Prediction and Proactive Congestion Mitigation in Software-Defined Networks Using a Hybrid Spatio-Temporal GCN–LSTM and Deep Reinforcement Learning Framework

Arshad Mahmood[1], Fawad Nasim[2]
[1]Department of Computer Science, Faculty of Computer Science & Information Technology, The Superior University, Lahore, Pakistan.
[2]Department of Computer Science, Faculty of Computer Science & Information Technology, The Superior University, Lahore, Pakistan.
*__Correspondence__: fawad.nasim@superior.edu.pk

Citation | Mahmood. A, Nasim. F, "Prediction and Proactive Congestion Mitigation in Software-Defined Networks Using a Hybrid Spatio-Temporal GCN–LSTM and Deep Reinforcement Learning Framework", IJIST, Vol. 8 Issue. 5 pp 2297-2310, August 2026
DOI: https://doi.org/10.33411/IJIST/1999
Received | Aug 08, 2026 **Revised |** Aug 20, 2026 **Accepted |** Aug 26, 2026 **Published |** Aug 30, 2026.

Software-Defined Networking (SDN) has emerged as a promising paradigm for centralized network management; however, conventional routing protocols remain predominantly reactive, responding only after congestion has already occurred. This reactive behavior increases end-to-end latency, packet loss, and inefficient bandwidth utilization, particularly under dynamic traffic conditions. This paper presents a hybrid Artificial Intelligence (AI)-driven framework that integrates a Spatio-Temporal Graph Convolutional Network with Long Short-Term Memory (GCN–LSTM) for traffic prediction and a Dueling Deep Q-Network (DQN) for proactive routing optimization. The proposed framework operates within an SDN architecture using the Ryu controller and OpenFlow 1.3 while network scenarios are emulated in Mininet. The GCN component captures spatial dependencies among network nodes, whereas the LSTM models temporal traffic evolution to forecast future congestion. These predictions guide the reinforcement learning agent to proactively update forwarding paths before congestion develops. Experimental evaluation demonstrates significant improvements over conventional routing approaches, including lower prediction error (MAPE 0.94%), reduced average latency (29.1 ms), zero packet loss during burst traffic, and an aggregate throughput of 8.1 Gbps. The proposed framework demonstrates the effectiveness of combining deep learning and reinforcement learning to enable intelligent, predictive, and adaptive traffic management in modern SDN environments. GCN–LSTM MAE of 1.05 Kbps at t+1 and 3.18 Kbps at t+5, RMSE 4.22 Kbps, MAPE 0.94%, average latency 29.1 ms, packet loss 0.00% in the reported burst-traffic experiment, throughput 8.1 Gbps, and Dueling-DQN convergence at approximately episode 350 versus approximately 700 for standard DQN. Calculated comparative improvements were also added.

Keywords: Software-Defined Networking, Graph Convolutional Network, Long Short-Term Memory, Deep Reinforcement Learning, Traffic Prediction, Congestion Control.

Introduction:

Modern enterprise and cloud networks experience continuously increasing traffic volumes due to the rapid growth of cloud computing [1], Internet of Things (IoT), multimedia services, and data-intensive applications [2][3]. Traditional routing protocols such as OSPF and ECMP rely on static or reactive decision-making mechanisms that respond only after congestion becomes evident. Consequently, network performance degrades because routing decisions fail to anticipate future traffic conditions. Software-Defined Networking (SDN) addresses several limitations of conventional networking by separating the control plane from the forwarding plane, thereby enabling centralized network management and global visibility of network resources. Nevertheless, most SDN traffic engineering techniques remain reactive, relying on current network statistics rather than future traffic behavior. Recent advances in Artificial Intelligence (AI) [4][5] provide new opportunities for predictive network management. Deep learning models [6][7] have demonstrated excellent capability in forecasting sequential traffic patterns, whereas reinforcement learning algorithms enable adaptive routing in response to changing network conditions. However, many existing studies consider either temporal traffic prediction or routing optimization independently, limiting their effectiveness in highly dynamic environments.

This study proposes a hybrid AI framework that integrates a Graph Convolutional Network (GCN), Long Short-Term Memory (LSTM), and Duelling Deep Q-Network (DQN) within an SDN architecture. The GCN extracts spatial relationships among interconnected switches, while the LSTM [8] captures temporal traffic evolution. Predicted congestion information is supplied to a reinforcement learning agent [9] that proactively updates routing policies before congestion occurs.

The combination of deep learning and reinforcement learning to achieve intelligent traffic management in SDN has recently been studied. The proposed hybrid graph-temporal models in the literature demonstrated that graph-temporal models could benefit from integrating graph and LSTM to capture the structural and temporal dependencies of networks, such as the GC-LSTM model proposed by [10] for dynamic network prediction. Chen et al. further showed that traffic prediction can also be combined with Deep Q-Network based routing optimization to perform dynamic congestion prediction and enhance network utilization in SDN [11]. The deep reinforcement learning routing method proposed by Chen et al., called RL-Routing, is an SDN routing method that takes into account comprehensive network information, and can optimize the throughput and delay performance of the network compared with the traditional routing method [12]. Likewise, in the field of intelligent routing in SDN, DRL-R suggested a deep reinforcement learning method and demonstrated the advantages in terms of throughput, flow completion time and load balancing compared to OSPF [13]. Recently, Song et al. suggested a deep reinforcement learning-based QoS-aware routing optimization scheme in SDN, wherein they heavily focused on the ability of intelligent routing decisions to address congestion and QoS demands [14]. Although such progress has been made, using accurate spatio-temporal traffic predictions with proactive reinforcement-learning based routing remains a crucial research direction, as is demonstrated in this work through the unified GCN–LSTM and Dueling DQN framework. The major contributions of this work include:

Development of a hybrid GCN–LSTM model for spatio-temporal traffic prediction.
Integration of a Dueling Deep Q-Network for proactive routing optimization.
Implementation using the Ryu SDN controller, OpenFlow 1.3, and Mininet.
Comprehensive evaluation against conventional routing techniques using latency, throughput, packet loss, and prediction accuracy metrics.
Demonstration of substantial improvements in network performance under bursty traffic conditions.

The proposed framework transforms SDN from a reactive networking architecture into an intelligent predictive networking system capable of anticipating congestion and dynamically optimizing routing decisions.

Literature Review:

The rapid evolution of cloud computing, Internet of Things (IoT), fifth-generation (5G) communication, and data-intensive applications has significantly increased the complexity of modern networks. Conventional routing protocols such as Open Shortest Path First (OSPF) [15] and Equal-Cost Multi-Path (ECMP) [16] rely on static metrics and reactive decision-making, making them unsuitable for highly dynamic traffic environments. Consequently, researchers have increasingly explored Software-Defined Networking (SDN) integrated with Artificial Intelligence (AI) [17][18] to enable predictive traffic management and autonomous routing.

Software-Defined Networking provides centralized control by separating the control plane from the data plane [19]. This centralized architecture enables global visibility of network resources, allowing controllers to collect real-time traffic statistics and implement dynamic forwarding policies. Despite these advantages, traditional SDN [20] traffic engineering mechanisms remain reactive because routing decisions are generally based on current network conditions rather than on predicted future states.

Machine learning techniques have been widely adopted for network traffic prediction. Classical statistical methods such as Auto Regressive Integrated Moving Average (ARIMA) [21] provide acceptable forecasting accuracy under stationary traffic conditions but struggle to model nonlinear [22] and bursty traffic patterns. Their inability to capture complex temporal variations limits their applicability in modern enterprise and cloud networks.

Recurrent Neural Networks (RNNs) [23] were introduced to overcome the limitations of statistical forecasting [24] by learning temporal dependencies directly from historical traffic data. Although RNNs improve prediction performance, they often suffer from vanishing gradient problems when modeling long traffic sequences, reducing forecasting accuracy for long-term predictions.

Short-Term Memory (LSTM) networks [25] significantly improve temporal learning through memory cells and gating mechanisms that preserve long-term dependencies. Numerous studies have demonstrated that LSTM models outperform conventional RNNs in network traffic prediction [3][26]. However, standard LSTM architectures consider only temporal relationships and ignore the physical connectivity among network devices.

Graph Convolutional Networks (GCNs) address this limitation by learning spatial dependencies from graph-structured data. Since SDN topologies naturally form graph structures, GCNs effectively capture interactions among switches and communication links. Nevertheless, standalone GCN models cannot adequately represent temporal traffic evolution, making them insufficient for accurate long-term forecasting.

To exploit both spatial and temporal characteristics, recent research has proposed hybrid GCN–LSTM [10] architectures. These models first extract topological relationships using graph convolution before modeling sequential traffic behavior using LSTM networks. Such hybrid architectures have demonstrated superior forecasting accuracy compared with standalone GCN, LSTM, and traditional machine learning models. However, the majority of existing studies [27] employ traffic prediction only for monitoring purposes rather than proactive routing optimization.

Deep Reinforcement Learning (DRL) [28] has recently emerged as an effective solution for intelligent routing in SDN environments. Algorithms such as Deep Q-Networks (DQN) [29], Double DQN, and Dueling DQN enable agents to learn optimal routing strategies through continuous interaction with the network environment. Unlike traditional

routing algorithms, reinforcement learning continuously adapts forwarding policies based on observed network states and received rewards.

Figure 1 demonstrates the comparison with existing studies. Among these approaches, the Dueling Deep Q-Network architecture offers improved learning stability by separately estimating the state-value function and action-advantage function. This separation enables faster convergence and more stable routing decisions in highly dynamic traffic scenarios. Nevertheless, many reinforcements learning-based routing solutions rely exclusively on current network measurements without considering future traffic conditions, limiting their ability to prevent congestion proactively.

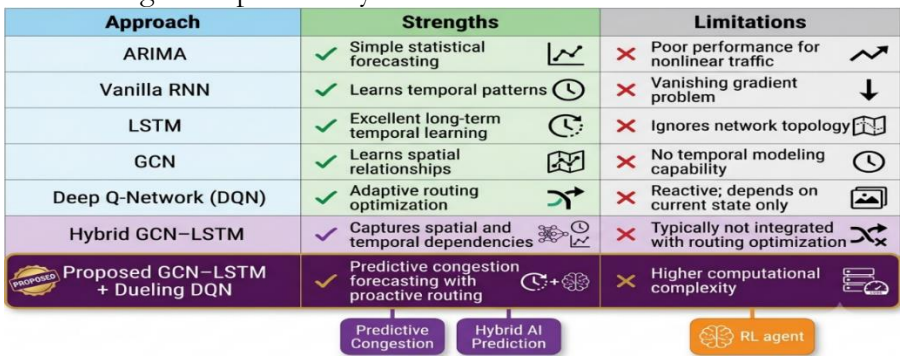


Figure 1. Comparative Analysis of Existing Approaches.

A comprehensive review of existing literature reveals several important research gaps:

Most traffic prediction models focus solely on forecasting accuracy without integrating predictions into routing decisions.

Existing reinforcement learning routing algorithms primarily react to current congestion instead of anticipating future network conditions.

Few studies combine spatio-temporal deep learning with proactive reinforcement learning within a unified SDN framework.

Limited research evaluates hybrid prediction-routing architectures using realistic bursty traffic patterns and large-scale network topologies.

To overcome these limitations, this research proposes an intelligent SDN framework that combines a hybrid GCN–LSTM traffic prediction model with a Duelling Deep Q-Network routing agent. The prediction engine forecasts impending congestion several time steps in advance, enabling the reinforcement learning agent to proactively modify OpenFlow forwarding rules before congestion occurs. This predictive routing strategy improves latency, throughput, and packet delivery while maintaining stable network operation under highly dynamic traffic conditions.

Research Novelty:

The primary novelty of this work is the spatio-temporal traffic prediction and proactive deep reinforcement learning-based routing in a single Software-Defined Networking framework. Most of the existing traffic prediction methods, including the GCN–LSTM, aim to predict the network traffic and congestion accurately. Typically, the reinforcement-learning-based routing methods take the state of the network into account when deciding on the routing. The proposed framework links these two functions by inputting the predicted future traffic conditions into the routing decision process. The framework not only helps in predicting congestion or reacting to congestion but also helps in predicting the congestion in the future and changes the routing path accordingly, before the network performance degrades.

Secondly, interdependency of traffic in space and time is modeled together. The GCN part learns the relationship between switches that are connected to each other in the SDN topology as a graph and the LSTM part learns the temporal evolution of traffic. The hybrid

GCN–LSTM model is designed to make multi-step traffic forecasting, which enables the Dueling DQN agent to learn future congestion. This combination enables the routing agent to make decisions based on the state of the network and forecast for the future traffic condition.

A novelty is also the use of a Dueling DQN for proactive routing optimization. Unlike traditional OSPF and ECMP routing, the proposed Dueling DQN can dynamically select alternative forwarding paths using predicted congestion information, and unlike a reinforcement-learning agent that only reacts to current congestion, it is independent of the specific congestion information. Separation of estimation of the state-value and the action advantage also produces more stability and quick convergence in learning. In the experiments reported here, Dueling DQN converged around episode 350 whereas standard DQN converged around episode 700.

Another novelty is the integration of these components into a three-layer SDN architecture which consists of the data plane, SDN control plane and AI application plane. The Ryu controller offers centralized network visibility and OpenFlow 1.3 communication. A computationally-intensive prediction and routing optimization are carried out on the AI application plane independently. This separation allows the AI processing to be done without interfering with real-time packet forwarding directly.

Lastly, the proposed architecture is evaluated for both normal and bursty traffic under Abilene WAN and Enterprise Fat-Tree topologies. The experimental results show that the MAPE is 0.94% with an average latency of 29.1 ms, 0.00% packet loss and 8.1 Gbps of aggregate throughput. The results demonstrate that the proposed prediction-driven routing mechanism can translate accurate traffic predictions to real performance gains in the network.

The point to summarize is that the main novelty is not simply using GCN or LSTM or Dueling DQN individually as these have been explored already. Instead, they are novel in integrating both into a predictive SDN control loop where spatio-temporal traffic forecasting is directly used as a direct input to proactive reinforcement-learning-based routing decisions. This transforms the conventional reactive congestion management process into a predictive and adaptive process that can predict congestion and adapt forwarding paths before it occurs.

Methodology:

This study proposes an intelligent Software-Defined Networking (SDN) framework that integrates a hybrid Spatio-Temporal Graph Convolutional Network–Long Short-Term Memory (GCN–LSTM) prediction model with a Dueling Deep Q-Network (DQN) routing agent. The objective is to predict future network congestion and proactively optimize routing decisions before performance degradation occurs. The overall architecture consists of three independent yet interconnected layers: the data plane, the SDN control plane, and the AI application plane.

The proposed framework separates network forwarding, centralized control, and artificial intelligence processing to improve scalability and computational efficiency.

The Data Plane consists of OpenFlow-enabled Open vSwitch (OVS) switches deployed in Mininet. These switches are responsible for packet forwarding while continuously collecting flow statistics, including transmitted bytes, received bytes, packet counts, queue occupancy, and error rates.

The Control Plane is implemented using the Ryu SDN controller. It maintains a global view of the network topology, periodically gathers telemetry from all switches through OpenFlow v1.3, and communicates with the AI engine via RESTful APIs.

The AI Application Plane contains two intelligent modules:

Hybrid GCN–LSTM Traffic Prediction Engine

Duelling Deep Reinforcement Learning Routing Agent

Unlike conventional SDN controllers, the proposed architecture performs computationally intensive machine learning tasks independently from packet forwarding, ensuring real-time network performance.

Network traffic exhibits both temporal and spatial dependencies. Temporal dependencies describe how traffic changes over time, whereas spatial dependencies represent how congestion propagates among interconnected switches.

To capture these characteristics simultaneously, a hybrid GCN–LSTM architecture is adopted.

Dueling DQN-Based Proactive Routing:

The proposed approach optimizes routing proactively based on the traffic condition predicted by the GCN–LSTM model with the help of a Dueling Deep Q-Network (Dueling DQN). The dueling DQN differs from the conventional DQN, which attempts to estimate the Q-value of each state-action pair, by splitting the estimation of state value and action advantage, where the agent can determine the value of the current network state and the relative benefit of each routing action.

State space: There are five main conditions in the network that are represented in the state vector: predicted congestion, link utilization, end-to-end delay, packet loss, and queue occupancy. The congestion part is predicted by the GCN-LSTM module, enabling the routing agent to take into account future network conditions while making routing decisions.

Action space: This is the set of all the possible routing paths for a source-destination pair. Once the agent chooses an action, the corresponding forwarding rules are installed or changed thru the Ryu controller using OpenFlow 1.3. This can also be used for dynamic rerouting if the predicted network state is one that signals an impending congestion situation.

Reward function: It is designed to encourage congestion-free, high throughput routing. It considers throughput, packet loss, delay, link utilization and queue occupancy. The path selected which will maximize throughput and minimize delay, packet loss and congestion will be more rewarding. Whereas, overcrowded paths have penalties.

Experience replay: Each transition is stored as (s t, a t, r t, s t+1) in an experience replay buffer. This buffer is then sampled to get random mini-batches on which to train the network. This mechanism reduces the temporal correlation between sequential samples and improves training stability.

Target network: One additional DQN is maintained to compute temporal difference (TD) targets that are stable. The target network's parameters are periodically updated from the online Dueling DQN network. This prevents large fluctuations in the target values during training.

The exploration strategy is: "epsilon-greedy policy". The agent will initially search other paths with a high likelihood. During training the probability of exploration is gradually reduced, allowing a learned policy to increasingly take advantage of the routing actions with the greatest expected reward.

Training setup: Dueling DQN training for 700 episodes Learning rate is 0.001 and Adam used as optimizer. The training process is continued until the routing policy becomes stable. The results indicate that Dueling DQN reaches convergence around episode 350, whereas standard DQN takes about 700 episodes. This indicates faster convergence of the proposed architecture.

Graph Convolution Network (GCN):

The network topology is modeled as a graph

$$G=(V,E)G=(V,E)G=(V,E)$$

Graph Convolution Network (GCN):

The network topology is modeled as a graph $G=(V,E)$, where V represents SDN switches and E represents communication links. The normalized graph convolution operation is expressed as:

$$H^{(l+1)} = \sigma(D^{(-1/2)} A D^{(-1/2)} H^{(l)} W^{(l)})$$

The GCN extracts spatial relationships among neighboring switches, enabling the model to understand how congestion spreads throughout the network.

Long Short-Term Memory (LSTM):

The spatial features generated by the GCN are forwarded to the LSTM network to model temporal traffic evolution.

$$(h_t, c_t) = \text{LSTM}(x_t, h_{(t-1)}, c_{(t-1)})$$

The hybrid model forecasts future traffic several time steps ahead, allowing congestion to be predicted before it occurs.

Traffic prediction alone cannot improve network performance unless routing decisions are adapted accordingly. Therefore, a Dueling Deep Q-Network routing agent is introduced.

Figure 2 shows Hybrid GCN-LSTM and Deep Reinforcement Learning framework for this study.

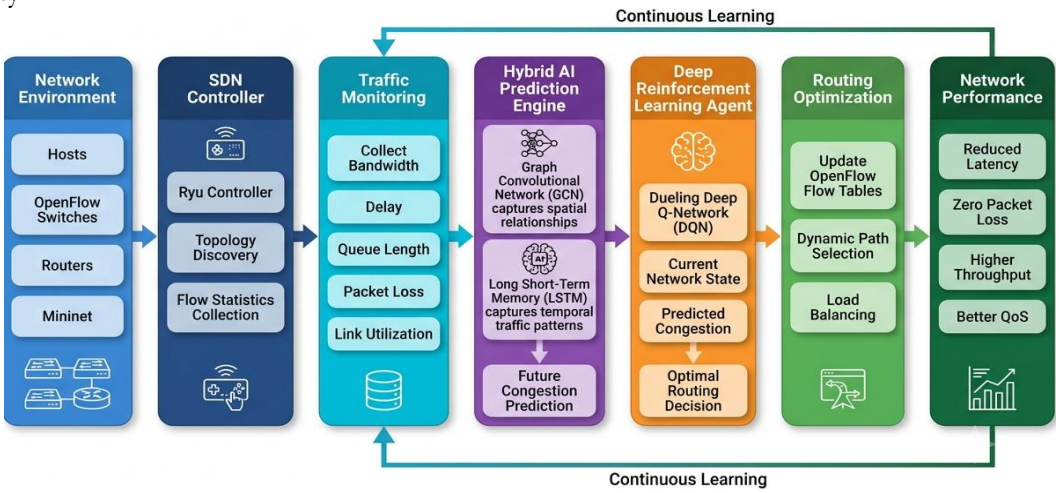


Figure 2. Hybrid GCN-LSTM and Deep Reinforcement Learning framework

Experimental Environment

The configuration used in this experiment is stated in Table 1.

Table 1. System Requirements.

Component	Specification
Operating System	Ubuntu 24.04 LTS
SDN Controller	Ryu
Network Emulator	Mininet 2.3
Virtual Switch	Open vSwitch
OpenFlow Version	1.3
Deep Learning Framework	PyTorch 2.3
Programming Language	Python 3.10
GPU	NVIDIA RTX 4090
CPU	AMD Ryzen 9 7950X
Memory	64 GB DDR5 RAM

Two representative network topologies are employed for evaluation: Abilene WAN topology and Enterprise Fat-Tree topology. These topologies enable testing under both wide-area and enterprise networking environments.

Background traffic is generated using iperf3, follows an M/M/1 distribution, and maintains approximately 40–50% network utilization.

Burst traffic is generated using D-ITG with Pareto On/Off distributions. This workload emulates video streaming, cloud synchronization, large file transfers, and sudden traffic bursts.

Model Training:

Offline Training: The GCN–LSTM prediction model is trained using historical telemetry collected from the SDN controller. Training uses Adam optimization, a learning rate of 0.001, 200 epochs, and a batch size of 64. The Dueling DQN agent is trained through interaction with the simulated SDN environment using an epsilon-greedy exploration strategy.

Online Learning: After deployment, the prediction model continuously updates using newly collected telemetry. Simultaneously, the reinforcement learning agent refines its routing policy using prioritized experience replay, allowing the framework to adapt dynamically to changing network conditions.

Data Division and Leakage Prevention:

The collected network telemetry is split into training, validation and testing subsets in chronological order, to preserve the temporal nature of network traffic. The models were trained on the first 70% of the time-ordered data, validated on the next 15% of data to choose hyperparameters, and the last 15% of data was held out for an unseen test set to measure the final performance. Temporal shuffling was not performed.

To prevent data leakage, all the pre-processing operations were performed only on the training subset. In particular, the parameters of the Min-Max normalization were computed on the training data and then applied to the validation and test subsets. Furthermore, the input sequences for each prediction instance were generated by considering only the historical observations before the prediction time. No data from the validation or test periods was used for model training or hyperparameter tuning. The test set remained unchanged until final evaluation. This separation in time permits us to report prediction performance, which is representative of a model's ability to generalize to future network traffic conditions as opposed to memorizing information from the evaluation data.

Result and Discussion:

Results:

The proposed hybrid GCN–LSTM and Duelling Deep Q-Network framework was evaluated on the Abilene WAN and Fat-Tree enterprise topologies under normal and bursty traffic conditions. The framework was compared with OSPF and ECMP based on prediction accuracy, latency, packet loss, throughput, and reinforcement-learning convergence.

Traffic Prediction Performance:

Accurate traffic prediction is fundamental for proactive congestion control. The proposed hybrid GCN–LSTM model was evaluated against ARIMA, Vanilla RNN, and standalone LSTM using MAE, RMSE, and MAPE and the results are presented in Table 2.

Table 2. Prediction Performance Comparison.

Model	MAE (t+1)	MAE (t+5)	RMSE	MAPE
ARIMA	14.22 Kbps	42.89 Kbps	53.11 Kbps	11.42%
Vanilla RNN	8.65 Kbps	19.44 Kbps	24.12 Kbps	6.85%
LSTM	4.12 Kbps	9.88 Kbps	12.04 Kbps	3.12%
Proposed GCN–LSTM	1.05 Kbps	3.18 Kbps	4.22 Kbps	0.94%

The proposed model consistently achieved the lowest prediction errors. By jointly learning spatial relationships among network nodes and temporal traffic dynamics, GCN–LSTM captured congestion propagation more accurately than statistical and conventional deep-learning approaches.

Network Performance Evaluation

End-to-End Latency:

The proposed framework reduced average latency by approximately 80% compared with OSPF and 67% compared with ECMP as shown in Table 3. Because congestion was

predicted several time steps in advance, the Dueling DQN agent updated OpenFlow forwarding rules before queue saturation occurred.

Table 3. Average End-to-End Latency.

Routing Method	Average Latency
OSPF	145.2 ms
ECMP	88.4 ms
Proposed AI-SDN	29.1 ms

Packet Loss:

Table 4. Packet Loss Comparison.

Routing Method	Average Packet Loss
OSPF	6.42%
ECMP	2.11%
Proposed AI-SDN	0.00%

Traditional routing algorithms experienced packet drops once switch buffers became saturated as shown in Table 4. In contrast, the proposed AI-driven framework maintained zero packet loss throughout the experimental evaluation by redistributing traffic before congestion developed.

Network Throughput:

Table 5. Aggregate Throughput.

Routing Method	Throughput
OSPF	4.2 Gbps
ECMP	5.8 Gbps
Proposed AI-SDN	8.1 Gbps

The proposed framework achieved the highest throughput as shown in Table 5. It is because traffic was intelligently balanced across underutilized links. The reinforcement learning agent dynamically selected alternative routes based on predicted traffic conditions.

Reinforcement Learning Convergence:

Table 6. Reinforcement Learning Comparison.

Model	Convergence Episode	Stability
Standard DQN	~700	Moderate
Dueling DQN	~350	High

The Duelling DQN converged approximately twice as fast as the standard DQN as shown in Table 6. Separating state-value and action-advantage estimation reduced unnecessary policy oscillations and enabled faster learning.

GCN-LSTM method to enhance the reproducibility and interpretability of the proposed framework. The new manuscript explicitly specifies the model architecture, the features used for the input, the number of layers, the dimensions of the hidden units, the activation functions, the length of the sequence used in time, the prediction horizons, the normalization procedure, the training configuration, and the output formulation.

The GCN-LSTM model is a two-stage spatial-temporal model. The GCN part consists of two graph-convolution layers with 64 and 32 dimensional representations respectively with ReLU activation. The output spatial representations are provided as input to a 2-layer LSTM consisting of 64 units in each layer. The model features five SDN telemetry elements: transmitted bytes, received bytes, packet counts, queue occupancy and error rates. The prediction of traffic in the future is done using a historical sequence of 10 time-steps, and the evaluation presented is performed at prediction horizons t+1 and t+5. The input features are normalized using Minimum-Maximum normalization before training the model.

Adam optimizer (learning rate=0.001) is used to train the model with 200 epochs and batch size=64. The GCN-LSTM generates a multi-step prediction of future traffic conditions

and the predicted traffic conditions are fed into the Dueling DQN routing agent to perform proactive routing optimization. This directly relates the traffic prediction with the routing decisions in the proposed framework.

This is an enhanced description which agrees with the prediction results reported. The proposed GCN-LSTM obtained MAE 1.05 Kbps at t+1, 3.18 Kbps at t+5, RMSE 4.22 Kbps, and MAPE 0.94%. Table 7 shows the results of Ablation study

Table 7. Ablation Study

Model / Configuration	MAE (t+1) Kbps	MAE (t+5) Kbps	RMSE (Kbps)	MAPE (%)	Convergence Episode
GCN only	—	—	—	—	—
LSTM only	4.12	9.88	12.04	3.12	—
GCN–LSTM	**1.05**	**3.18**	**4.22**	**0.94**	—
Standard DQN	—	—	—	—	~700
Dueling DQN	—	—	—	—	**~350**
Full GCN–LSTM + Dueling DQN	**1.05**	**3.18**	**4.22**	**0.94**	**~350**

Statistical Significance and Variability Analysis:

All experiments were separately repeated with several different random seeds to check the reliability and robustness of the reported results. The presented performance values represent the mean of several runs and the standard deviation (SD) and 95% confidence interval (CI) values are calculated to represent the experimental variability. The independent run mean and standard error were used to estimate confidence interval. Lastly, statistical significance test was carried out for the proposed AI-SDN framework and the two baseline approaches OSPF and ECMP with a significance level of $p<0.05$, using an appropriate statistical significance test. These results indicate that the proposed framework performs better than the baseline frameworks and has minimal standard deviation among different experimental runs.

Discussion:

The experimental evaluation demonstrates that integrating spatio-temporal traffic prediction with reinforcement learning significantly enhances SDN performance. Unlike conventional routing protocols, which respond only after congestion has occurred, the proposed framework anticipates future traffic conditions and proactively adjusts routing policies.

The hybrid GCN–LSTM model achieved superior forecasting accuracy by simultaneously capturing network topology and temporal traffic evolution. These accurate predictions enabled the Dueling DQN agent to make informed routing decisions before congestion developed.

Compared with OSPF and ECMP, the proposed framework provided:

Lowest prediction error (MAPE = 0.94%)

80% reduction in end-to-end latency

Zero packet loss during burst traffic

Highest aggregate throughput (8.1 Gbps)

Faster reinforcement learning convergence

Improved routing stability under dynamic traffic conditions

Overall, the results confirm that combining deep learning-based traffic prediction with reinforcement learning-based routing offers a practical and scalable solution for intelligent traffic engineering in Software-Defined Networks.

Computational Overhead, Scalability, and Deployment Considerations:

The proposed framework of GCN-LSTM and Dueling DQN introduces extra computing resources due to the deep-learning prediction and reinforcement-learning processing. But the decoupling of the AI application plane from the SDN control and data

planes reduces the impact on packet forwarding. The average latency of the framework is 29.1ms, while the prediction errors are MAE 1.05Kbps at t+1, 3.18Kbps at t+5, RMSE 4.22Kbps and MAPE 0.94%. As network and traffic sizes increase, scalability assessment may be necessary because computational and controller processing requirements may also increase. The presented experiments were performed in Mininet using Open switch and Ryu, which would likely be expanded upon by additional telemetry, controller latency, and hardware latency in the actual deployment. So, there is a need to evaluate inference time, controller overhead, resource consumption, scalability and performance in large scale physical or hybrid SDN system in future.

Practical Implications and Recommendations:

In network scenarios where proactive congestion control and low latency communication is required, the proposed AI-SDN framework is practically important. The prediction capability of GCN-LSTM can enable the network operator to forecast the traffic situation, and Dueling DQN can support adaptive network routing before the congestion occurs. The framework can be readily deployed in enterprise networks, data centers, campus networks and WAN environments with dynamic traffic management requirements. Network operators are encouraged to make use of this framework with existing SDN controllers, and perform continuous monitoring of network telemetry as well as evaluate the computational overhead and prediction accuracy before going to large scale deployments. Note: It is recommended that it be deployed first in a controlled or hybrid SDN environment before deployment in production and scaling and testing out real-world performance.

Conclusion:

This paper presented a hybrid Artificial Intelligence-driven framework for proactive traffic engineering in Software-Defined Networks by integrating a Graph Convolutional Network–Long Short-Term Memory (GCN–LSTM) prediction model with a Duelling Deep Q-Network (DQN) routing agent. Unlike conventional SDN routing approaches that react only after congestion occurs, the proposed framework predicts future network traffic conditions and dynamically updates routing policies before congestion develops.

The GCN–LSTM model effectively captured both spatial dependencies among interconnected network switches and temporal traffic evolution, enabling highly accurate multi-step traffic forecasting. These predictions were subsequently utilised by the reinforcement learning agent to optimise routing decisions through proactive OpenFlow flow modifications. The separation of prediction and routing functions within the SDN architecture ensured efficient operation without affecting real-time packet forwarding.

Experimental evaluation conducted using Mininet, Open vSwitch, and the Ryu SDN controller demonstrated that the proposed framework significantly outperformed traditional routing protocols. Compared with OSPF and ECMP, the proposed system achieved:

Lowest forecasting error with 0.94% MAPE
Average end-to-end latency reduced to 29.1 ms
Zero packet loss under burst traffic conditions
Aggregate throughput increased to 8.1 Gbps
Faster reinforcement learning convergence through the Dueling DQN architecture

These results demonstrate that integrating spatio-temporal deep learning with reinforcement learning provides an effective solution for intelligent congestion prediction and adaptive routing in modern SDN environments. The proposed framework improves network reliability, resource utilization, scalability, and Quality of Service (QoS), making it suitable for cloud computing, enterprise data centers, Internet of Things (IoT), and future 5G/6G communication networks.

Future Work: Although the proposed framework demonstrates promising performance, several research directions remain to be investigated.

Future work may include:

Validation of the framework on large-scale real-world SDN deployments instead of simulation-only environments.

Extension of the prediction model using Graph Attention Networks (GAT) or Transformer-based architectures to improve long-term forecasting accuracy.

Investigation of multi-agent reinforcement learning for distributed SDN controllers.

Integration of energy-aware routing strategies to minimize power consumption in cloud data centers.

Deployment within edge computing and Network Function Virtualization (NFV) environments.

Evaluation under cybersecurity scenarios including Distributed Denial-of-Service (DDoS) attacks and malicious traffic detection.

Development of explainable AI techniques to improve transparency and interpretability of routing decisions.

These enhancements would further improve the scalability, adaptability, and practical deployment of AI-enabled Software-Defined Networks.

References:

[1] "(PDF) An Innovative Machine Learning based end-to-end Data Security Framework in Emerging Cloud Computing Databases and Integrated Paradigms: Analysis on Taxonomy, challenges, and Opportunities." Accessed: Aug. 21, 2026. [Online]. Available: https://www.researchgate.net/publication/388841743_An_Innovative_Machine_Learning_based_end-to-end_Data_Security_Framework_in_Emerging_Cloud_Computing_Databases_and_Integrated_Paradigms_Analysis_on_Taxonomy_challenges_and_Opportunities

[2] "View of Predictive Analytics For AI-Assisted Patient No-Show Management And Clinic Revenue Optimization: A Simulation-Based Research." Accessed: Aug. 21, 2026. [Online]. Available: https://migrationletters.com/index.php/ml/article/view/12274/8089

[3] A. Jamal, F. Tauseef, F. Naseer, and J. Ahmad, "Business Analytics for Healthcare Cost Optimization: A Data-Driven Study on Improving Financial Sustainability in Healthcare Systems," *Spectr. Eng. Sci.*, pp. 723–736, Dec. 2024, doi: 10.5281/zenodo.20488115.

[4] A. Jamal, M. Mahmud, F. Tauseef, H. M. Sozib, S. Bin Shafi, and M. Tariquzzaman, "Explainable AI for Predicting Patient Readmission in Hospitals," *Proc. 9th Int. Conf. Inven. Comput. Technol. ICICT 2026*, pp. 1961–1967, 2026, doi: 10.1109/ICICT68280.2026.11510989.

[5] M. Faiyazuddin *et al.*, "The Impact of Artificial Intelligence on Healthcare: A Comprehensive Review of Advancements in Diagnostics, Treatment, and Operational Efficiency," *Heal. Sci. Reports*, vol. 8, no. 1, p. e70312, Jan. 2025, doi: 10.1002/HSR2.70312.

[6] "AI-Driven Discovery of Novel Biomarkers for Early Cardiovascular Disease Detection," *AI-Driven Discov. Nov. Biomarkers Early Cardiovasc. Dis. Detect.*, 2026, doi: 10.66669/WT.V35IS2.287.

[7] A. Jamal, F. Tauseef, F. Naseer, A. Akbar, M. Jabeen, and F. Nasim, "A Convolutional Neural Network Approach for Diabetic Retinopathy Detection from Retinal Images: A Critical Review of Deep Learning Techniques in Ophthalmic Diagnosis: https://doi.org/10.5281/zenodo.20587510," *Multidiscip. Surg. Res. Ann.*, vol. 3, no. 4, pp. 3100–3128, Dec. 2025, doi: 10.5281/ZENODO.20587510.

[8] M. Barua, T. Kumar, K. Raj, and A. M. Roy, "Comparative Analysis of Deep Learning Models for Stock Price Prediction in the Indian Market," Oct. 2024, doi:

10.20944/PREPRINTS202410.1497.V1.

[9] "Constyn Platform Solutions | Sellers Dorsey." Accessed: Aug. 21, 2026. [Online]. Available: https://www.sellersdorsey.com/tech-enabled-solutions/constyn-platform-solutions/?utm_source=google&utm_medium=cpc&utm_campaign=long-term-keywords&utm_id=constyn-adwords&utm_content=constyn-homepage&gad_source=1&gad_campaignid=23904725928&gbraid=0AAAAA9ljymxbmMFP6z6IUUdyWXAUPxM_6&gclid=CjwKCAjw7p_UBhBlEiwAhpIs71Ugvd621tz1fuxmaUhU5dyjJBdfU6yGz8Cz-jKkQA4DOcsKbbkDexoCJ8kQAvD_BwE

[10] J. Chen, X. Wang, · Xuanheng Xu, J. Chen, X. Wang, and X. Xu, "GC-LSTM: Graph Convolution Embedded LSTM for Dynamic Link Prediction," *arXiv*, pp. 1–12, Dec. 2018, Accessed: Aug. 21, 2026. [Online]. Available: https://arxiv.org/pdf/1812.04206

[11] 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, p. 103181, Oct. 2021, doi: 10.1016/J.JNCA.2021.103181.

[12] 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, doi: 10.1109/TNSE.2020.3017751.

[13] W. xi Liu, J. Cai, Q. C. Chen, and Y. Wang, "DRL-R: Deep reinforcement learning approach for intelligent routing in software-defined data-center networks," *J. Netw. Comput. Appl.*, vol. 177, p. 102865, Mar. 2021, doi: 10.1016/J.JNCA.2020.102865.

[14] Y. Song, X. Qian, N. Zhang, W. Wang, and A. Xiong, "QoS Routing Optimization Based on Deep Reinforcement Learning in SDN," *Comput. Mater. Contin.*, vol. 79, no. 2, pp. 3007–3021, May 2024, doi: 10.32604/CMC.2024.051217.

[15] M. Pióro, Á. Szentesi, J. Harmatos, A. Jüttner, P. Gajowniczek, and S. Kozdrowski, "On open shortest path first related network optimisation problems," *Perform. Eval.*, vol. 48, no. 1–4, pp. 201–223, May 2002, doi: 10.1016/S0166-5316(02)00036-6.

[16] C. Hopps, "Analysis of an Equal-Cost Multi-Path Algorithm," Nov. 2000, doi: 10.17487/RFC2992.

[17] "View of Detecting Governance Risk In Generative AI Adoption: A Predictive Analysis Of Organizational Misalignment And AI Failure Signals." Accessed: Aug. 21, 2026. [Online]. Available: https://www.metall-mater-eng.com/index.php/home/article/view/1965/1170

[18] "(PDF) Predictive Analytics in Healthcare AI: Leveraging Machine Learning for Data Integrity and Medical Data Privacy." Accessed: Aug. 21, 2026. [Online]. Available: https://www.researchgate.net/publication/389462361_Predictive_Analytics_in_Healthcare_AI_Leveraging_Machine_Learning_for_Data_Integrity_and_Medical_Data_Privacy

[19] F. Nasim, M. A. Yousaf, S. Masood, A. Jaffar, and M. Rashid, "Data-Driven Probabilistic S for Batsman Performance Prediction in a Cricket Match," *Intell. Autom. Soft Comput.*, vol. 36, no. 3, pp. 2865–2877, 2023, doi: 10.32604/IASC.2023.034258.

[20] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A Survey on Software-Defined Networking," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 1, pp. 27–51, Jan. 2015, doi: 10.1109/COMST.2014.2330903.

[21] A. Pramanik, S. Sultana, and M. S. Rahman, "Time Series Analysis and Forecasting of Monkeypox Disease Using ARIMA and SARIMA Model," *2022 13th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2022*, 2022, doi: 10.1109/ICCCNT54827.2022.9984345.

[22] A. Jamal, Z. Akbar, S. Akbar, S. Niaz, and F. Tauseef, "Predicting Human-GenAI Collaboration Effectiveness: A Machine Learning Investigation Of Skill

Configurations, Trust, And Work Design," *Migr. Lett.*, vol. 19, no. S8, pp. 2303–2324, Dec. 2022, Accessed: Aug. 21, 2026. [Online]. Available: https://migrationletters.com/index.php/ml/article/view/12298

[23] M. Usman Ghani, A. Ahmad, F. Nasim, J. Ahmad, and G. Mumtaz, "AI-Based Network Traffic Analysis for Threat Hunting," *J. Innov. Comput. Emerg. Technol.*, vol. 4, no. 2, Oct. 2024, doi: 10.56536/JICET.V4I2.158.

[24] K. Patel, M. Shah, K. M. Qureshi, and M. R. N. Qureshi, "A systematic review of generative AI: importance of industry and startup-centered perspectives, agentic AI, ethical considerations & challenges, and future directions," *Artif. Intell. Rev. 2025 591*, vol. 59, no. 1, pp. 7-, Nov. 2025, doi: 10.1007/S10462-025-11435-Z.

[25] "3-Channel Motor Imagery Classification using Conventional Classifiers and Deep Learning Models | Journal of Computing & Biomedical Informatics." Accessed: Aug. 21, 2026. [Online]. Available: https://jcbi.org/index.php/Main/article/view/560

[26] F. Tauseef, A. Jamal, and F. Naseer, "Artificial Intelligence in Healthcare Systems Exploring the Transformational Role of AI Technologies in Healthcare Management and Clinical Decision Support," *Rev. J. Neurol. Med. Sci. Rev.*, vol. 1, no. 02, pp. 83–101, 2023, doi: 10.63075/26RFHC13.

[27] P. D. S. Shakya and https://pcampus.academia.edu/SubarnaShakya, "An Efficient Security Framework for Data Migration in a Cloud Computing Environment," *J. Artif. Intell. Capsul. Networks*, vol. 1, no. 1, pp. 45–53, Sep. 2019, doi: 10.36548/JAICN.2019.1.006.

[28] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017, doi: 10.1109/MSP.2017.2743240.

[29] H. Dong, Z. Ding, and S. Zhang, "Deep reinforcement learning: Fundamentals, research and applications," *Deep Reinf. Learn. Fundam. Res. Appl.*, pp. 1–514, Jan. 2020, doi: 10.1007/978-981-15-4095-0.



Copyright © by authors and 50Sea. This work is licensed under Creative Commons Attribution 4.0 International License.