because its MDP model is known, its decision-making process has better interpretability than other deep learning-based methods. So it has a wide range of applications in the scenarios where the network state fluctuates greatly, such as the WSN network. However, for routing optimization problems with higher input and output dimensions and more complex optimization objectives, it is very difficult to explicitly establish an MDP model. In addition, the packet-level routing control methods commonly used in the existing Q-learning-based routing optimization methods are difficult to meet the requirements of the backbone network. Therefore, the application scenarios of existing intelligent routing algorithms based on Q-Learning still have great limitations.

4.3. Intelligent Routing Algorithm Based on Deep Reinforcement Learning. With the development of deep learning technology in recent years, researchers have begun to try to apply deep reinforcement learning (DRL) technology to intelligent routing and traffic engineering scheme design. Compared with Q-learning, the DRL methods can learn more complex strategies to solve routing optimization problems with larger states, larger decision spaces, and more complex optimization objectives.

Reference [14] applied deep reinforcement learning to intradomain traffic engineering (DRL-TE) scheme. Similar to the classic semi-state-independent traffic engineering (SMORE) scheme proposed by [7] in 2018, the DRL-TE divides the traffic engineering problem into two parts: static multipath solution and online dynamic adjustment of path split ratio. The DRL-TE uses traditional methods to generate paths and utilizes a deep reinforcement learning unit to complete the process of dynamically adjusting the path split ratio online. In this scheme, the deep reinforcement learning model takes the current delay and throughput corresponding to each session as the state of reinforcement learning, the path split ratio as the action of reinforcement learning, and the performance evaluation function of each session as the feedback of reinforcement learning. In this way, the network status information is dynamically sensed, the distribution ratio of each path is controlled, and the optimal distribution is learned adaptively according to the feedback results of each session. In order to deal with the continuous action space problem caused by the split ratio, the DRL-TE adopts the deep deterministic policy gradient algorithm (DDPG) as the reinforcement learning model and adopts the experience playback method specially designed for traffic engineering to ensure the convergence of the reinforcement learning model. Compared with SMORE, which needs to accurately predict the traffic matrix at the next moment in order to use the linear programming model to solve the optimal split ratio and can only optimize a limited target (such as maximum link utilization), DRL-TE only needs traffic characteristic information can automatically predict future traffic changes and make decisions that maximize the value of the total benefit function of each session. Therefore, it has better generality and robustness than SMORE method, which requires less assumptions about application scenarios. It is simulated in the NS-3 environment, and the experimental results show that compared with traditional routing and traffic engineering algorithms, the DRL-TE has obvious advantages in terms of delay, throughput and the utility function index. In addition, the comparative experiment directly using the original DDPG algorithm shows that the machine learning model is used to solve the problem. It is necessary to improve the original machine learning algorithms in traffic engineering problems, and it may be difficult to achieve ideal results by directly applying the existing machine learning models to routing optimization and traffic engineering problems.

In addition to the field of traffic engineering, DRL has also been applied to the optimization task of intelligent routing configuration. Reference [31] tried to use a DRL unit to predict the future network traffic based on historical traffic data and calculated the appropriate routing configuration based on the traffic prediction ability of the reinforcement learning model. It takes the historical traffic matrix as the input of the reinforcement learning model, the weight of each link is used as the output of the reinforcement learning model, and the reinforcement learning model (TRPO) passes the historical traffic according to the learned experience and knowledge. The matrix predicts the future traffic and performs routing configuration by adjusting the link weights, so as to achieve the goal of optimizing the maximum link utilization of the entire network and completing the load balancing. It is also pointed out that the representation of routing rules has a strong correlation with the convergence of reinforcement learning models. For a network topology $G(V, E)$, a destination node-based routing rule form with an output dimension of $|V|.|E|$ is directly used as the output action of the above reinforcement learning model. That is, for each node v for each destination node d, set a split ratio to all its neighbor nodes, and then, the above reinforcement learning model will be difficult to converge due to the high output dimension. Therefore, the action of the reinforcement learning model in this work sets a real weight for each link, and the link weight is mapped into a routing rule through a traditional rule-based approach. This reduces the output dimension of the reinforcement learning model to $|E|$, so as to reduce the size of the action space of the reinforcement learning model, reduce the difficulty of exploration and learning, and achieve the effect of accelerating the convergence. In this work, sparse and nonsparse gravity/bimodal models are used to generate different types of flow matrix sequences to test the performance of the algorithm. The simulation results show that for the traffic matrix with obvious regular characteristics, the reinforcement learning model can achieve good routing configuration through traffic prediction, which is better than the traffic-independent optimal routing [50] and close to the optimal routing configuration effect. However, when the traffic matrix no longer has obvious regular characteristics, the performance of this method will drop significantly. In fact, traffic changes in real scenarios may be irregular, including many burst traffic, so the traffic prediction and routing configuration capabilities of the above models under real traffic data are still a problem worth exploring.

Although the DRL model can theoretically predict the future traffic and make optimal routing decisions based on the network state data or historical information, the results of the DRL model in the current experiments are far from optimal. Ref. [38] compared the effects of several reinforcement learning models on routing tasks and put forward some guiding suggestions for using reinforcement learning models to solve routing problems. First of all, the author through a simple scenario deployment experiment of a Q-routing model [16] shows that the reinforcement learning intelligent routing model of packet-level routing control is difficult to apply to application scenarios with high throughput, and the time-segment-level routing control model will be a more recommended way. Secondly, the intelligent routing scheme that uses explicit path selection as the action of reinforcement learning unit is difficult to converge to the ideal result. As mentioned in Section 3.2, the number of paths increases exponentially with the growth of the network size, and the path selection-based scheme will undoubtedly greatly increase the learning and exploration capabilities of the reinforcement learning model. Based on the above two points, this article also chooses the scheme of controlling the link weight through the reinforcement learning model and then indirectly realizing the routing control. Compared with the direct generation of real link weights by Ref. [37], Ref. [38] scheme discretizes the link weights, further reduces the size of the action space from infinite to finite, and selects the corresponding weights for each link. The process is processed by a single reinforcement learning model, which further reduces the decision difficulty and exploration space of each reinforcement learning model. The generated link weight is used as the edge weight of the shortest path algorithm for routing calculation. In order to ensure the policy consistency of this multiagent cooperative routing model, Ref. [50] used the latest multiagent deep deterministic policy gradient (MADDPG) algorithm [51] to train the model. The final experimental results show that the reinforcement learning intelligent routing algorithm based on the offline link weight has better load balancing characteristics than the shortest path routing, that is, the shorter router average waiting time.

Existing intelligent routing schemes based on deep reinforcement learning have achieved certain results in intradomain traffic engineering and intelligent routing optimization tasks. The deep reinforcement learning model has good versatility and generalization. It can not only optimize the global performance evaluation indicators of the network, such as the maximum link utilization rate of the entire network and the average waiting length of routers, but also optimize the private benefit value corresponding to each session function. In addition, compared with traditional routing optimization algorithms based on rules or mathematical models, intelligent routing algorithms based on deep reinforcement learning do not need to make assumptions about the environment and can adapt to dynamically changing network environments. However, it is not difficult to find that there is a strong correlation between the convergence of a deep reinforcement learning model and the form of routing rules generated, and an excessively high output dimension often makes the deep reinforcement learning model unable to converge. Therefore, in the existing research work, the deep reinforcement learning model generally completes the flow control indirectly by controlling the path split ratio or link weight, rather than directly generating the routing path by path selection or path generation. In fact, even though the existing work has tried to reduce the routing decision difficulty of deep reinforcement learning units as much as possible, and has made significant progress, there is still a lot of room for improvement in the performance of existing solutions in complex application scenarios. In addition, limited by the model performance of deep reinforcement learning, most of the existing schemes adopt time-segment-level routing control methods, while packet-level routing control methods are not suitable for such intelligent routing schemes. Robustness and reliability are very important properties for routing algorithms, but the existing research on intelligent routing algorithms based on deep reinforcement learning is far from enough.

5. Training and Deployment of Intelligent Routing Algorithms

Although there have been many related works on intelligent routing algorithms based on machine learning in recent years, these works mainly focus on the principle design of intelligent routing algorithms, algorithm accuracy, convergence, and other issues. There is not yet a mature and complete framework for training and deployment. This article discusses the advantages and disadvantages of different training methods and deployment methods of intelligent routing algorithms and proposes two types of reasonable intelligent routing training and deployment frameworks, so that intelligent routing algorithms can be used in real scenarios with low cost and high reliability.

5.1. Training Method: Online and Offline. The training methods of the intelligent routing algorithm model are mainly divided into two types: online and offline. Figure 7 shows the training method of the existing intelligent routing scheme. The intelligent routing models based on supervised learning are all trained offline, while the models based on reinforcement learning can be trained both online in the real environment and offline in the simulation environment.

Generally speaking, the offline training process of the model first needs to collect data from the real environment, which may be the traffic matrix, the status information of each node in the network, and the corresponding routing decision labels. After the data are processed, it is used in the offline training process of the machine learning model on the server. After the training is completed, the model is deployed to the real environment to make online routing decisions. Offline training and online testing, deployment is a common training deployment method in the field of deep learning. However, for intelligent routing algorithms, offline training often faces three challenges: (1) the collection of training data may require relatively high costs; (2) the network state in the real scene may be different from the training data set, causing the routing algorithm to fail to achieve the expected effect or even to make errors; and (3) for reinforcement learning, it may be difficult to build a simulated training environment similar to the real environment.

For reinforcement learning methods, online training can ensure that the model adapts to changes in the network environment and avoids the difficulties and extra costs brought by the offline simulation environment construction. However, the routing security and reliability problems brought by online training make it difficult to deploy intelligent routing methods that require online training in actual deployment. In fact, in online reinforcement learning, security is an issue that has been widely studied [52, 53]. The reinforcement learning models may produce unpredictable behaviors in the initial stage of training and the exploratory stage in the training process. When reinforcement learning methods are applied to routing tasks, these unpredictable behaviors may cause serious consequences including routing loops and link congestion. Therefore, ensuring the security and reliability of the online reinforcement learning routing algorithm training process will be an important prerequisite for its deployment in real scenarios.

5.2. Deployment Method: Centralized and Distributed. As there are many intelligent routing algorithms proposed, how to deploy these algorithms in the existing computer network architecture is receiving more attention. The deployment methods of intelligent routing algorithms are mainly divided into two types: distributed and centralized.

Figure 8 shows the schematic diagrams of the framework structures of the two deployment schemes. The intelligent routing algorithm is deployed in the centralized controller, and the routing decision is made dynamically according to the network state information collected by the controller, and the routing decision is sent to each routing node through the centralized controller. The proposal of the SDN network structure provides the theoretical possibility for the centralized deployment of intelligent routing algorithms, and the above centralized control process can be completed by using the intelligent routing control unit as an application on the SDN controller. In a relatively independent application scenario such as data center network traffic engineering, it is a feasible solution to deploy the intelligent routing scheduling scheme using a centralized method.

The deployment of a centralized solution requires deploying a centralized routing controller in the network and designing a centralized routing control protocol. However, the routing protocols in the current computer network architecture are still dominated by distributed routing protocols.

Compared with centralized routing protocols, distributed routing protocols have better scalability. As can be seen from Figure 7, there are many existing intelligent routing algorithms that can support distributed routing decisions. These distributed intelligent routing algorithms have made progress in terms of convergence and robustness. The corresponding router hardware needs to be further developed and improved [15]. With the development of programmable routing devices, it will be possible to deploy distributed intelligent routing algorithms in real networks in the future. However, the existing distributed intelligent routing algorithms mainly focus on the accuracy and convergence of routing methods and do not consider the compatibility of existing network layer structures and protocols. For the distributed intelligent routing algorithm, how to carry out incremental deployment on the basis of compatibility with the existing network layer structure will be a problem worth thinking about in the future.

5.3. Intelligent Routing Training and Deployment Model Design. Based on the above discussion, this section summarizes and proposes two types of future feasible intelligent routing training and deployment frameworks: (1) an intelligent routing framework combining centralized offline training and online decision-making; and (2) a secure online reinforcement learning routing framework.

Figure 9 shows the workflow of the intelligent routing deployment framework combining centralized offline training and online routing decision-making. In this intelligent routing deployment scheme, the router data plane needs to collect the network traffic characteristic information and pass it up to the control layer to complete the intelligent routing model training and online routing decision-making process. The intelligent routing decision-making model uses historical network state information and network simulation environment to complete offline training in a single node with sufficient computing power

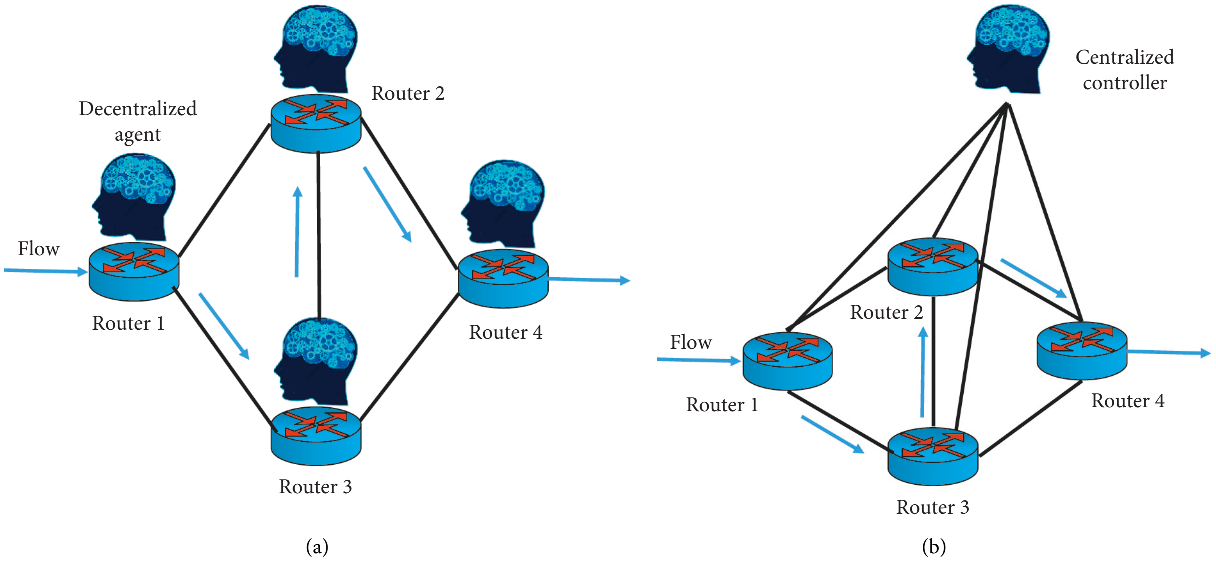

Figure 8: Illustrations of ML routing structures.

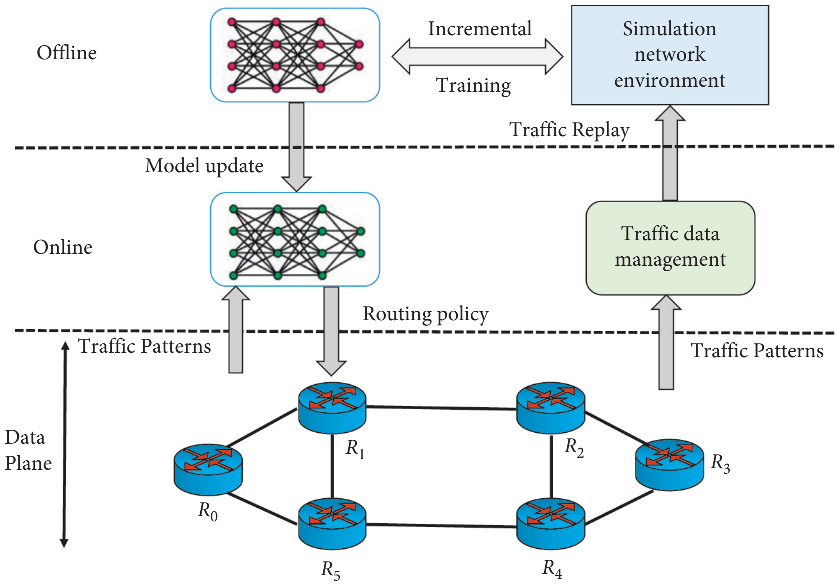

Figure 9: ML-based routing architecture for training and deployment.

and publishes the trained model parameters to the online routing decision-making unit. The corresponding routing decision unit can either deploy the online intelligent routing unit to the control plane of each router by means of distributed deployment or place the intelligent routing unit in a centralized routing controller by means of centralized deployment. For example, in order to adapt to the network topology and traffic characteristics that change dynamically with time, the above model adopts the closed-loop learning method to periodically train the intelligent routing model incrementally according to the latest network traffic characteristics. The training process of the intelligent routing model based on machine learning needs to consume a lot of computing and storage resources, and the centralized offline training makes each routing node in the network do not need to deploy these resources, which can effectively reduce the deployment cost of the intelligent routing algorithm.

The intelligent routing deployment scheme of centralized offline training and online routing decision is suitable for most existing intelligent routing algorithms, and is consistent with the idea of offline training and online decision-making in machine learning. However, for reinforcement learning models, whether it is an on-policy model or an off-policy model, the interaction with the environment is an essential part of the learning process. Different from game tasks, it is very difficult to build a simulation environment consistent with the real network environment in routing optimization problems [30]. Correspondingly, the poor strategy at the beginning of the deep reinforcement learning model and its exploratory behavior in the learning process make it possible to directly train the intelligent routing model based on DRL in a real network environment, in order to solve the challenges faced by the intelligent routing strategy based on DRL in the training process.

In this article, referring to the idea of secure online reinforcement learning [53], an online training scheme of DRL intelligent routing model with reliability guarantee is proposed. Figure 10 shows the working flow of the scheme. Compared with the traditional reinforcement learning method, this scheme introduces a security monitoring module to judge whether the routing decision made by the reinforcement learning unit is safe or not based on rules. When the routing decision made by the reinforcement learning unit may have security risks, for example, including routing loops and triggering network congestion, the reinforcement learning unit uses a simple and reliable routing decision (such as shortest path routing) to replace the original routing decision, and at the same time imposes a penalty factor p on the reinforcement learning unit to avoid the reinforcement learning unit. The related work of online security learning in other network application scenarios shows that the DRL intelligent routing scheme based on online security learning has the ability to ensure the reliability of the routing learning process without affecting the original routing optimization goal [53]. It can not only solve the security problems that have not yet converged, but also ensure the reliability of the model without guaranteeing the interpretability of the model. It concerns about the unpredictability of routing behavior in network emergencies.

For the training and deployment framework of intelligent routing, the existing research work is still relatively small, but this article believes that the uninterpretability of the model and the unpredictability of routing behavior brought by the intelligent routing scheme will be an important challenge in the design of its training and deployment framework. Using the rule-based scheme to constrain the intelligent routing control unit may be an effective means to ensure the reliability of intelligent routing.

6. Opportunities and Challenges Faced by Intelligent Routing Algorithms

In recent years, intelligent routing algorithms have received considerable attention. In this section, the advantages of intelligent routing algorithms in solving the routing optimization problems and the challenges they face in the future development process are discussed.

6.1. Advantages of Intelligent Routing Algorithms. The data-driven intelligent routing algorithms are usually based on deep learning or reinforcement learning, which have five main advantages:

(1) *The Network State is Sensitive*. Compared with the traditional model-based routing algorithm, the intelligent routing algorithm can process higher-dimensional network state feature information, which makes the intelligent routing algorithm more sensitive to changes in the network state, and can quickly converge when the network state changes.

(2) *Data-driven*. Unlike traditional routing algorithms that use a fixed model to solve the routing strategy, the intelligent routing algorithm is data-driven, relies on fewer environmental assumptions, and uses historical data and spontaneous exploration of the environment to automatically model application scenarios and complete routing optimization, allowing it to adapt to different application scenarios and network environment changes.

(3) *Oriented to Service Quality*. Intelligent routing can help facilitate routing requests with varying levels of service quality. The data-driven intelligent routing algorithm can automatically learn the appropriate routing decisions according to the Quality-of-Service (QoS) requirements, unlike the traditional QoS routing optimization scheme, which creates a complex optimization model for each QoS requirement based on a large number of assumptions about the application scenarios.

(4) *Experience-Driven and Memory Characteristics*. Unlike standard routing algorithms based on models and rules, intelligent routing algorithms based on machine learning may remember prior experience by studying historical data, allowing the model to “eat a little and gain a wisdom” similar to a human being. The effect of route optimization improves as the company grows.

(5) *Routing Decisions Consider the Past, Present, and Future*. The recurrent neural network structure (RNN) and its corresponding extensions (GRU, LSTM) can model the past historical information well, and the reinforcement learning model endows the intelligent routing algorithm not only with the current routing effect, but also in predicting the future network state changes, the ability to avoid possible future network congestion in advance.

6.2. Challenges to Intelligent Routing Algorithms. Corresponding to the advantages of intelligent routing algorithms, the future development process of intelligent routing methods also faces many challenges:

(1) *Network Feature Information Extraction*. In the intelligent routing method, the network state information may be organized in the form of topology structure, and due to the dynamic change of the network scene, the dimension of the network state information may change. Traditional machine learning methods have difficulties in processing this type of network state information. Existing intelligent routing algorithms try to use graph neural network model (GNN) to model and extract network state information [26, 30]. The GNN method has good generalization for different topological structures, but whether the existing GNN methods can complete the modeling of dynamic large-scale

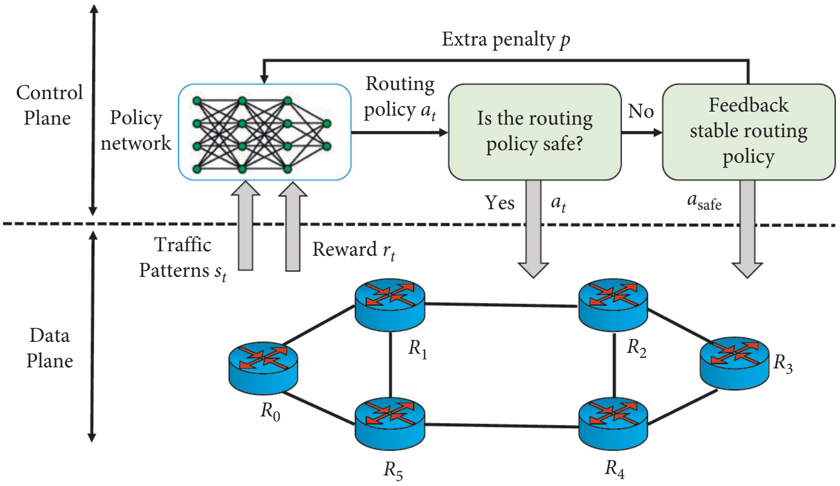

Figure 10: DRL-based framework for secure online learning in routing.

topological structures in real scenarios of routing optimization problems still lacks sufficient experimental support.

(2) *Algorithm Convergence*. Compared with games, image recognition, natural language processing, and other scenarios where machine learning has been widely used, the input and output dimensions of routing optimization problems are higher, and the target strategy is more complex. Existing research shows that for complex routing optimization with high input and output dimensions. However, the existing machine learning schemes are often difficult to converge to the optimal solution. In order to solve the problem that the model is difficult to converge, it is often necessary to reduce the input and output dimensions, discretize the decision space, or use indirect control of routing decisions to simplify the policy complexity to reduce the convergence difficulty of the model. However, even with these schemes, many convergence results are still far from the theoretical optimal value.

(3) *Algorithm Scalability*. Routing algorithms must meet a number of requirements, one of which is scalability. Existing machine learning-based intelligent routing methods are mostly created and tested on small topologies with little more than 20 nodes. A bigger topology results in an exponential growth in the number of network states and a greater difficulty in making routing decisions. The design of intelligent routing algorithms in the future will have a problem in ensuring that the algorithm can still get good results in a big topology. Furthermore, when the topology is complex, the centralized routing control method might result in high data exchange costs and network state transfer delays, reducing scalability. The future difficulty of ensuring the consistency of each node's routing strategy under the huge topology of a distributed intelligent routing algorithm will be solved.

(4) *Algorithm Interpretability*. Another problem faced by intelligent routing methods is the unpredictability and uninterpretability of routing strategies. Compared with traditional routing algorithms based on mathematical models, deep learning-based methods often have unpredictable behaviors. When poor routing decisions are made, it is difficult for the operator to locate the cause of the error, and it is almost impossible to correct the model for the error. Therefore, how to improve the interpretability of intelligent routing algorithms will be a challenge in the future development of intelligent routing methods.

(5) *Model Training Cost*. For intelligent routing algorithms based on supervised learning, collecting enough and accurate enough labeled data is sometimes a very expensive thing. Different from face recognition and other application scenarios where training is done once and for all, as the network environment changes, existing intelligent routing may need to repeatedly collect training data and retrain. Therefore, how to improve the data efficiency of the intelligent routing training process is an important challenge in the deployment of intelligent routing solutions. When faced with similar problems, reducing the training cost through meta-learning is a feasible solution [54]; however, there is no perfect research in the field of routing. In addition, for the intelligent routing method based on deep reinforcement learning, whether it is online training or offline training, the high training cost and the hidden reliability risks brought to the system during the training process are challenges that need to be solved urgently.

(6) *Handling of Network Emergencies*. Another issue that intelligent routing algorithms will encounter in the future development phase is figuring out how to cope with network crises. In practice, traffic surges and network state changes induced by network equipment failures are all too prevalent. However, these

crises come in a variety of forms, and many of them have never been seen in training data. It is challenging to verify that these situations are handled effectively with the present data-driven intelligent routing algorithms. Even approaches that can dynamically adjust to environmental changes, such as Q-Learning, cannot cope with unexpected and significant network shifts. To deal with abrupt changes in network circumstances, the concept of "secure online reinforcement learning" [53] is applied. It might be a future solution, but determining how to effectively recognize network crises is an issue.

(7) *Real Scenario Deployment*. For intelligent routing methods, how to deploy them in real scenarios is a huge challenge. Intelligent routing, as compared to standard routing methods, necessitates greater computational resources and higher routing performance. Simultaneously, the training data collecting and routing perception processes for the original routing protocol must be changed so that the intelligent routing algorithm may get data from the intelligent unit. Although the emergence of SDN networks and programmable routing equipment increases the processing capacity of the router control layer, even intelligent routing algorithms remain challenging to deploy on a broad scale under the current network design. It may be the future trend of intelligent routing algorithms to develop routing equipment that matches the intelligent routing scheme while maximizing the performance of intelligent routing algorithms and boosting their compatibility and scalability with traditional routing algorithms.

7. Conclusion

The present intelligent routing algorithms are largely split into two types, based on supervised learning and based on reinforcement learning, according to this article's findings. (1) The supervised learning-based intelligent routing technique primarily completes the routing solution by either replacing the existing routing algorithm with the deep learning model or supporting the traditional routing algorithm. The deep learning method makes the intelligent routing algorithm more sensitive to the environment and has a faster convergence speed. The data-driven auxiliary module can also make the routing decision made by the traditional routing algorithm more accurate and avoid congestion in advance. (2) Reinforcement learning-based routing algorithms can adapt to diverse routing application settings and maximize various network performance metrics. The model-based Q-Learning method is widely used in the routing optimization process of wireless sensor networks, whereas the deep reinforcement learning method is used to solve various complex routing optimization problems like intradomain traffic engineering and intelligent routing algorithms based on traffic prediction.

This article analyzes the advantages and disadvantages of online and offline intelligent routing training schemes, centralized and distributed intelligent routing deployment schemes, and further proposes a closed-loop learning framework of offline centralized training plus online deployment, as well as adaptive online training and security learning. The combined intelligent routing deployment framework has reliable performance. These two frameworks provide the possibility for low-cost and high-reliability deployment of intelligent routing algorithms based on machine learning in real scenarios.

This article discusses the opportunities and challenges in the future development of intelligent routing algorithms and proposes possible future research directions for intelligent routing algorithms based on machine learning in response to these challenges.

Data Availability

The data used for the findings of this study are available from the authors upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the National Research Foundation (NRF), Korea (2022R1A2C4001270).

References

[1] H. Baars, A. Tank, P. Weber, H. G. Kemper, H. Lasi, and B. Pedell, "Cooperative approaches to data sharing and analysis for industrial internet of things ecosystems," *Applied Sciences*, vol. 11, no. 16, pp. 7547–7618, 2021.

[2] C. Lim, "Enhancing robustness of per-packet load-balancing for fat-tree," *Applied Sciences*, vol. 11, no. 6, pp. 2664–2718, 2021.

[3] E. Akin and T. Korkmaz, "Comparison of routing algorithms with static and dynamic link cost in software defined networking (SDN)," *IEEE Access*, vol. 7, pp. 148629–148644, 2019.

[4] P. Tsai, J. Zhang, and M. Tsai, "An efficient probe-based routing for content-centric networking," *Sensors*, vol. 22, no. 1, pp. 1–18, 2022.

[5] B. Isyaku and K. A. Bakar, E. H. Alkhammash, F. Saeed, and F. A. Ghaleb, Route path selection optimization scheme based link quality estimation and critical switch awareness for software defined networks," *Applied Sciences*, vol. 11, no. 19, pp. 9100–9115, 2021.

[6] Y. Wu, G. Hu, F. Jin, and S. Tang, "Multi-objective optimisation in multi-QoS routing strategy for software-defined satellite network," *Sensors*, vol. 21, no. 19, p. 6356, 2021.

[7] G. Németh, "On the competitiveness of oblivious routing: a statistical view," *Applied Sciences*, vol. 11, no. 20, pp. 9408–9419, 2021.

[8] Q. Lai, Z. Zhou, and S. Liu, "Joint entity-relation extraction via improved graph attention networks," *Symmetry*, vol. 12, no. 10, pp. 1–17, 2020.

[9] S. Wu, S. Zhong, and Y. Liu, “Deep residual learning for image steganalysis,” *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 10437–10453, 2018.

[10] D. Silver, J. Schrittwieser, K. Simonyan et al., “Mastering the game of go without human knowledge,” *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[11] B. P. R. Killi and S. V. Rao, “Controller placement in software defined networks: a comprehensive survey,” *Computer Networks*, vol. 163, pp. 106883–106954, 2019.

[12] P. Bosshart and G. Gibb, H. S. Kim, G. Varghese, and McKeown, Forwarding metamorphosis,” *ACM SIGCOMM - Computer Communication Review*, vol. 43, no. 4, pp. 99–110, 2013.

[13] P. Bosshart, D. Daly, G. Gibb, M. Izzard, and McKeown, “P4 programming protocol-independent packet processors,” *ACM SIGCOMM - Computer Communication Review*, vol. 44, no. 3, pp. 87–95, 2014.

[14] X. Zhiyuna, T. Jian, and M. Jingson, “Experience-driven networking: a deep reinforcement learning based approach,” in *Proceedings of the IEEE International Conference on Computer Communications*, pp. 1871–1879, Los Angeles, LA, U.S.A, January 2018.

[15] B. Mao, Z. M. Fadlullah, F. Tang et al., “Routing or computing? the paradigm shift towards intelligent computer network packet transmission based on deep learning,” *IEEE Transactions on Computers*, vol. 66, no. 11, pp. 1946–1960, 2017.

[16] D. R. Militani, H. P. Moraes, R. Rosa, L. Wuttisittkulkij, M. A. Ramirez, and Z. R. Demóstenes, “Enhanced routing based on reinforcement machine learning –a case of VoIP service,” *Sensors Journal*, vol. 21, no. 2, pp. 1–23, 2021.

[17] M. Gregurić, M. Vujić, C. Alexopoulos, and M. Miletić, “Application of deep reinforcement learning in traffic signal control: an overview and impact of open traffic data,” *Applied Sciences*, vol. 10, no. 11, pp. 4011–4017, 2020.

[18] Q. Ding, R. Zhu, H. Liu, and M. Ma, “An overview of machine learning-based energy-efficient routing algorithms in wireless sensor networks,” *Electronics*, vol. 10, no. 13, pp. 1539–1615, 2021.

[19] T. Yunsi Fei and Y. Fei, “QELAR: a machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 796–809, 2010.

[20] S. Basagni, V. D. Valerio, P. Gjanci, and C. Petrioli, “Finding MARLIN: exploiting multi-modal communications for reliable and low-latency underwater networking,” in *Proceedings of the IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, no. 1–9, Atlanta, GA, U.S.A, May 2017.

[21] S. Basagni, V. D. Valerio, P. Gjanci, and C. Petrioli, “MARLIN-Q: multi-modal communications for reliable and low-latency underwater data delivery,” *Ad Hoc Networks*, vol. 82, no. 5, pp. 134–145, 2019.

[22] N. Jay, N. Rotman, and B. Godfrey, “A deep reinforcement learning perspective on internet congestion control,” in *Proceedings of the ACM International Conference on Machine Learning*, pp. 3050–3059, Washington DC, U.S.A, May 2019.

[23] P. Sirinam, M. Imani, M. Juarez, and M. Wright, “Deep f,” in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1928–1943, New York, NY, U.S.A, October 2018.

[24] H. Mao, R. Netravali, and M. Alizadeh, “Neural adaptive video streaming with pensieve,” in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pp. 197–210, Kentucky, KY, U.S.A, August 2017.

[25] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, and Osamu, “State-of-the-Art deep learning: evolving machine intelligence toward tomorrow’s intelligent network traffic control systems,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.

[26] D. J. Shin and J. J. Kim, “Deep reinforcement learning-based network routing technology for data recovery in exa-scale cloud distributed clustering systems,” *Applied Sciences*, vol. 11, no. 18, pp. 8727–8819, 2021.

[27] V. P. Rekkas, S. Sotiroudis, P. Sarigiannidis, S. Wan, G. K. Karagiannidis, and S. K. Goudos, “Machine learning in beyond 5G/6G networks-state-of-the-art and future trends,” *Electronics*, vol. 10, no. 22, p. 2786, 2021.

[28] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[29] G. E. Hinton, S. Osindero, and Y. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[30] K. Rusek, J. S. Varela, and A. Mestres, “Unveiling the potential of graph neural networks for network modeling and optimization in SDN,” in *Proceedings of the 2019 ACM Symposium on SDN Research*, pp. 140–151, New York, NY, U.S.A, April 2019.

[31] K. B. Lee, D. K. Kang and Y. C. Kim, Deep reinforcement learning based optimal route and charging station selection,” *Energies*, vol. 13, no. 23, pp. 6255–6324, 2020.

[32] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[33] B. Yue, J. Fu, and J. Liang, “Residual recurrent neural networks for learning sequential representations,” *Information*, vol. 9, no. 3, pp. 56–14, 2018.

[34] H. Li, D. Xu, T. Zhu, F. Shang, Y. Liu, and Lu, “Graph convolutional networks by architecture search for PoISAR image classification,” *Remote Sensing Journal*, vol. 13, no. 7, pp. 1–17, 2021.

[35] Q. Wang, H. Zhang, C. Qu, Y. Shen, X. Liu, and J. Li, “RLSchert: an hpc job scheduler using deep reinforcement learning and remaining time prediction,” *Applied Sciences*, vol. 11, no. 20, pp. 9448–9516, 2021.

[36] Z. Zhuang, J. Wang, Q. Qi, and H. J. Sun, “Graph-aware deep learning based intelligent routing strategy,” in *Proceedings of the 2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, pp. 441–444, Chicago, IL, U.S.A, October 2018.

[37] D. Nace and M. Pioro, “Max-min fairness and its applications to routing and load-balancing in communication networks: a tutorial,” *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 5–17, 2008.

[38] Q. Xu, Y. Zhang, and K. Wu, “Evaluating and boosting reinforcement learning for intra-domain routing,” in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensors Systems*, pp. 955–961, Monterey, CA, U.S.A, January 2019.

[39] N. Kato, Z. M. Fadlullah, B. Mao, F. Tang, and Akashi, “The deep learning vision for heterogeneous network traffic control: proposal, challenges, and future perspective,” *IEEE Wireless Communications*, vol. 24, no. 3, pp. 146–153, 2017.

[40] B. Chen, D. Zhu, Y. Wang, and P. Zhang, “An approach to combine the power of deep reinforcement learning with a graph neural network for routing optimization,” *Electronics*, vol. 11, no. 3, pp. 368–416, 2022.

[41] Y. Hua, Z. Zhao, Z. Liu, X. R. Chen, and H. Zhang, “Traffic prediction based on random connectivity in deep learning with long short-term memory,” in *Proceedings of the 2018*

IEEE 88th Vehicular Technology Conference (VTC-Fall), pp. 1–6, Chicago, IL, U.S.A, August 2018.

[42] L. Nie, D. Jiang, L. Guo, and S. Yu, "Traffic matrix prediction and estimation based on deep learning in large-scale IP backbone networks," *Journal of Network and Computer Applications*, vol. 76, no. 3, pp. 16–22, 2016.

[43] X. Xiang and S. Foo, "Recent advances in deep reinforcement learning applications for solving partially observable Markov decision processes (POMDP) problems: Part 1-fundamentals and applications in games, robotics and natural language processing," *Machine Learning and Knowledge Extraction*, vol. 3, no. 3, pp. 554–581, 2021.

[44] B. Gašperov, S. Begušić, P. Šimović, and Z. Kostanjčar, "Reinforcement learning approaches to optimal market making," *Mathematics*, vol. 9, no. 21, pp. 2689–2719, 2021.

[45] D. Silver, G. Lever, N. Heess, D. Thomas, W. Daan, and R. Martin, "Deterministic policy gradient algorithms," in *Proceedings of the ACM 31st International Conference on Machine Learning*, pp. 387–395, New York, NY, U.S.A, June 2014.

[46] S. Backman, D. Lindmark, K. Bodin, M. Servin, H. J. Löfgren, and H. Lofgren, "Continuous control of an underground loader using deep reinforcement learning," *Machines*, vol. 9, no. 10, pp. 216–218, 2021.

[47] J. Schulman, S. Levine, P. Mortiz, M. Jordan, and P. Abbeel, "Trust region policy optimization," in *Proceedings of the ACM 32nd International Conference on Machine Learning*, pp. 1889–1897, Los Angeles, LA, U.S.A, May 2015.

[48] W. Zhao, H. Chu, X. Miao et al., "Research on the multiagent joint proximal policy optimization algorithm controlling cooperative fixed-wing UAV obstacle avoidance," *Sensors*, vol. 20, no. 16, pp. 1–16, 2020.

[49] S. Basagni, C. Petrioli, R. Petroccia, and D. Spaccini, "CARP: a channel-aware routing protocol for underwater acoustic wireless networks," *Ad Hoc Networks*, vol. 34, no. 5, pp. 92–104, 2015.

[50] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Räcke, "Optimal oblivious routing in polynomial time," *Journal of Computer and System Sciences*, vol. 69, no. 3, pp. 383–394, 2004.

[51] S. Gu, M. Geng, and L. Lan, "Attention-based fault-tolerant approach for multi-agent reinforcement learning systems," *Entropy*, vol. 23, no. 9, pp. 1–15, 2021.

[52] J. Garcia and F. Fernandez, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[53] S. Dass and A. S Namin, "Reinforcement learning for generating secure configurations," *Electronics*, vol. 10, no. 19, pp. 1–19, 2021.

[54] C. Kim, "Deep reinforcement learning by balancing offline Monte Carlo and online temporal difference use based on environment experiences," *Symmetry*, vol. 12, no. 10, pp. 1685–1714, 2020.