IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 June 2024, accepted 26 July 2024, date of publication 5 August 2024, date of current version 20 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3438938

RESEARCH ARTICLE

Software-Defined Networking-Based Resilient Proactive Routing in Smart Grids Using Graph Neural Networks and Deep Q-Networks

MD AMINUL ISLAM[1], RACHAD ATAT[2], (Senior Member, IEEE), AND MUHAMMAD ISMAIL[3], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, Jagannath University, Dhaka 1100, Bangladesh
[2]Department of Electrical and Computer Engineering, Texas A&M University at Qatar, Doha, Qatar
[3]Department of Computer Science, Tennessee Technological University, Cookeville, TN 38505, USA

Corresponding author: Md Aminul Islam (aminul@cse.jnu.ac.bd)

ABSTRACT The enhanced functionality of the smart grid depends on the robust interconnection between its physical and cyber-layer components. Two distinct categories of control data packets exist within smart grids: fixed-scheduling (FS) and event-driven (ED). An intelligent routing strategy is required to satisfy latency requirements for FS and ED packets across various quality-of-service (QoS) levels and must be resilient to failures. Our proposed software-defined routing strategy balances requirements by dynamically adjusting decisions based on packet types. It prioritizes paths with lower latency and higher throughput for ED packets while prioritizing paths with higher redundancy and lower congestion for FS packets. This strategy switches between proactive and resilient modes based on network conditions. First, the proactive routing module (PRM) utilizes a graph neural network (GNN) and a Q-learning (QL) algorithm to fix sub-optimal routes for efficient packet delivery under normal conditions. Second, the resilient routing module (RRM) combines a deep Q-network with GNN to select optimal routes that remain viable even during failures, ensuring continued operation and robustness. Both modules update the queue service rate (QSR) using QL-agent while avoiding congestion. The GNN ensures proactive module selection based on excessive congestion violations indicating failure conditions. Given the efficient performance of the PRM in normal conditions and the resilience of the RRM under failures, the proposed strategy presents a dual-mode routing that minimizes overhead with a high level of resilience. The proposed approach, evaluated using the IEEE 39-bus test system cyber-layer, effectively ensures desired QoS routing regardless of the conditions of the cyber-layer.

INDEX TERMS Smart grids, software-defined routing, data traffic prediction, graph neural network, deep Q-networks, resilient proactive routing.

I. INTRODUCTION

In light of advancements in traditional power grids, smart grids offer numerous consumer benefits, including increased reliability, economic viability, operational effectiveness, environmental friendliness, and security [1]. The cyber layer receives two distinct types of packets from field devices such as actuators and phasor measurement units (PMUs): event-driven (ED) and fixed scheduling (FS) packets [2]. Under normal conditions, FS traffic occurs at predefined intervals to ensure uninterrupted monitoring and control of the power system. However, ED packets are generated during emergencies like inclement weather, physical attacks, vandalism, etc. As a result, the ED traffic follows a pattern of sporadic arrivals related to unexpected emergencies. In addition, timely delivery of ED packets is vital to handle any emergency event effectively. Therefore, an efficient, resilient routing strategy is a proactive one capable of dynamically switching transmission between both types of packets depending on network conditions. Such a routing

The associate editor coordinating the review of this manuscript and approving it for publication was Alfeu J. Sguarezi Filho.

strategy must be capable of avoiding congested routes while being immune to the unexpected failure of communication nodes in the cyber layer.

The fundamental prerequisite for avoiding a congested or failed route is global observability and rapid network state updates, which can be accomplished using the software-defined networking (SDN) paradigm [2]. OpenFlow [3] is a communication protocol that grants network-based access to the forwarding plane of the router or forwarding switch. It is used by the SDN paradigm to facilitate communication between the SDN controller and switches in the cyber layer of the smart grid. Separating the control plane from the data plane in SDN unlocks the possibility of integrating advanced intelligence, enabling smart grids to adopt adaptive routing strategies. SDN generates a forwarding rule and installs it on OpenFlow switches along the path from source to destination. Changing routes is necessary to avoid congested switches as well as failed ones. To switch to a new route, we need to set up a new path from the central control of the network to the switches along the new direction. This route installation incurs delay, which causes additional packet delays and eventually raises the percentage of packets that do not meet the ED and FS latency constraints.

To avoid route installation delays, we proposed an adaptive routing strategy in SDN using RL, as described in [2]. In [2], we configured distinct queues for forwarding ED and FS packets. This separation ensures that ED packets are not impeded by FS packets when these packets are forwarded using a single queue. The RL-based routing approach satisfies various QoS (i.e., the percentage of packets that fail to meet the latency thresholds, packet loss, and average latency) achieved by selecting a sub-optimal route and adjusting the QSR (It measures how many packets can be served by the switch's queue in a given amount of time) based on the arrival rate. As we adapt the QSR to match the arrival rate for ED and FS packets, the already-received packets persist in congesting the forwarding switches. Hence, some packets fail to meet their required latency thresholds.

To address the issue of congestion in the forwarding switch, we proposed a GNN-based proactive routing strategy in [4]. This strategy involved selecting a sub-optimal fixed route, predicting congestion levels at the forwarding switches in real-time, and proactively adjusting queue service rates to accommodate the congestion and meet the desired QoS requirements. While our routing strategy in [4] satisfies all the intended QoS standards, it relies on a pre-fixed suboptimal route. Hence, the strategy is vulnerable to failure if a forwarding node malfunctions during real-time routing operations. To summarize, our previous work in [2] utilizes a data-driven approach based on QL, but it cannot predict future traffic congestion. In another work in [4], we employ a data-driven approach using GNN and a QL-based adaptive routing algorithm. However, this approach is not adaptive to switch failure conditions in the cyber layer of the smart grid. In this paper, we aim to propose a resilient, proactive routing strategy operating in dual mode (i.e., PRM, RRM) based on cyber layer conditions. The proposed strategy avoids congestion and is adaptive to node failure conditions to meet the desired QoS criteria under normal and failure conditions. The QoS requirements for network operations are crucial for the effective utilization of resources and immediate response during emergencies. In regular scenarios, the network resources must be used optimally to achieve the desired latency and packet loss for all types of traffic, particularly for ED packets. Hence, under failure conditions such as switch failure, timely and accurate communication is critical for the next course of action. Resilient network operations, such as RRM, help to maintain the stability of the network by constantly redirecting the traffic around the paths that are either unavailable or congested. This flexibility ensures that QoS is not compromised during disruptions and improves the grid's reliability. The main contributions of the dual-mode resilient, proactive routing strategy are as follows:

- First, we propose a dual-mode routing strategy based on normal and failure conditions of the cyber layer. Under normal conditions, the PRM utilizes GNN and RL techniques to proactively update the QSR on a fixed source-destination path [4]. Under failure conditions, the RRM combines deep Q-network (DQN) and GNN to select feasible routes.
- Second, we employ DQN agents trained to consider all outgoing ports. Their task is to find optimal and alternate routes in the event of failures, avoiding overloaded and failed nodes. DQN agents ensure timely transmission of ED and FS packets with low latency across the cyber layer. By predicting congestion and failures in the cyber layer, we formulate the ahead-of-time route that eliminates the route installation delay incurred by SDN.
- Third, we develop a GNN-based prediction module integrated into the intelligence plane to predict all outgoing port traffic conditions. The proposed GNN-based prediction model is then compared with multi-layer perceptron (MLP), convolutional neural network (CNN), and long-short-term memory (LSTM).
- Fourth, we test our proposed routing strategy using the cyber layer of the IEEE 39-bus test system [5] under both normal and failure conditions. Our investigations reveal that the proposed routing strategy, namely PRM and RRM, effectively guarantees the desired QoS for the smart grids, regardless of whether the cyber layer is functioning normally or experiencing failure.

The remainder of the paper is organized as follows. Related works are discussed in Section II. In Section III, the system model is presented. The problem statement and the proposed resilient proactive routing strategy are presented in Sections IV and V, respectively. Section VI presents the implementation details, hyperparameter optimization, prediction performance metrics, results, and discussions. Finally, conclusions and directions for future research are presented in Section VII.

II. RELATED WORKS

In this section, we summarize related works on data routing in the smart grid and prediction models in data and transportation networks that motivated us to build the prediction model of the cyber layer network conditions. In addition, we summarize works that used DQN as a solution to the routing problem.

A. TRADITIONAL ROUTING STRATEGIES

To mitigate the overloading of data networks due to emergency conditions, the works in [6] devised a load-balancing following position-based QoS-aware protocol. The work in [7] proposed a data collection strategy that considered FS traffic only. The work in [8] studied the relationship between throughput and transmission latency for smart meter data transfer while satisfying latency requirements. The objective of the multi-cast routing strategy proposed in [9] was to reduce the end-to-end delay while simultaneously satisfying bandwidth limits. The work in [10] introduced heuristic greedy algorithms designed for data routing where there was either no demand for latency or only a minimal latency requirement exists. An opportunistic routing approach for power line communications was proposed in [11] to reduce the delay associated with the delivery of packets.

B. SDN-BASED ROUTING STRATEGIES

By exploiting the flexibility of SDN, the work in [12] proposed an SDN data routing algorithm that guarantees QoS constraints and controls congestion in smart grids. However, the proposed strategy did not consider the type of packets used. A risk-aware route planning mechanism for an SDN based on an evolutionary algorithm was proposed in [13]. The work in [14] investigated various traffic flow routing strategies while considering various service classes required to increase the system's dependability. Additionally, a content-aware queuing algorithm was adopted to maximize capacity for different types of traffic. In the queue management system detailed in [14], lower priority packets were discarded to prioritize data packets with higher priorities. In [15], the authors proposed an SDN-based routing system that achieved low end-to-end delay with high delivery ratios by employing global load-balanced routing in an advanced metering infrastructure (AMI) communication network. A fog-enabled smart grid data transfer approach was proposed in [16] based on the Dijkstra algorithm.

C. PREDICTION IN DATA NETWORKS

To the best of the authors' knowledge, no existing work has explored predicting ED traffic or utilized such predictions in the routing process within the cyber layer. Previous studies have predominantly concentrated on forecasting traffic intensity in data networks. One approach for forecasting sensory input in wireless sensor networks was proposed in [17], which employed a 1-D CNN with a bi-directional LSTM. A model proposed in [18] to predict the network's future status to proactively reduce congestion problems. A hybrid model combining LSTM and MLP was utilized in [19] to estimate network traffic based on previous traffic observations; however, this work did not consider traffic classes that may display stochastic arrival patterns. In addition, the work in [20] presented a deep learning model for predicting future traffic loads and congestion events in SDN-based Internet-of-Things (IoT) networks. The authors in [21] introduced a model for predicting data traffic in a cyber-physical smart grid using GNN. The model utilizes a unique dataset of emergency events (i.e., ED packets) in the smart grids.

D. PREDICTION IN TRANSPORTATION NETWORKS

We examine prediction models in transportation networks because of the resemblance between the data and transportation networks. In particular, data packets and routing links can be similar entities to roads and vehicles in this analogy. In [22], a temporal CNN model was proposed for short-term traffic forecasting while capturing the temporal and spatial characteristics of the traffic flow. The work in [23] presented a city-wide deep-learning prediction model of traffic congestion based on image data. The work in [24] proposed a hybrid deep learning method for long-term traffic forecasting based on wavelet decomposition and a CNN-LSTM model. Additionally, in [25], LSTM was adopted to extract spatial-temporal dependencies while estimating future traffic flow.

E. DEEP Q-NETWORKS IN ROUTING SOLUTIONS

Deep Q-learning (DQL) was adopted in [26] as a potential routing scheme for application in SDN data center networks. In this work, DQN agents were trained to meet the various requirements presented by mice and elephant flows. To address the routing problem, the work in [27] adopted a dual double QL (DDQL) architecture with prioritized memory experience and the ϵ-greedy policy. This design improved learning consistency and addressed the issue of inflated Q-values. In [28], a DQN approach was adopted for routing optimization in SDN using a unique ''conjoint optimization'' mechanism to address the challenge of global routing. A DQL method for solving the global routing problem was proposed in [29]. This method enabled an agent to produce an optimal policy for routing problems by leveraging the conjoint optimization mechanism of deep reinforcement learning.

F. LIMITATIONS

The limitations of existing works can be summarized as follows:

- Traditional Routing: Traditional routing solutions depend on local observations at the routing node. Most of the existing works, e.g., [6], [7], [8], [9], [10], and [11] did not consider the mutual interaction between FS and ED packets and their competition on network resources.

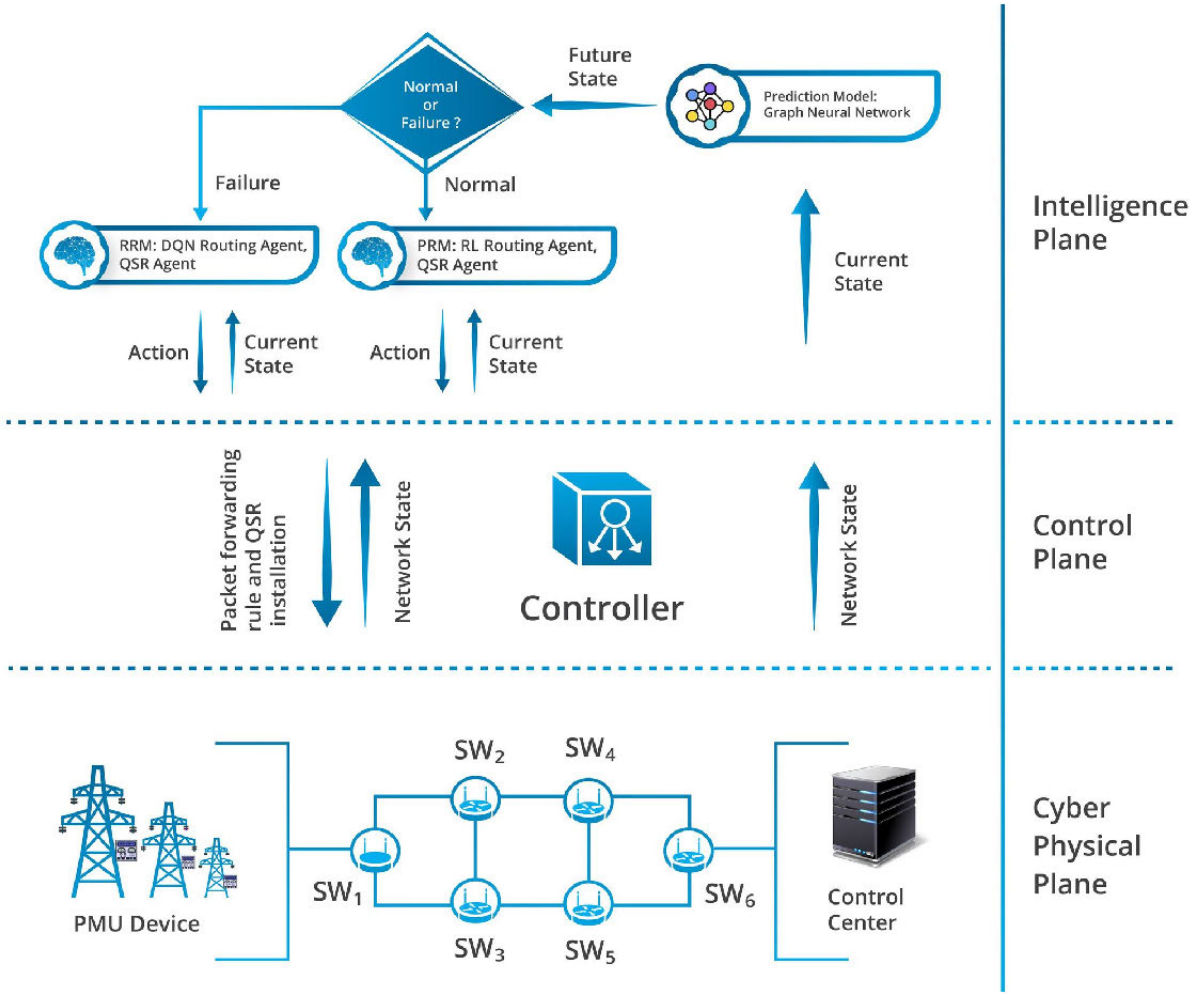


FIGURE 1. Illustration of the system model.

None of these approaches were adaptive to network conditions.

- SDN-based Routing: Although these solutions [12], [13], [14], [15], [16] relied on global observations, they did not adapt to network conditions. Accurate mathematical models are necessary for handling different types of data traffic.
- Prediction Works: Prediction models adopted in [17], [18], [19], [20], [22], [23], [24], and [25] were not directly applicable to the cyber layer of smart grids because there was no ready dataset to train these models. In addition, none of the models are topology-aware models for time-series prediction.
- DQN Applications: DQN used in [26], [27], [28], and [29] are not directly applicable in the cyber layer of the smart grids. In the context of smart grid routing, novel routing data was generated specifically for the application of DQN.

III. SYSTEM MODEL

Figure 1 illustrates the system model and the architecture of the proposed SDN-based resilient, proactive routing strategy. The interconnection between smart grid field devices (e.g., PMUs and actuators) and the cyber plane constitutes a cyber-physical plane. The control plane includes an SDN controller that provides a global perspective of the cyber plane. On top of the control plane, the PRM and RRM modules interconnected with the GNN-based prediction model have been included as the intelligence plane. The details are as follows.

A. CYBER-PHYSICAL PLANE

The components of this plane consist of the physical and cyber layers that are tightly integrated, in addition to the link between the field devices in the physical layer and the OpenFlow switches in the cyber layer. The following are the specifics:

- **Physical Layer:** This plane consists of power nodes and field devices. Power nodes (substations) include power generation and consumption nodes, while field devices include PMUs and actuators. These field devices are installed on the power nodes for real-time monitoring. PMUs are activated to generate a signal for any sensor

update, which is subsequently transformed into a data packet and sent to the control center (CC) over the cyber plane. Similarly, actuators receive signals/data packets from the CC via cyber plane switches. As previously mentioned, these data packets fall into the FS and ED categories. The CC here is the server of the utility office.

- **Cyber Layer:** In a smart grid, the cyber layer is where the CC and the field devices in the physical plane can communicate. The cyber plane must maintain connectivity with the field devices and the CC of the grid. As per the forwarding rule (how the packets will reach the destination) implemented by the SDN controller, OpenFlow switches [3] are specifically engineered within the cyber layer to direct incoming packets to their designated outgoing ports.

B. CONTROL PLANE

As shown in Figure 1, the control plane is connected with the cyber-physical plane through the southbound interface of the OpenFlow protocol [3]. The controller detects the cyber layer topology using the Link Layer Discovery Protocol (LLDP) [30]. The controller will send state query messages to each switch of the cyber layer to collect the current states of the switches regularly. This information includes the status of the flow table, ports, and the number of packets currently in the queues. When a new packet enters the network, the controller will first engage with the intelligence plane to determine the most efficient path for the packet. After that, it will install those paths to the OpenFlow switches along with the path the packet will travel from its origin to its destination. If the type of service (ToS) bit indicates an ED packet, then the optimal route is specified using the ED queue state. If this is not the case, the computation will be according to the FS queue state information.

C. INTELLIGENCE PLANE

The plane shown in Figure 1 consists of the model's intelligent components, which include a GNN-based prediction module, a PRM (RL-based sub-optimal fixed route), and an RRM (DQN-based feasible routes). The plane serves as the core component of the system. The GNN prediction determines the network condition in the cyber layer of the smart grid, indicating whether there is a failure (such as a high congestion violation indicating a node failure) or a normal condition. Under normal conditions, the PRM module operates, while under failure, the RRM module functions as described next.

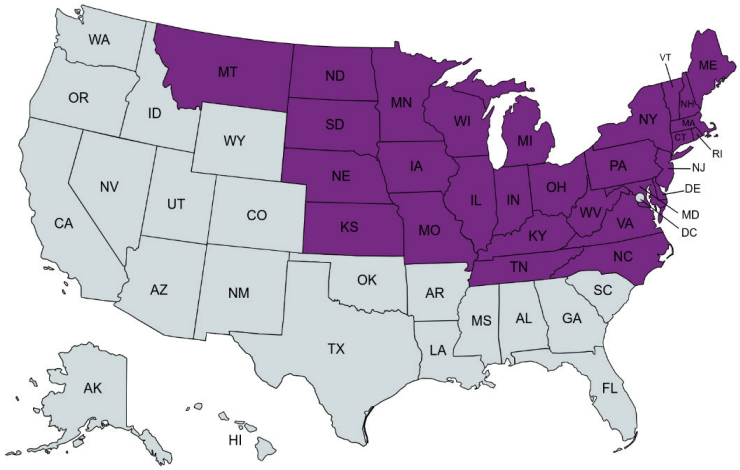
- **Proactive Routing Module (PRM)**: This module works when the cyber layer operates under normal conditions. The QL agent learns to find fixed sub-optimal routes from source to destination. Another QL agent proactively updates the QSR using GNN-based predicted future conditions of the network to reduce congestion in the queues on fixed sub-optimal routes from source to destination. QL-agent lowers the overhead and complexity of operating PRM, making it a lightweight and simple-to-operate system.
- **Resilient Routing Module (RRM):** This module is selected when there is a failure condition in the cyber layer. In this module, the DQN agents learn to find feasible routes under failure conditions in the cyber layer and proactively adjust the QSR to reduce congestion. This module supplies alternative routes to adapt to the failure conditions. The training of the DQN agent considering all the outgoing interfaces of the switches incurs a significant computational burden but is resilient in dealing with failures.

Intelligent routing policies are updated dynamically to reduce the end-to-end latency for ED and FS packets and to maximize the percentage of packets that satisfy the latency threshold under any condition (failure or usual) of the cyber layer of the smart grids. This plane obtains the network state information from the control plane using the northbound interface of the OpenFlow protocol [3]. The GNN-based model predicts the network's future condition using these network statuses. The RL and DQN agents use this information (predicted by the GNN model) to design intelligent routes that avoid congestion and combat failure. To combat failure conditions, reduce congestion, and set an adaptive QSR, the PRM, RRM, and GNN-based prediction models collaborate.

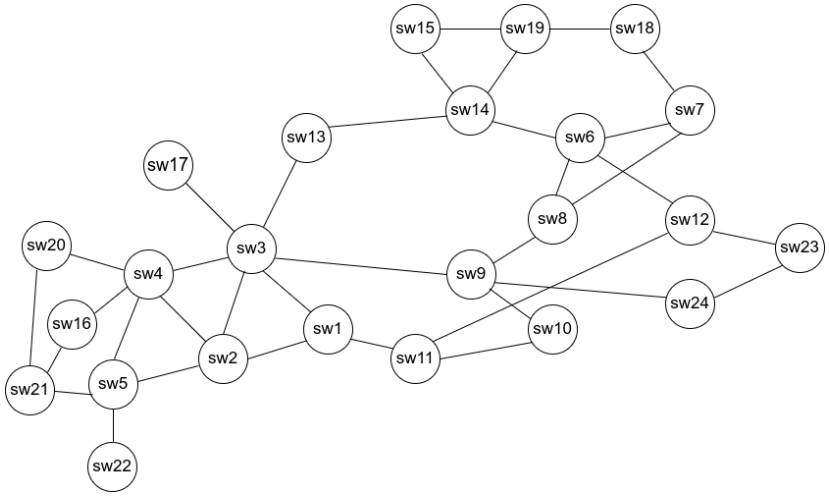
D. DATA TRAFFIC PATTERN AND QUEUING SYSTEM

The data traffic pattern and associated queuing models are explained as follows:

- **Data Traffic Pattern:** The arrival of FS packets is a deterministic process with a defined rate λ_{FS}, whereas the arrival rate of ED packets $\lambda_{\text{ED}}(t)$ is determined by different conditions such as harsh weather, fuel supply shortfalls, physical attacks, etc. [31] during time $t \in \mathcal{T}$. As a result, the ED packet follows a stochastic process that features an arrival rate $\lambda_{\text{ED}}(t)$ that varies over time $\mathcal{T} = \{1, 2, \ldots, T\}$. Let the set of ED and FS packets, represented by $\mathcal{P}_{\text{ED}}(t)$ and $\mathcal{P}_{\text{FS}}(t)$, available during $t \in \mathcal{T}$, respectively. Therefore, the set of total packets in the cyber layer can be found using $\mathcal{P}(t) = \mathcal{P}_{\text{ED}}(t) \cup \mathcal{P}_{\text{FS}}(t)$.
- **Queuing System:** Two hierarchical token buckets (HTB) [32] class high-priority and low-priority queues are set up for each OpenFlow [33] switch to serve ED and FS packets separately. HTB class queues are functional for managing how much outbound bandwidth is used on a particular link because they allow different types of traffic to be sent via various links using a single physical link. HTB uses the token bucket filter algorithm [34] to shape traffic. This algorithm is independent of the interface characteristics. Hence, it does not need to be aware of the outgoing interface's underlying capacity. The maximum buffer size for each ED and FS queue type is represented by the symbol C_{x}, where $\text{x} \in \{\text{ED}, \text{FS}\}$.



(a) The cyber layer in Figure 2b relates to each state colored in purple.



(b) Layout of the cyber layer of the IEEE 39-bus test system [5].

FIGURE 2. An illustration of the cyber layer of the IEEE 39-bus test system and a mapping of emergency events that have occurred in the different states connected to the Eastern Interconnections in the United States of America [35].

IV. PROBLEM STATEMENT

We define the cyber layer of the smart grid as an undirected graph as

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}), \tag{1}$$

where $\mathcal{V} = \{\text{sw}_1, \text{sw}_2, \ldots, \text{sw}_\text{N}\}$ is set of OpenFlow switches and $\mathcal{E} = \{e_{\text{ij}}, e_{\text{jk}}, \ldots\}$ is the set of links that connect those switches. The SDN controller must discover the best route (e.g., $\text{sw}_\text{src} \longrightarrow \text{sw}_\text{i} \longrightarrow sw_\text{j} \longrightarrow sw_\text{k} \longrightarrow sw_\text{cc}$) from source to destination (CC) for a given packet in the cyber layer to meet a predefined QoS for ED and FS packets under both normal and failure conditions. The installed best and next best route must satisfy various latency thresholds based on the types of traffic (ED or FS) while minimizing end-to-end latency and maximizing reliability. Since the ED packets are generated in response to emergency conditions, each ED packet must arrive within the restricted threshold limits. Therefore, it is crucial to minimize the average packet delay to keep it low or eliminate the percentage of ED packets that do not meet the specified criterion.

The end-to-end latency for each data packet is determined by the total queuing and propagation (negligible) delays across all the switches and links from source to destination. The average delay for each type of traffic can be derived by aggregating the per-packet latency and dividing it by the total number of packets available within $\mathcal{T}$. L_{ED} and L_{FS} denote the average end-to-end delay for the ED and FS packets, respectively. The packet loss rate is the ratio of the number of packets dropped to the total number of packets sent to a specific switch for a specific type of packet. The average packet loss ratio is determined for each type of traffic by summing the per-switch packet loss rate and dividing it by the total number of packets within $\mathcal{T}$. D_{ED} and D_{FS} represent the average packet loss ratio for ED and FS packets, respectively.

The resilient routing strategy formed as the optimization problem presented below.

$$\begin{aligned} \min_{\mathcal{A}} \quad & \{(L_{\text{ED}}, L_{\text{FS}}), (D_{\text{ED}}, D_{\text{FS}})\}, \\ \text{subject to}: \quad & L_{\text{ED}} \leq L_{\text{ED}}^{\max}, \quad L_{\text{FS}} \leq L_{\text{FS}}^{\max}, \\ & D_{\text{ED}} \leq D_{\text{ED}}^{\max}, \quad D_{\text{FS}} \leq D_{\text{FS}}^{\max}. \end{aligned} \tag{2}$$

The desirable routing strategy aims to discover the best routing paths (with alternative paths) and a service rate for the ED and FS queues at the OpenFlow switches, such that the average latency and packet loss rate satisfy $L_{\text{ED/FS}}^{\max}$ and $D_{\text{ED/FS}}^{\max}$, which denote the latency thresholds on the average latency and packet drop, respectively, for ED and FS traffic. The standard range for latency is typically between 20 and 200 milliseconds, and a high level of reliability, with a packet loss rate of up to 99.99%, is required [36]. An additional (neutral) constraint for each switch is the number of incoming and outgoing traffic equals. It is the default constraint that the vendor design specification guarantees. The proposed routing strategy aims to maintain equal incoming and outgoing packet flows at OpenFlow switches by implementing dynamic traffic allocation based on real-time grid response, alternative path discovery, traffic prediction, proactive management, service rate adjustment, and vendor-specific constraints.

V. PROPOSED RESILIENT PROACTIVE ROUTING

The proposed routing strategy will determine the routing option based on the GNN prediction update under normal or failure conditions. Under normal conditions, the PRM utilizes a GNN and a QL agent to fix sub-optimal routes for efficient packet delivery. In the event of failure, the RRM combines GNN and DQN to choose viable routes in the cyber layer of the smart grids. Both modules update the QSR using QL-agent for adaptive QSR based on the arrival rate of the packets while avoiding congestion. The routing operation for normal conditions will be selected, following the approach used in our prior work [4] i.e., PRM. For the failure event, the DQN agents will require the current and future conditions of the network to select the path that is most efficient in terms of end-to-end latency per packet while having alternative paths that are next-best at the time of failure conditions in the cyber layer of the smart grid. It is essential to develop a prediction model to forecast the future conditions of ED traffic, providing DQN agents with insights regarding future conditions (congestion/failure). A dataset representing the ED traffic pattern is necessary to train the prediction model. The trained prediction model will be able to offer future state information so that the DQN agents can be adaptive to both the current and future states of ED traffic to find the paths that avoid and alleviate congestion while bypassing failure nodes. This results in the assimilation of data generation, the development of prediction models, and the DQN routing mechanism, enabling a comprehensive resilient-proactive routing strategy detailed below.

A. DATA GENERATION AND ENHANCEMENT

In our earlier work [21] on the traffic data prediction in cyber-physical smart grid, the *Electric Emergency and Disturbance Report* [31] of USA was used to generate ED packets across the cyber layer of the smart grid. The report provides several emergency events along with their starting and ending times, affected areas, and event types, such as vandalism, system malfunctions, transmission interruptions, sabotage, severe weather, etc. In [21], we used the emergency events over 5 years from 2017 to 2021 to generate the ED traffic. In this propsed work, we have generated ED traffic data for the cyber layer shown in Figure 2b.

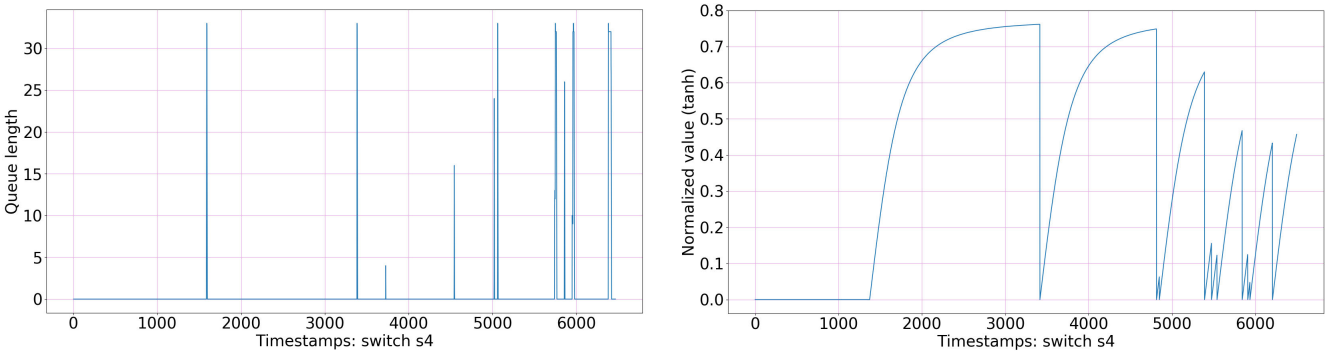
Mininet [37] was utilized to simulate the cyber layers based on the routing protocol of [2]. Next, the number of ED packets waiting at each switch is counted and recorded as time-series data. For example, the number of packets waiting in a queue at the switch is denoted by the label sw_4, depicted in Figure 3a. The duration of five years is represented by the timestamps $0 - 6,468$ in Figure 3a. The queue length represents sparse time-series data because emergencies are rare. For example, Figure 3a shows the queue length data of switch sw_4 related to the emergency occurrences in Mississippi from 2017 to 2021. Due to the sparse aspect of the time-series queue length data, it is arduous to train a machine learning model based on gradient descent optimization to predict the future queue length of ED packets at each switch. Hence, the conversion of sparse data into dense data is required. As FS packets are generated due to regular updates from the grid, a regular arrival pattern with a load of 60 packets per second [38] is used for the FS packets.

1) DATA CONVERSION

It is not essential to make an accurate prediction of the length of the ED packet queue on each switch. Instead, it is sufficient to be aware of the current state of the queue (i.e., congested, as the queue length approaches a certain threshold). When such information is available, congestion can be avoided, which makes proactive routing possible. As a result, the purpose is not to estimate the precise length of the queue; rather, it is to predict the moment when the ED queue will begin to reach the threshold for queue capacity. The sparse queue length data has an embedded time stamp for when an ED queue is congested. As a result, when a certain threshold for queue length is exceeded, we convert the queue length data into a timer indicator. Let x_n denote the queue length at timestamp n, C_{th} is the queue length threshold, then the timing indicator y_n is described as

$$y_n = \begin{cases} y_{\text{n-1}} + 1, & \text{for } x_n \leq C_{\text{th}}, \\ 0 & \text{for } x_n > C_{\text{th}}, \end{cases} \tag{3}$$

According to the definition of (3), y_n is a dense signal unlike x_n. Given the value of y_n, the objective is to predict the value of y_{n+k} for some duration k. This prediction of y_{n+k} can then be interpreted as an indicator of congestion. However, the dense data from this conversion procedure has a high



(a) ED queue length for switch sw_4 in the cyber layer 2b.

(b) Timing indicator sparse data corresponds to Figure 3a.

FIGURE 3. **Illustration of dense timing indicator data for switch *sw4* that corresponds to sparse ED queue length.**

peak-to-average ratio (PAR). If threshold violations occur far apart in time, $\mathbf{y}$ will have a large amplitude. If these events occur near each other, $\mathbf{y}$ will have an extremely low amplitude. This high PAR data may skew the results of the prediction model. Data normalization is a solution that can be used to address this problem.

2) DATA NORMALIZATION

Normalizing $\mathbf{y}$ so that the influence of each event has the same amount of impact on the overall prediction loss is one way to reduce the PAR. It is possible to apply the hyperbolic tangent (*tanh*) function in such a way that

$$y_{\mathrm{n}} \leftarrow \tanh\left(y_{\mathrm{n}}/y_{\max}\right), \tag{4}$$

where $y_{\max}$ is an estimate of the maximum value of y_n. In contrast to the sparse queue length data displayed in Figure 3a, the normalized timer indicator signal shown in Figure 3b reflects a dense signal. The threshold is set at 80% of the queue's maximum capacity. When the queue length equals or exceeds 80% of the queue's capacity, a congestion event is encountered, as indicated by the falling edge of the signal in Figure 3b.

We generate a dataset of ED traffic patterns by applying the previously mentioned strategy to the cyber layer shown in Figure 2. This dataset is adopted to implement a prediction model and data-driven routing strategy.

B. PREDICTION OF CONGESTION EVENTS USING GNN

A GNN is a deep learning architecture developed specifically for graph-structured data [39]. This type of architecture allows for associated feature vectors to be assigned to the nodes, edges, and the entire graph itself. The most necessary property of GNN is its ability to simultaneously capture features at both the adjacency level and the node level, which allows it to capture spatiotemporal features in time-series prediction.

The cyber layer shown in Figure 2b can be represented as (1). For a given switch sw_i, the feature is determined by the timer indicator signal $\mathbf{y}$, e.g., as shown in Figure 3b for switch sw_4. On the other hand, the connectivity feature is described by the adjacency matrix given in (5).

$$A_{ij} = \begin{cases} 1 & \text{if } sw_{\mathrm{i}} \rightarrow sw_{\mathrm{j}}, \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

where sw_{i} and sw_{j} are OpenFlow switches in the cyber layer. The Laplacian of graph $\mathcal{G}$ in (1) can be formulated as

$$L = I_n A D^{-1}, \tag{6}$$

where I_n is the identity matrix, A is the adjacency matrix in (5) while D is the diagonal degree matrix such that $D_{ii} = \sum_j A_{ij}$. The eigendecomposition of L can be represented as $L = U\Lambda U^{\mathrm{T}}$ where U and Λ stand for the matrix of eigenvectors and the diagonal matrix of eigenvalues, respectively, and T stands for the transpose operator.

The graph signal $\mathbf{y}$ (switch features) is filtered by a kernel g_θ resulting in $\mathbf{z} = g_\theta *_{\mathcal{G}} \mathbf{y} = g_\theta(U\Lambda U^{\mathrm{T}})\mathbf{y} = U g_\theta(\Lambda) U^{\mathrm{T}}\mathbf{y}$ where $*_{\mathcal{G}}$ denotes the graph convolution operator, $g_\theta(\Lambda) = \mathrm{diag}(\theta)$ is a nonparametric kernel, and θ is a vector of Fourier coefficients [40]. However, the nonparametric kernel is nonlocalized and presents high computational complexity. Hence, g_θ is approximated by D-localized Chebyshev polynomials [40], and hence, we have

$$\mathbf{z} = \sum_{d=0}^{D-1} \theta_d T_d(L)\mathbf{y}, \tag{7}$$

where θ_d denotes the learned Chebyshev coefficients, the Chebyshev polynomial of order d is $T_d(L) = 2L \circ T_{d-1}(L) - T_{d-2}(L)$, with $T_0(L) = I_n$, $T_1(L) = L$, and $\circ$ is the Hadamard (element-wise) matrix product. In practice, a rescaled version of L, namely, $2L/\lambda_{max} - I_n$ is considered instead of L with λ_{max} being the largest eigenvalue of L. This is to confirm the orthogonality of the basis $T_d(L)$.

In this paper, a multi-layer GNN model is adopted where the input graph signal $\mathbf{y}$ is filtered by a Chebyshev according to (7), then a ReLu activation function is adopted, followed by another Chebyshev layer and ReLu function, etc. Finally, a dense layer is adopted to predict $\mathbf{y}$ a 3 step ahead to install route and adaptive QSR in advance. The optimal hyperparameters are given in Section VI-C.

To compare, we also investigate a collection of topology-unaware prediction models. In these models, the interactions (adjacency matrix/connectivity) between the switches are not considered, but the timer indicator signals $\mathbf{y}$. Deep prediction models adopted in [21] based on MLP, LSTM, and CNN are considered, with each model trained to employ $\mathbf{y}$ data from all the switches. The hyperparameter optimization procedures are described in Section VI-C.

C. PRM: PROACTIVE ROUTING STRATEGY

Under normal conditions, the PRM utilizes a GNN and a QL agent to fix sub-optimal routes for efficient packet delivery. PRM updates the QSR using QL-agent for adaptive QSR based on the arrival rate of the packets while avoiding congestion. The routing operation for normal conditions will be selected, following the approach used in our prior work [4] i.e., PRM. We train two Q-learning agents: one to identify sub-optimal paths from the source to the destination and another to determine the adaptive queue service rate. The algorithm 2 outlines the process of training the QL agent for the QSR setting. The process for training QL agents for routing actions is summarized as follows.

1) STATE

The state is defined by the collective queue status of all OpenFlow switches in the cyber layer and the arrival state of the packets that can be divided as the low, medium, or high arrival rate. The elaborate procedure has been explained in [4].

2) ACTION

The routing action encompasses all the possible routes from the source to the destination, while the QSR action represents the discrete levels of adjustable QSR at a switch.

3) REWARD

To compute end-to-end delay, the queuing delay is the major delay whereas the delay caused by the communication link itself is negligible in comparison to the overall delay per packet. Therefore, the reward function is formed to prioritize the depletion of the queue for the designated switch.

The GNN prediction model is used to forecast the forthcoming congestion state of the cyber layer. The QSR agent updates the QSR of the queue at a specific switch based on predicted future congestion. Once adequately trained, the routing agent, QSR setting agents, and GNN prediction model collaborate.

D. RRM: DQN-BASED RESILIENT ROUTING STRATEGY

DQN is a method that combines QL and deep neural networks (DNN) to provide a good generalization capacity [41]. DQN uses DNN, such as CNN, to estimate the Q-value rather than a Q-tabular form. In this sub-section, we begin by elaborating on how DQN is used to formulate resilient routing as defined in Section IV to generate routes that satisfy the latency threshold for ED and FS packets in smart grids. Since the arrival rate of ED packets varies based on whether or not an emergency event is occurring and the severity of the event, it is not guaranteed that the chosen route will meet the intended latency and packet loss thresholds for each traffic type. It is necessary to mitigate congestion to minimize the latency experienced by each packet. The route must be reinstalled to reroute a packet from a congested path to a less congested path. It is evident that route re-installation from the controller is required to avoid congestion/failure; nevertheless, this route installation results in additional latency, which may also increase the latency per packet and violate the latency thresholds. Consequently, we intend to predict congestion/failure using a GNN-based time-series prediction model, enabling the SDN controller to install the route in advance and eliminating the route installation delay. In addition, we train another QL agent to proactively reconfigure the adaptive QSR to mitigate the queue's underlying congestion. The DQN routing strategy is represented by the relational tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{S}')$, where

- $\mathcal{S}$ is the state space defined by the queue's current and future predicted state,
- $\mathcal{A}$ is the action space defined by the possible route from source to destination,
- $\mathcal{R}$ is a reward function,
- $\mathcal{S}'$ is the updated state after the action is carried out.

1) STATE

The state instances at time t are reflected by the present and future states of all of the queues for ED packets on the switches that comprise the cyber layer of the smart grid. The state instances can be represented by

$$\begin{aligned} s &= [s_{\text{sw}}, \bar{s_{\text{sw}}}], \\ s_{\text{sw}} &= [\rho_{(\text{ED},\text{sw}_{(1,\text{K})})}, \rho_{(\text{ED},\text{sw}_{(2,\text{K})})}, \ldots..\rho_{(\text{ED},\text{sw}_{(\text{N},\text{K})})}], \\ \bar{s_{\text{sw}}} &= [\boldsymbol{\rho}_{(\text{ED},\bar{\text{sw}}_{(1,\text{K})})}, \boldsymbol{\rho}_{(\text{ED},\bar{\text{sw}}_{(2,\text{K})})}, \ldots.\boldsymbol{\rho}_{(\text{ED},\bar{\text{sw}}_{(\text{N},\text{K})})}], \end{aligned} \quad (8)$$

where s_{sw} and $\bar{s_{\text{sw}}}$ denotes the current and future queue state of all switches in the cyber layer, respectively, $\rho_{(\text{ED},\text{sw}_{(1,\text{K})})}$ is the state of the ED packet queues on all of the interfaces linked to switch sw_1's neighboring switches, K is the number of those neighboring switches, N denotes the number of switches in the cyber layer, and $\boldsymbol{\rho}_{(\text{ED},\bar{\text{sw}}_{(1,\text{K})})}$ represents the predicted state of $\rho_{(\text{ED},\text{sw}_{(1,\text{K})})}$. To construct the state space, we first address the issue of data sparsity by converting the queue length data into a timing indicator utilizing (3). Next, we normalize the data utilizing (4) to address the PAR previously discussed in this section. To compute the future state, we adopt a prediction model based on GNN. The details of this model are presented in Section V-B.

2) ACTION

The action space $\mathcal{A}$ is defined based on two action set $\mathcal{A}_{\text{r}}$ and $\mathcal{A}_{\text{s}}$ as follows

$$\mathcal{A} = [\mathcal{A}_{\text{r}}, \mathcal{A}_{\text{s}}], \quad (9)$$

where $\mathcal{A}_s$ denotes the configurable discrete levels of QSR for the queues on a given switch, and $\mathcal{A}_r$ can be represented as

$$\mathcal{A}_r = \{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_{\mathcal{P}}\}, \tag{10}$$

where action α_i is any feasible path.

Feasible Path Search (FPS): This algorithm produces the routing action space $\mathcal{A}_r$ with $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, sw_{src}, sw_{cc}, and $\mathcal{P}$ as inputs. In the cyber layer of smart grids, we use Yen's technique [42] to determine the $\mathcal{P}$ loopless shortest paths that lead from source switches to the destination. When $\mathcal{P}$ is sufficiently large, the action space will have many pathways. Nevertheless, the expense of training grows in proportion as $\mathcal{P}$ increases.

3) REWARD

The objective stated in (2) is to be optimized by the DQN agents while complying with the associated constraints. In light of the definitions of the state space and action space, we propose two distinct reward functions: one for discovering the optimal route from the source to the destination and another for the adaptive discrete level of QSR by the QL agent. We have

$$\mathcal{R}_{ED} = \begin{cases} \eta_1 & \text{if } L_{ED} \leq L_{ED}^{max}, \\ -\eta_1 & \text{otherwise}, \end{cases} \tag{11}$$

$$\mathcal{R}_{FS} = \begin{cases} \eta_2 & \text{if } L_{FS} \leq L_{FS}^{max}, \\ -\eta_2 & \text{otherwise}, \end{cases} \tag{12}$$

where L_{ED} and L_{FS} represent the time needed to transit from the source to CC for ED and FS packets, respectively, L_{ED}^{max} and L_{FS}^{max} denote the latency thresholds for ED and FS, respectively, and η_1 and η_2 are big numbers.

The following is the definition of the reward function used to train the QL agent for configuring adaptive queue service rate:

$$\mathcal{R}_{SR} = \eta_3(\mathcal{C}_{ED} - \rho_{ED}), \tag{13}$$

where $\mathcal{C}_{ED}$ and ρ_{ED} are maximum queue capacity and current queue length, respectively. η_3 is a large integer. The reward function defined in (11) promotes the routing mechanism that discovers the route that avoids congestion and minimizes the end-to-end delay by offering a significant reward. In addition, the reward function in (13) benefits the routing mechanism that empties the ED and FS queues as quickly as possible, as this mitigates congestion and decreases the average packet loss rate.

E. RESILIENT PROACTIVE ROUTING

Resilient proactive routing operates in dual modes: proactive routing mode (usual condition) and resilient routing mode (failure). In proactive mode, the PRM module is used as in our prior work [4] and summarized in Section V-C. In RRM, the DQN agents' training is based on the QL algorithm, which uses deep CNN to map the relationship between state and action and achieve an expedited solution to the issue of large-scale system state. As was stated before, our goal is to enable rerouting and reconfiguring of the QSR based on current and future conditions of the cyber layer of the smart grid. This allows us to avoid congestion and better address node failures. Therefore, to mitigate the extra delay introduced by the installation of forwarding rules, which results from rerouting and reconfiguration, it's imperative to perform rule installation in advance. To do so, a GNN prediction model is used to determine the future network condition (ED traffic congestion or failures in the cyber layer) based on the current state of the network. If the congestion persists for a duration that exceeds the QSR updating period (5 seconds), a failure condition occurs in the cyber layer. If the length of congestion is greater than the QSR interval, then this is considered a congestion condition. According to this cyber-layer condition, the routing mode is swapped between proactive (i.e., PRM) and resilient (i.e., RRM). The DQN agent chooses an alternative route for any failures along with the optimal path. The QSR agent updates the QSR of the queue in the switches along the path from source to destination based on the predicted congestion event. The SDN controller will then reinstall the future route and configure the QSR accordingly. As a result, we train DQN agents to seek out the best route and QL agents to provide an adaptive QSR.

1) LEARNING FOR DQN ROUTING

In QL, by executing an action a_t at a state s_t, an agent learns the expected discounted cumulative cost $Q(s_t, a_t)$, which is called Q-value. For a given current state s_t, the agent chooses its action a_t that exhibits the maximum Q-value among all possible actions in its action space $\mathcal{A}$. DQN updates the value function as follows

$$\begin{aligned} Q(s_t, a_t) \leftarrow\ & Q(s_t, a_t) \\ & + \alpha(r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)), \end{aligned} \tag{14}$$

where α and γ are the learning rate and discount factor, respectively, and $Q(s_t, a_t)$ and $Q(s_{t+1}, a_{t+1})$ are the Q-value of current time t and next timestamp $t+1$, respectively.

Instead of looking for Q values in the Q-table, the DQN algorithm uses a deep neural network like CNN to estimate $Q(s_t, a_t)$, i.e., $Q(s_t, a_t, \theta) \approx Q(s_t, a_t)$, where θ stands for the parameters of the neural network that are the sets of weights and biases. A loss function is as follows

$$\begin{aligned} y_t &= r_{t+1} + \gamma Q(s_{t+1}, (\arg\max_{a_t \in \mathcal{A}_\nabla} Q(s_t, a_t, \theta)); \theta^-), \\ Loss(\theta) &= (y_t - Q(s_t, a_t; \theta))^2, \end{aligned} \tag{15}$$

where y_t represents the desired Q-value, whereas $Q(s_t, a_t; \theta)$ denotes the current best estimate for Q-value; our objective is to bring the current best estimate for Q-value as close as possible to the desired Q-value. We use two separate neural networks, assessed Q-network and target Q-network, which have the same basic structure to generate the two distinct kinds of Q-value. The former produces the estimated Q-value based on the present and future conditions. Every

episode has a different set of parameters to lessen the loss. The latter prepares the target Q by producing a Q-value corresponding to the subsequent condition. Every few steps, the parameters are updated with the assessed Q-network. The DQN architecture includes an experienced buffer that compiles and saves historical information from which the neural network is trained using a random selection process.

Algorithm 1 DQN Training for Path Selection

Input: α, γ
1 **Variables:** $\mathcal{A}_{\mathrm{r}}$
2 $\mathcal{G}(\mathcal{V}, \mathcal{E}) \leftarrow$ Open-Flow Network Discovery
3 $\mathcal{A}_r$=FPS($\mathcal{G} = (\mathcal{V}, \mathcal{E})$, sw_{src}, sw_{cc}, P)
4 Initialize experience buffer E to capacity C
5 Initialize current and target action-value functions with random weights θ and θ^-, respectively
6 Initialize state s based on the current and future condition as given in (8)
7 **repeat**
8 Based on the probability ϵ, select an action a_t randomly from action space A_r; otherwise select from $a_t = \arg\max_{a_t \in \mathcal{A}_\nabla} Q(s_t, a_t, \theta)$
9 Select an action from action space $\mathcal{A}_r$ randomly
10 Execute the selected action a_t enabling SDN controller to install forwarding rule on the switches
11 Obtain updated network state for computing s_{t+1} and R_{t+1}
12 Append the interaction data tuple $(S_t, A_t, R_{t+1}, S_{t+1})$ in experience buffer E
13 Select random mini-batch e from E
14 Set target $y_t = r_{t+1} + \gamma Q(s_{t+1}, (\arg\max_{a_t \in \mathcal{A}_\nabla} Q(s_t, a_t, \theta)); \theta^-)$
15 Perform a gradient descent step on $(y_t - Q(s_t, a_t; \theta))^2$ with respect to θ
16 Set $\theta^- = \theta$ every L step
17 **until** *new packet;*

As shown in Algorithm 1, the agent in step 2 uses OpenFlow Network Discovery to find the network topology $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Step 3 involves triggering the FPS to establish the action space $\mathcal{A}$ for the provided topology $\mathcal{G}(\mathcal{V}, \mathcal{E})$ (for more details, see FPS under the definition of action in Section V-D2). During steps 3 and 4, the agent performs initialization of the E buffer. After that, it sets the default values for the current value function $Q(s_{\mathrm{t}}, a_{\mathrm{t}}; \theta)$ and the desired value function $Q(s_{\mathrm{t}}, a_{\mathrm{t}}; \theta^-)$, accordingly. These are approximated by the deep neural networks using θ and θ^- as their parameters. In step 5, the controller supplies state information to initialize the state space.

2) LEARNING FOR QSR UPDATE

To train the QL-agent that specifies the queue service rates, the arrival rate of the ED packet $\mathcal{S}_{\mathrm{ar}}$ is monitored, and actions $\mathcal{A}_{\mathrm{s}}$ are selected, which update the Q-value Q_{SR} as

$$Q_{SR}(s_t, a_t) \leftarrow (1 - \alpha_{SR}) Q_{SR}(s_t, a_t) + \alpha_{SR}(R_{ED}(s_t, a_t) + \gamma_{SR} \max_{a_{t+1} \in \mathcal{A}_f} Q_{SR}(s_{t+1}, a_{t+1})), \tag{16}$$

where α_{SR} and γ_{SR} are the learning rate and discount factor, respectively, and $s_t \in \mathcal{S}_{ar}$ and $a_t \in \mathcal{A}_s$. Since the arrival of FS packets is a deterministic process with a fixed rate while the arrival of ED packets is a stochastic process with a varying rate, the adaptation of the queue service rates is based on R_{ED}.

Algorithm 2 QL Training for Service Rate Update

Input: $\alpha_{\mathrm{SR}}, \gamma_{\mathrm{SR}}$
1 **Variables:** $Q_{\mathrm{SR}}, \mathcal{A}_{\mathrm{s}}$
2 Initialize: $Q_{\mathrm{SR}} \leftarrow$ All zero
3 **repeat**
4 Select an action from action space $\mathcal{A}_{\mathrm{s}}$ randomly
5 Compute R_{ED} using (13) and monitor the next state
6 Update Q-value Q_{SR} using (16)
7 **until** *training is complete;*

The training of the QL agents for adaptive QSR is based on Algorithm 2. The Q-value Q_{SR} is initialized with zeros and an action is selected at random from the action space $\mathcal{A}_{\mathrm{s}}$. The reward is then computed using (13) and the Q-value Q_{SR} is updated using (16). Finally, the Q-values Q_{SR} are used as the trained agents for adaptive QSR on the switches, as detailed next.

F. PREDICTION BASED ROUTING ALGORITHM

Algorithm 3 consists of the summary of the proposed resilient proactive (i.e., PRM, RRM) routing strategy. The controller collects the network state and then determines if the condition of the cyber layer is normal or a failure. For normal conditions, the controller generates the forwarding rules using PRM and configures the QSR using Q_{SR} for both ED and FS packets. For the failure condition, the RRM operates where the DQN agents supply the next best alternative path to forward the packets from the source to the destination. In addition, configures the QSR using Q_{SR} as in PRM. Following that, the SDN controller installs the forwarding rules on the switches along the path from the source to the destination.

G. COMPLEXITY ANALYSIS

This subsection analyzes the complexity of the scalability of the proposed routing scheme. The proposed approach consists of two steps: (a) utilizing GNN to predict the network condition, specifically congestion or failures, and (b) making routing decisions using QL during normal conditions and DQN during failures.

To begin with the GNN prediction model, we initially find the number of trainable parameters. Each K-localized Chebyshev layer l has c_l channels for $1 \leq l \leq L$, which presents $K \times c_{l-1} \times c_l$ Chebyshev coefficients and c_l bias terms with N_r is neighborhood order. The final dense layer has $|\mathcal{V}| \times c_L$ dense weights and bias terms. Therefore, the GNN prediction model has a total number of parameters as

$$\Sigma = K \sum_{l=1}^{l=L} ((c_{l-1} + 1) \times c_l) + |\mathcal{V}| \times c_L + 1. \tag{17}$$

Algorithm 3 Resilient Proactive Routing Algorithm

Input: Δ
Variable: t
repeat
  repeat
    Collect the current and future state
    Determine whether the network condition is normal or failure
    if *normal* **then**
      Determine the best route using PRM
    end
    if *failure* **then**
      Determine the best alternative route using RRM
    end
  until *new packet;*
  if Δ *slots have elapsed* **then**
    Estimate the arrival rate of ED packets
    Set the QSR using Q-value Q_{SR}
  end
  Inform the switches about the forwarding rule
  Each switch forwards packets based on the forwarding table
  $t = t + 1$
until $\mathcal{T}$ *is complete;*

The equation in (17) shows that the number of parameters in the GNN model is independent of the bus size $|\mathcal{V}|$, except for the last dense layer. Furthermore, the number of parameters varies linearly with N_r, the size of the neighborhood. Additionally, the final dense layer scales in a linear fashion with $|\mathcal{V}|$, the bus size.

For the QL-based routing decision under normal conditions, the complexity of the Q-learning algorithm is $O((|\mathcal{S}|^2 \cdot |\mathcal{A}|)/({\epsilon_Q}^3(1-\gamma)^3))$ [4] where $|.|$ corresponds to the cardinality of space. In this proposed routing system, it approximates to $O((|\mathcal{S}_{\text{sw}} \times \mathcal{S}_{\text{ar}} \times \mathcal{S}_{\text{c}}|^2 \cdot |\mathcal{A}_{\text{r}} \times \mathcal{A}_{\text{s}}|)/({\epsilon_Q}^3(1-\gamma)^3))$ for discount factor γ and ϵ_Q is the exploration-exploitation probability. Hence, the RL strategy presents polynomial complexity with the state and action space.

The primary factor contributing to the computational complexity of the RRM routing module (namely, the DQN-based routing) in the presence of cyber layer failures is the computation of the neural network. During the testing phase, the duration of action computations is done by the design of the neural network used by DQN [27]. The process primarily comprises a sequence of matrix multiplications, which is $O(n_1 \times n_2 + n_2 \times n_3 + \ldots + n_{\text{d-1}} \times n_{\text{d}})$, where n_{i} is the number of neurons in each layer of the neural network and d is the number of layers. In our design $d = 3$, $n_{1=|\mathcal{S}_{\text{t}}|}$ and $n_{\text{d}=|\mathcal{A}|}$. The number of neurons in the remaining layers $(i.e., n_2, n_3 \ldots n_{\text{d-1}})$ is 64.

VI. SIMULATION

A. EXPERIMENT SETUP

We set up the network topology (cyber layer of the IEEE 39-bus test systems [5] in Figure 2) with the help of the Mininet [37] emulator, which offers virtual elements that make it possible to set up a network that is compatible with the OpenFlow protocol. Pox [43] is an open-source SDN controller that has been implemented and used to run the network. The traffic is generated using socket programming based on the translated timing of real-time emergency events as stated in [31]. We used Tensorflow-Keras to implement all the prediction models. Torch is used to implement DQN agents. In the cyber layer of the IEEE 39-bus test system, switch sw_3 is assumed to be the CC server. The collected reports for 2017-2021 [31] are converted into 100 minutes. The first 60 minutes of the simulation are dedicated to data generation and model training. The last 40 minutes are spent evaluating the system with PRM and RRM. The arrival pattern for ED packets is based on the real-time emergency events discussed in Section V-A. A regular arrival pattern with a load of 60 packets per second [38] is used for the FS packets. The latency thresholds range from 20 to 200 milliseconds, which conform to most of the QoS standards established for network performance. These thresholds are crucial for ensuring the reliability of various applications, particularly those involving real-time services and data-processing tasks [36]. The values were determined using criteria that define the permissible latency for preserving optimal performance and customer satisfaction. We set the latency threshold bar at 100 milliseconds for the ED and 120 milliseconds for the FS packets. This will determine the percentage of ED and FS packets that failed to reach the control center within the latency threshold. The target reliability is 99.99% [36], the standard for crucial networking services and guarantees that data is not distorted during transmission. The hardware used for this study has the following configuration: memory 500 GB, AMD EPYC 7402 24-Core Processor × 96, OS 64-bit (Ubuntu 20.04.4 LTS).

B. BENCHMARK PREDICTION MODELS FOR ED TRAFFIC STATE

For comparison's sake, we also investigate several prediction models (MLP, LSTM, CNN detailed in [21]), which are developed for all ports to predict the network state in the cyber layers shown in Figure 2b. The spatial information recorded by the interaction (adjacency matrix/connectivity) between the switches is not considered in these models; only the timer indicator signal $\mathbf{y}$ is used at the switches. Each model is trained as in [4]. Then, the only difference in the RRM module is the consideration of all ports' traffic rather than a single port along the sub-optimal route in [4] adopted here as PRM. The following is a basic description.

1) MULTI-LAYER PERCEPTRON

It is an addition to a feed-forward neural network [44]. The input, hidden, and output layers are the three different layers that make up this system [21]. Each layer consists of a collection of neurons. The input layer gets the signal ($\mathbf{y}$) that needs to be processed. To extract relevant information an arbitrary number of hidden layers is formed between the input

TABLE 1. Optimized hyper-parameters.

Cyber Layer of the IEEE 39-Bus Test System Considering All the Ports Per Switch				
Model	Hidden Layers	No. of Neurons	Activation Function	Optimizer
GNN	$\{1, 2, 3, 4\}$	$\{112, 32, 64, 16\}$	ReLu	RMSprop
MLP	$\{1\}$	$\{128\}$	Softplus	Nadam
LSTM	$\{1, 2\}$	$\{112, 112\}$	ELu	Nadam
CNN	$\{1, 2, 3\}$	$\{112, 112, 512\}$	Softsign	Adam

and output layers. Optimal hyperparameters are provided in Section VI-C.

2) CONVOLUTIONAL NEURAL NETWORKS

This model can capture spatial correlation within the input signal $\mathbf{y}$. It consists of a set of convolution layers, pooling layers, and fully-connected layers [21]. Pooling layers are used after convolution layers to carry out a downsampling operation while maintaining the input data quality. The values obtained from the convolution layer are sent to the pooling layer for feature extraction [45]. After the pooling operation, the output data is flattened and forwarded to the dense layer. Eventually, it makes a 1D output sequence. The optimal hyperparameters are detailed in Section VI-C.

3) LONG-SHORT-TERM-MEMORY

This model captures the temporal correlation within the input signal $\mathbf{y}$. It is a variant of recurrent neural network (RNN) that can overcome the problems of gradient vanishing and gradient explosion in RNN [46]. It stores the information in the preamble data and uses it in the subsequent processing. An LSTM network employs *memory cells* and *gates* to remember the long-term dependencies in temporal sequences [47]. It calculates a *hidden state* as detailed in [4].

C. HYPERPARAMETER OPTIMIZATION

During the training phase, in addition to optimizing the parameters of each model while using gradient descent optimization, a random search strategy is used to optimize the hyper-parameters using cross-validation to have the best results possible. Table 1 presents an overview of the optimal hyper-parameters about the activation function, the optimizer, and the number of neurons discovered in each hidden layer. This table includes the optimized parameters for the models trained for the cyber layer of the IEEE 39-bus test system considering all outgoing Ethernet ports. For example, in the GNN model for the cyber layer of the IEEE 39-bus test system, 4 *Chebyshev* layers are adopted with *ReLu* activation functions. For the CNN model, 3 hidden layers with 112, 112, and 512 neurons are adopted. The selected optimizers and activation functions for each model are listed in Table 1.

D. PREDICTION PERFORMANCE METRICS

To assess the performance of prediction models for the cyber layer of the IEEE 39-bus test system in the smart grids, the following metrics are utilized: *valid hits (VH)*, *missed hits (MH)*, and *false triggers (FT)*. As discussed earlier, the time is divided into several discrete intervals, namely, *slot-1*, *slot-2*, *slot-3*, etc, as presented in Figure 4. In time *slot-1* of Figure 4, both the ground truth congestion event-1 and its prediction are hit within the same time slot. This represents a *VH*. On the other hand, event-2 and event-3 were not correctly predicted in the same time slot of their ground truth. This is called *MH*. Finally, events 3 and 4 were predicted in *slot-4* and *slot-5*, respectively, where there is no ground truth event. These are denoted as *FT*.

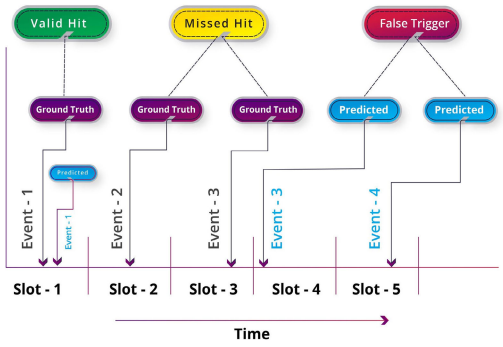
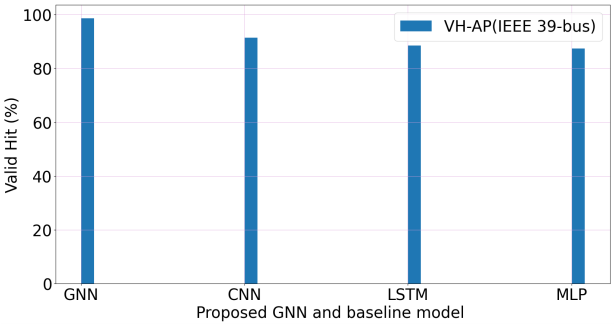


FIGURE 4. Illustration of performance metrics for the prediction model.

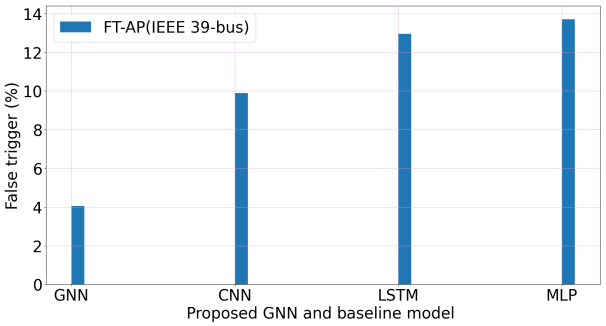
In Figure 5a, we provide the average *VH* rate for total events and all the ports of all the switches. The values of *MH* are excluded because they are complementary to *VH*. By taking into account the feature presenting all the ports on the cyber layer of the IEEE 39-bus test systems, VH for our proposed GNN-based prediction model and baselines (MLP, LSTM, CNN) are provided in Figure 5a. In the results, the GNN-based model for PRM outperforms all the baselines.

In Figure 5b, the prediction performance metric *FT* for the GNN-based prediction model and baseline are depicted. Our proposed GNN-based prediction model beats the baselines (MLP, LSTM, and CNN) because it considers both temporal and spatial features, whereas the baselines only consider temporal features. Because it operates directly on graph-structured data, the GNN-based model is topology-aware while others are not.

The functionalities of PRM and RRM are contingent upon the predictive performance of the GNN-based prediction model. The GNN-based prediction model demonstrates superior performance compared to other models such as



(a) Prediction model valid hit (%) considering all Ethernet ports on the cyber layer of the IEEE 39-bus test system.



(b) Prediction model false trigger (%) considering the cyber layer of the IEEE 39-bus test system considering all Ethernet ports.

FIGURE 5. Performance evaluations of GNN models in comparison to MLP, CNN, and LSTM for RRM.

CNN, LSTM, and MLP. Consequently, the PRM and RRM modules also exhibit better performance.

E. RESULTS, PERFORMANCE, AND DISCUSSIONS

The proposed strategy is evaluated on the cyber layer of the IEEE 39-bus test system. To measure the performance of the proposed approach, we utilize average delay per packet, percentage of packets failing to meet latency thresholds, and packet loss rate as assessment metrics. We tested our proposed routing strategy under two conditions in the cyber layer: normal and failure conditions. We assessed the effectiveness of our proposed routing strategy by comparing it to our previous QL-based routing method as an adaptive baseline and another non-adaptive baseline where the SDN controller employs the Bellman-Ford (BF) algorithm to determine the routing path from the source to the destination. Under failure conditions, our proposed RRM strategy offers QoS routing for smart grids.

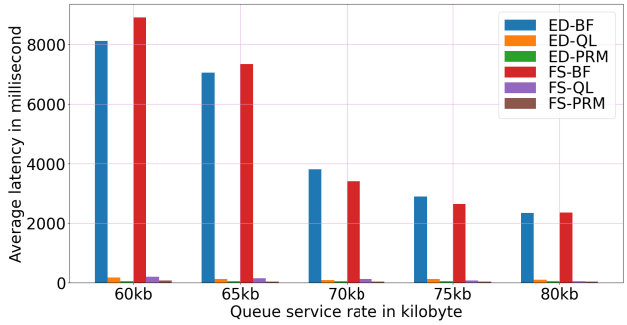
1) RESULTS UNDER NORMAL CONDITION

Figure 6a displays the average delay for ED and FS packets on the cyber layer of the IEEE 39-bus test systems under normal conditions. These results are achieved by the proposed resilient-proactive routing strategy (PRM in conjunction with GNN-based prediction). The results are then compared to the adaptive QL [2] and the non-adaptive benchmark (BF). The x-axis shows the QSR, which remains constant for the non-adaptive benchmark (BF) and indicates the initial QSR for the adaptive benchmark (QL) and our proposed resilient-proactive strategy (PRM). Based on the data presented in Figure 6a, under normal conditions, the mean delay achieved by the proposed mechanism (PRM) on the cyber layer of the IEEE 39-bus test systems is below 57.36 milliseconds for both ED and FS packets. In contrast, the average delay for the non-adaptive benchmark (BF) and adaptive QL strategy increased to 8900 and 206.85 milliseconds, respectively. Hence, the results demonstrate that under normal conditions of the cyber layer, our proposed resilient proactive routing strategy outperforms the adaptive (QL) and non-adaptive approach (BF).

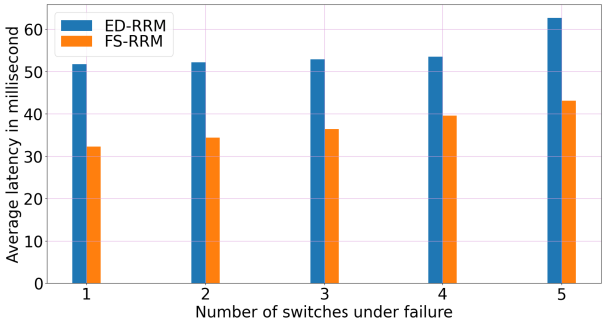
Figure 6c illustrates the percentage of latency not satisfied for ED and FS packets on the cyber layer of the IEEE 39-bus test systems using the proposed resilient-proactive technique (PRM) compared to the adaptive QL [2] and non-adaptive benchmark (BF). Based on the results presented in Figure 6c, the percentage of latency not satisfied with the proposed routing strategy is below 1%. In contrast, the corresponding percentages for the non-adaptive benchmark (BF) and adaptive QL-based strategy on the cyber layer of the IEEE 39-bus test systems are above 28.50% and up to 14.0%, respectively. Hence, the results demonstrate that under normal conditions of the cyber layer, our proposed resilient-proactive routing outperforms the adaptive (QL) and non-adaptive approach (BF).

Figure 6e depicts the average packet loss percentage under normal conditions of the cyber layer for both ED and FS packets on the cyber layer of the IEEE 39-bus test system adopting the proposed resilient-proactive routing strategy (PRM) in comparison to the adaptive QL [2] and non-adaptive benchmark (BF). The target reliability is 99.999% [36]. According to Figure 6e, the average packet loss rate using the proposed strategy (PRM) is 0%, whereas it is up to 24.38% for the non-adaptive benchmark (BF), and nearly 0% for the adaptive QL. Hence, the results demonstrate that in terms of average packet loss percentage, under normal conditions, our proposed resilient-proactive strategy outperforms the adaptive (QL) and non-adaptive approach (BF).

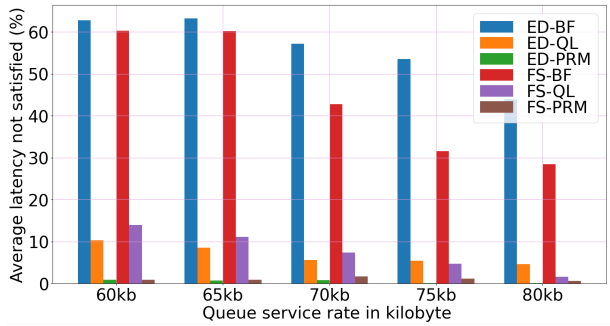
The proposed resilient-proactive (PRM phase) routing demonstrates its advantages under normal conditions of the cyber layer due to its ability to predict congestion, adjust the QSR, and utilize separate queues for different types of ED and FS packets. The resilient-proactive strategy leverages congestion prediction using GNN to eliminate the additional delay required for installing forwarding rules (by SDN controller) on the forwarding switches between the source and the destination. Configuring the adaptive QSR beforehand reduces queuing delay, minimizes the likelihood of packet loss, and enhances system reliability.



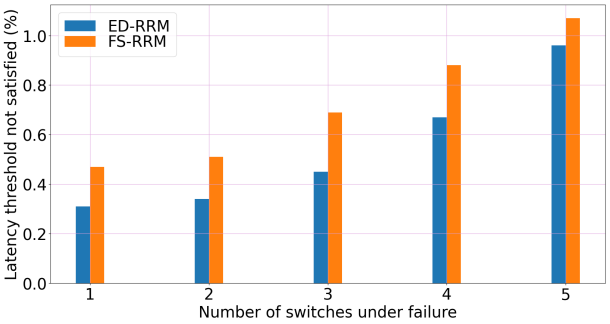
(a) Average latency per packet on the cyber layer of the IEEE 39-bus test system under normal condition



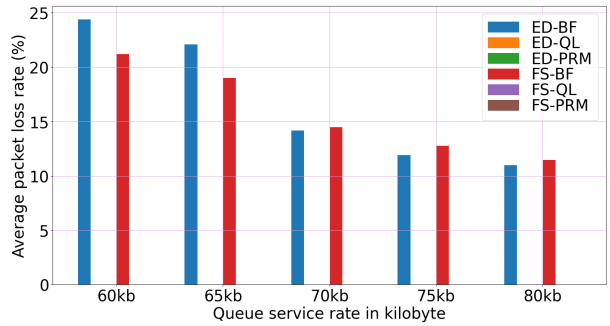
(b) Average latency per packet on the cyber layer of the IEEE 39-bus test system under failure condition



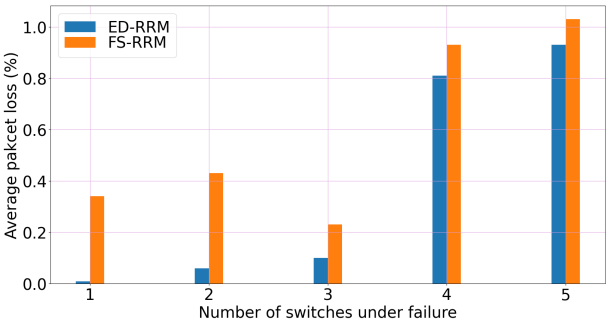
(c) Average latency threshold not satisfied (%) on the cyber layer of the IEEE 39-bus test system under normal condition



(d) Average latency threshold not satisfied (%) on the cyber layer of the IEEE 39-bus test system under failure condition



(e) Average packet loss rate (%) on the cyber layer of the IEEE 39-bus test system under normal conditions



(f) Average packet loss rate on the cyber layer of the IEEE 39-bus test system under failure condition

FIGURE 6. Results analysis on the cyber layer of the IEEE 39-bus test system [5] under both normal and failure condition.

2) RESULTS UNDER FAILURE CONDITION

The forwarding device (OpenFlow switch) may become dysfunctional for various reasons, including physical attacks, cyberattacks, packet jamming, etc. When any forwarding device from the source to the destination cannot work, the route becomes useless. Figure 7 presents the cyber layer of the IEEE 39-bus test system under failure conditions, where the failed nodes are marked using the red cross symbol. As in Figure 7, switch *sw16* is under failure condition. So, the route ($e.g. sw21 \rightarrow sw16 \rightarrow sw4 \rightarrow sw3$) from *sw21* to destination server *sw3* is no more active route. The average latency per packet becomes infinite when using the PRM in conjunction with GNN-based prediction because it relies on a fixed sub-optimal route. Therefore, the non-adaptive benchmark (BF), adaptive QL-based routing, as well as the proactive approach presented in [4] are unable to function during failure conditions since they depend on a fixed sub-optimal route from the source to the destination. Any node failure on the fixed sub-optimal route makes the route unreachable, i.e., the packets will never reach the destination. There is no standardized routing approach that adeptly operates under failure conditions and comparable to our proposed innovative resilient-proactive routing strategy.

Figure 6b, 6d, and 6f present the results when the cyber layer is under failure conditions. The average delay on the cyber layer of the IEEE 39-bus test system under failure conditions is less than 62.67 milliseconds for ED packets and less than 43.13 milliseconds for FS packets.

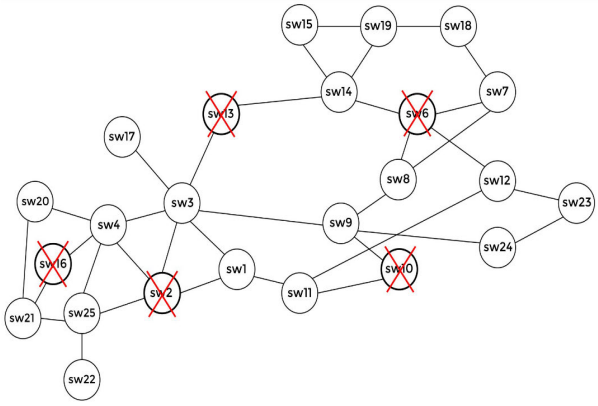


FIGURE 7. Illustration of the cyber layer of IEEE the 39-bus test system under failure. The red-crossed switches (e.g. sw-2, sw-6) are failed devices which means that the paths over that device are no longer active.

This performance is similar to that of the PRM under usual conditions in the cyber layer of the smart grids. Figure 6d demonstrates that the proposed method under failure exhibits an unsatisfied delay percentage of less than 1% for both the ED and FS packets, aligning with the intended QoS targets. In Figure 6f, our proposed strategy (RRM) achieves an average packet loss ratio of approximately 1% in the presence of a failure in the cyber layer of the IEEE 39-bus test system. This performance is slightly higher than that of PRM under normal conditions but still within an acceptable range for ensuring the reliability of the smart grids.

The success of the proposed resilient-proactive routing under failure conditions can be attributed to the DQN agents and GNN-based prediction model. The ability of the DQN agents to generate an alternate route based on the network's present and potential future conditions (congestion or failure) ensures that resilient-proactive routing remains active even under failure. The future condition of the cyber layer of the smart grids is predicted by the GNN-based prediction model, which considers all outgoing port features of all switches in the cyber layer and reduces the additional latency required for setting the forwarding rule on the switches from the source to the destination.

Our proposed software-defined routing strategy balances requirements by adjusting decisions based on traffic types. It prioritizes paths with lower latency and higher throughput for ED packets while prioritizing paths with higher redundancy and lower congestion for FS packets.

VII. CONCLUSION

This paper proposes a resilient proactive routing strategy for smart grids to improve routing efficiency, enhance reliability, and react to failure circumstances. The proposed software-defined routing approach achieves equilibrium by dynamically adapting decisions according to the types of traffic. The strategy prioritizes paths with lower latency and higher throughput for ED packets while preferring paths with more redundancy and lower congestion for FS packets. We developed PRM and RRM using RL and DQN along with a GNN-based prediction model to meet latency thresholds for ED and FS packets under both normal and failure conditions. PRM and RRM function based on the conditions of the cyber layer. When the cyber layer is functioning normally, the PRM operates. If the cyber layer has a failure, the RRM takes over. While the RRM uses DQN agents to create feasible routes under failure, the PRM uses a QL agent for routing decisions and another QL agent for adaptive QSR settings under normal conditions. The results demonstrate that the proposed dual-mode routing method, namely PRM and RRM, effectively guarantees the desired QoS (e.g. normal condition: average latency less than 57.36, unsatisfied latency percent less than 1%, packet loss 0%; failure condition: average latency less than 62.67, unsatisfied latency percent nearly 1%, packet loss nearly 0%) for smart grids, regardless of whether the cyber layer is functioning normally or experiencing a failure.

Directions for Future Research: Although the proposed routing strategy can surmount the failure conditions of the forwarding node, it is not adaptable to controller failure conditions, which could result in a single point of failure within the cyber layer of the grid due to the adoption of the single controller-based SDN paradigm. Consequently, our forthcoming research will formulate a routing strategy that employs several SDN controller-based routing paradigms to handle situations where controllers fail. A multiple-controller-based SDN paradigm consists of several joint controllers. Under the failure of a single controller, another controller can act as an alternative.

ACKNOWLEDGMENT

This publication was made possible by NPRP13S-0127-200182 from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the authors.

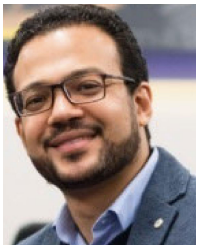
REFERENCES

[1] N. Saputro, K. Akkaya, and S. Uludag, "A survey of routing protocols for smart grid communications," *Comput. Netw.*, vol. 56, no. 11, pp. 2742–2771, Jul. 2012.
[2] M. A. Islam, M. Ismail, R. Atat, and S. Shannigrahi, "Software-defined routing strategy based on reinforcement learning in smart power grid," in *Proc. SoutheastCon*, Mar. 2022, pp. 64–70.
[3] P. Göransson, C. Black, and T. Culver, "The OpenFlow specification," in *Software Defined Networks*, 2nd ed., Boston, MA, USA: Morgan Kaufmann, 2017, ch. 5, pp. 89–136.
[4] M. A. Islam, M. Ismail, R. Atat, O. Boyaci, and S. Shannigrahi, "Software-defined network-based proactive routing strategy in smart power grids using graph neural network and reinforcement learning," *e-Prime Adv. Electr. Eng., Electron. Energy*, vol. 5, Sep. 2023, Art. no. 100187.
[5] R. Atat, M. Ismail, S. S. Refaat, and E. Serpedin, "Stochastic geometry model for interdependent cyber-physical communication-power networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.
[6] A. Abdrabou, "A wireless communication architecture for smart grid distribution networks," *IEEE Syst. J.*, vol. 10, no. 1, pp. 251–261, Mar. 2016.
[7] N. Saputro and K. Akkaya, "Investigation of smart meter data reporting strategies for optimized performance in smart grid AMI networks," *IEEE Internet Things J.*, vol. 4, no. 4, pp. 894–904, Aug. 2017.
[8] Y. Cao, D. Duan, X. Cheng, L. Yang, and J. Wei, "QoS-oriented wireless routing for smart meter data collection: Stochastic learning on graph," *IEEE Trans. Wireless Commun.*, vol. 13, no. 8, pp. 4470–4482, Aug. 2014.

[9] X. Li, Y.-C. Tian, G. Ledwich, Y. Mishra, X. Han, and C. Zhou, "Constrained optimization of multicast routing for wide area control of smart grid," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3801–3808, Jul. 2019.
[10] H. Li, L. Lai, and H. V. Poor, "Multicast routing for decentralized control of cyber physical systems with an application in smart grid," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 6, pp. 1097–1107, Jul. 2012.
[11] S.-G. Yoon, S. Jang, Y.-H. Kim, and S. Bahk, "Opportunistic routing for smart grid with power line communication access networks," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 303–311, Jan. 2014.
[12] Y. Su, P. Jiang, H. Chen, and X. Deng, "A QoS-guaranteed and congestion-controlled SDN routing strategy for smart grid," *Appl. Sci.*, vol. 12, no. 15, p. 7629, Jul. 2022.
[13] B. Liu, P. Yu, F. Chen, F. Chen, Q. Xue-song, and L. Shi, "Risk-aware service routes planning for system protection communication network in energy Internet," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, Apr. 2019, pp. 295–303.
[14] M. Rezaee and M. H. Y. Moghaddam, "SDN-based quality of service networking for wide area measurement system," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3018–3028, May 2020.
[15] A. Montazerolghaem, M. H. Y. Moghaddam, and A. Leon-Garcia, "OpenAMI: Software-defined AMI load balancing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 206–218, Feb. 2018.
[16] F. Y. Okay, S. Ozdemir, and M. Demirci, "SDN-based data forwarding in fog-enabled smart grids," in *Proc. 1st Global Power, Energy Commun. Conf. (GPECOM)*, Jun. 2019, pp. 62–67.
[17] H. Cheng, Z. Xie, Y. Shi, and N. Xiong, "Multi-step data prediction in wireless sensor networks based on one-dimensional CNN and bidirectional LSTM," *IEEE Access*, vol. 7, pp. 117883–117896, 2019.
[18] S. LaMar, J. J. Gosselin, I. Caceres, S. Kapple, and A. Jayasumana, "Congestion aware intent-based routing using graph neural networks for improved quality of experience in heterogeneous networks," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2021, pp. 477–481.
[19] M. Alizadeh, M. T. H. Beheshti, A. Ramezani, and H. Saadatinezhad, "Network traffic forecasting based on fixed telecommunication data using deep learning," in *Proc. 6th Iranian Conf. Signal Process. Intell. Syst. (ICSPIS)*, Dec. 2020, pp. 1–7.
[20] F. Tang, Z. M. Fadlullah, B. Mao, and N. Kato, "An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: A deep learning approach," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5141–5154, Dec. 2018.
[21] M. A. Islam, M. Ismail, O. Boyaci, R. Atat, and S. Shannigrahi, "Graph neural network based prediction of data traffic in cyber-physical smart power grids," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Oct. 2022, pp. 322–327.
[22] W. Zhao, Y. Gao, T. Ji, X. Wan, F. Ye, and G. Bai, "Deep temporal convolutional networks for short-term traffic flow forecasting," *IEEE Access*, vol. 7, pp. 114496–114507, 2019.
[23] N. Ranjan, S. Bhandari, H. P. Zhao, H. Kim, and P. Khan, "City-wide traffic congestion prediction based on CNN, LSTM and transpose CNN," *IEEE Access*, vol. 8, pp. 81606–81620, 2020.
[24] Y. Li, S. Chai, Z. Ma, and G. Wang, "A hybrid deep learning framework for long-term traffic flow prediction," *IEEE Access*, vol. 9, pp. 11264–11271, 2021.
[25] H. Bouchemoukha, M. Nadjib, and Z. A. Lahoulou, "Is classical LSTM more efficient than modern GCN approaches in the context of traffic forecasting?" in *Proc. Int. Conf. Recent Adv. Math. Informat. (ICRAMI)*, Sep. 2021, pp. 1–6.
[26] Q. Fu, E. Sun, K. Meng, M. Li, and Y. Zhang, "Deep Q-learning for routing schemes in SDN-based data center networks," *IEEE Access*, vol. 8, pp. 103491–103499, 2020.
[27] Y.-R. Chen, A. Rezapour, W.-G. Tzeng, and S.-C. Tsai, "RL-routing: An SDN routing algorithm based on deep reinforcement learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 3185–3199, Oct. 2020.
[28] E. H. Bouzidi, A. Outtagarts, R. Langar, and R. Boutaba, "Deep Q-network and traffic prediction based routing optimization in software defined networks," *J. Netw. Comput. Appl.*, vol. 192, Oct. 2021, Art. no. 103181.
[29] H. Liao, W. Zhang, X. Dong, B. Poczos, K. Shimada, and L. B. Kara, "A deep reinforcement learning approach for global routing," 2019, *arXiv:1906.08809*.
[30] *IEEE Standard for Local and Metropolitan Area Networks—Station and Media Access Control Connectivity Discovery*, IEEE Standard 802.1AB-2016, Revision IEEE Std 802.1AB-2009, 2016, pp. 1–146.
[31] U.S. Dept. Energy. *Electric Disturbance Events (DoE-417) Annual Summaries*. Accessed: Oct. 25, 2021. [Online]. Available: https://www.oe.netl.doe.gov/OE417_annual_summary.aspx
[32] M. A. Brown. *Queuing Discipline*. Accessed: May 15, 2021. [Online]. Available: https://tldp.org/en/Traffic-Control-HOWTO/ar01s07.html
[33] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and OpenFlow: From concept to implementation," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2181–2206, 4th Quart., 2014.
[34] D. Medhi and K. Ramasamy, "Traffic conditioning," in *Network Routing* (The Morgan Kaufmann Series in Networking), 2nd ed., Boston, MA, USA: Morgan Kaufmann, 2018, ch. 18, pp. 626–644.
[35] *U.S. Grid Regions*. Accessed: Dec. 15, 2022. [Online]. Available: https://www.epa.gov/green-power-markets/us-grid-regions
[36] M. H. Rehmani, A. Davy, B. Jennings, and C. Assi, "Software defined networks-based smart grid communication: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2637–2670, 3rd Quart., 2019.
[37] Mininet Project Contributors. *Mininet*. Accessed: Jul. 20, 2021. [Online]. Available: http://mininet.org/
[38] G. Barchi, D. Fontanelli, D. Macii, and D. Petri, "On the accuracy of phasor angle measurements in power networks," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 5, pp. 1129–1139, May 2015.
[39] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020.
[40] H. Sahbi, "Learning Laplacians in Chebyshev graph convolutional networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2064–2075.
[41] I. Yilmaz, K. Kapoor, A. Siraj, and M. Abouyoussef, "Privacy protection of grid users data with blockchain and adversarial machine learning," in *Proc. ACM Workshop Secure Trustworthy Cyber-Physical Syst.* New York, NY, USA: Association for Computing Machinery, Apr. 2021, pp. 33–38, doi: 10.1145/3445969.3450431.
[42] E. V. Martins and M. B. Pascoal, "A new implementation of yen's ranking loopless paths algorithm," *Quart. J. Belg., Fr. Italian Oper. Res. Societies*, vol. 1, no. 2, pp. 121–133, Jun. 2003.
[43] R. Jmal and L. Chaari Fourati, "Implementing shortest path routing mechanism using OpenFlow POX controller," in *Proc. Int. Symp. Netw., Comput. Commun.*, Jun. 2014, pp. 1–6.
[44] S. Abirami and P. Chitra, "Energy-efficient edge based real-time healthcare support system," *Adv. Comput.*, vol. 117, no. 1, pp. 339–368, 2020.
[45] R. K. Halder, M. N. Uddin, M. A. Uddin, S. Aryal, M. A. Islam, F. Hossain, N. Jahan, A. Khraisat, and A. Alazab, "A grid search-based multilayer dynamic ensemble system to identify DNA N4-methylcytosine using deep learning approach," *Genes*, vol. 14, no. 3, p. 582, 2023. [Online]. Available: https://www.mdpi.com/2073-4425/14/3/582
[46] Z. Wang, X. Su, and Z. Ding, "Long-term traffic prediction based on LSTM encoder–decoder architecture," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 10, pp. 6561–6571, Oct. 2021.
[47] R. Chandra, S. Goyal, and R. Gupta, "Evaluation of deep learning models for multi-step ahead time series prediction," *IEEE Access*, vol. 9, pp. 83105–83123, 2021.

MD AMINUL ISLAM received the B.Sc. degree in computer science and engineering (CSE) from Patuakhali Science and Technology University, Patuakhali, Bangladesh, in 2009, the Master of Information Technology (MIT) degree from the University of Dhaka, Dhaka, Bangladesh, in 2011, and the Ph.D. degree in engineering (computer science) from Tennessee Technological University, Cookeville, TN, USA. He is currently the Director of the ICT Cell and an Associate Professor with the Department of Computer Science and Engineering (CSE), Jagannath University, Dhaka. His research interests include machine learning, smart power grids, cyber security, software-defined networking, and routing security. Specifically, this pertains to the use of reinforcement learning (RL) and graph neural networks to enhance the security and intelligence of smart power grids in the cyber layer.

RACHAD ATAT (Senior Member, IEEE) received the B.Eng. degree in computer engineering from Lebanese American University, in 2010, the master's degree in electrical engineering from the King Abdullah University of Science and Technology, in 2012, and the Ph.D. degree (Hons.) in electrical engineering from The University of Kansas, Lawrence, KS, USA, in 2017. He is currently an Assistant Research Scientist with Texas A&M University at Qatar. He has authored 22 peer-reviewed journal articles and 30 conference papers in the top venues, with over 1000 citations, with at least nine papers among the most highly cited for their years of publication. He is an investigator in grants on applications of AI in cybersecurity and networking, and currently leading a research team of Ph.D. students with national and international collaborations.



MUHAMMAD ISMAIL (Senior Member, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees in electrical engineering (electronics and communications) from Ain Shams University, Cairo, Egypt, in 2007 and 2009, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2013. He is currently the Director of the Cybersecurity Education, Research, and Outreach Center (CEROC) and an Associate Professor with the Department of Computer Science, Tennessee Technological University, Cookeville, TN, USA. He was a co-recipient of the Best Paper Awards from the IEEE ICC 2014, the IEEE GLOBECOM 2014, the SGRE 2015 and 2024, the Green 2016, and the IEEE IS 2020, and the Best Conference Paper Award from the IEEE Communications Society Technical Committee on Green Communications and Networking for his publication in IEEE ICC 2019. He is the Track Chair of the IEEE Globecom 2024. He was the Track Co-Chair of the IEEE SmartGridComm 2023 and the IEEE VTC 2017 and 2016, the Workshop Co-Chair of the IEEE GreenCom 2018, the Publicity and Publication Co-Chair of the CROWNCOM 2015, and the Web-Chair of the IEEE INFOCOM 2014. He was an Associate Editor of the *IET Communications*, *PHYCOM*, and IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING. He was an Editorial Assistant of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, from 2011 to 2013. He is an Associate Editor of IEEE INTERNET OF THINGS JOURNAL and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He has been a technical reviewer of several IEEE conferences and journals.