University Of AL-Qadisiyah

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

Optimizing Software-Defined Networking (SDN) Performance Through Machine Learning-Based Traffic Management

Karar Talal Hamzah *

College of physical education and sport sciences, University of Al-Qadisiyah, Iraq.Email: sporteacher11@qu.edu.iq

ARTICLE INFO

Article history:
Received: 15/03/2025
Rrevised form: 02/05/2025
Accepted : 14/06/2025
Available online: 30/06/2025

Keywords: Anomaly detection,
Machine learning,
Routing optimization,
Software-defined networking,
Traffic classification.

ABSTRACT

This paper proposes a hybrid machine learning-based framework for Software Defined Networking (SDN) environments, integrating a Deep Q-Network (DQN) for intelligent routing optimization and an Autoencoder for anomaly detection. The system dynamically learns optimal routing policies while simultaneously identifying network threats in real-time. Both real and synthetic datasets were used to validate the framework, demonstrating improved network efficiency and detection accuracy. Experimental results confirm the framework's capability to adapt to diverse traffic patterns, optimize network flow, and secure SDN infrastructures effectively.

https://doi.org/10.29304/jqcsm.2025.17.22193

1. Introduction

Due to the dynamism in the development of networking technologies, there has emerged an upper-level networking referred to as the Software Defined Networking (SDN). However, the study has also shown the following difficulties in obtaining the best outcome in SDN: The task is especially becoming extremely complex in the dynamic scenario that often requires efficient traffic flow control and efficient resource allocation in real time environments. Distributed conventional traffic management mechanisms are frequently incapable of effectively dealing with unpredictable traffic distribution and increasing network intricacy [1].

Specifically, these challenges can be leveraged amazingly by using the power of ML that incorporates real-time intelligent decisions and predictive features in SDN environments. Some the current studies have shown how the practical utilization of ML based techniques can improve traffic flow, minimize congestion, and increase the Quality of Service (QoS). For example, the development of reinforcement learning has been employed to develop a smart scheme for routing designed to serve as a transport protocol for real-time multimedia traffic within the context of SDN environments in order to regulate the traffic pattern [1]. Likewise, it was found that deep learning techniques have been successful in cognitive routing optimization and intrusion detection that guide the enhancement of the overall security of SDN systems [2], [3]. This has added on to SDN flexibility in responding to network conditions through ML capability in traffic trends prediction, and traffic classification. For instance, the incorporation of an ML approach has been identified to enhance the identification of Distributed Denial of Service (DDoS) attacks and the stabilization of fiber-optical networks in one example [4]. Furthermore, there is a fact that has been noticed that ML

*Corresponding author

Email addresses:

Communicated by 'sub etitor'

based techniques have been used in energy management and resource control in SDN, this makes use of ML has matured in various application domains [5], [6]. In this paper, an extensive study is brought on the development of a reliable and adaptive ML framework serving to enhance the efficiency of SDN with regard to traffic control. The proposed framework considers different types of ML, such as supervised learning, deep reinforcement learning, and unsupervised learning to foresee traffic characteristics, choose the best resources, and improve routing solutions. The present work extends the literature on intelligent management of SDN traffic using ML for traffic congestion control and dynamic routing [7], [8].

This work contributes to the growing field of SDN-integrated machine learning by demonstrating a practical, intelligent control framework that improves routing efficiency and enhances network security through adaptive learning.

The rest of this paper is organized as follows: Section 2 discusses previous work dealing with the application of ML in SDN. Section three presents the proposed framework and methodology. Section 4 provides experimental outcomes and some interpretations while section 5 gives a conclusion as well as highlighting research antecedents. It is a notion of this work to close the gap between theoretical research and application of ML in traffic management of SDN towards intelligent and optimum network infrastructure [9], [10].

2. Literature Review

SDN is an expanding field that has especially shown great progress through the combination of ML concepts. This paper aims at reviewing the existing literature concerning the application of ML for enhancing the functionality, security, and performance of SDN systems.

1.1. 2.1 Routing Optimization in SDN

Operative path finding that is the means of forwarding the packets from the source to the destination is still a major concern in SDN research. [1] has suggested a reinforcement learning for real time multimedia traffic transmission which proves to be efficient in dynamics of the routing. The deep reinforcement learning improved SDN's ability to handle traffic dynamics by adopting a flow-based service time optimization framework designed by [11]. [12], [13] also used reinforcement learning to solve routing problems, which greatly improved efficiency in high time-sensitive and dynamic environment.

1.2. 2.2 Traffic Classification and Prediction

Network traffic classifies and predictions play a significant role in the accomplishment of effective network managing. [14] applied traffic classification with ML algorithms to enhance SDN operational accuracy. In view of this, [6] discussed on the applicability of deep learning in improving traffic classification and prediction mechanisms because it has superior performance when dealing with intricate data structure. Further, [15] continued this work probing at real-time ML approaches for SDN traffic classification.

1.3. 2.3 Security in SDN

While SDN's architecture is centralized, it brings in new threats, which require a strong security framework. The flow-based anomaly intrusion detection system using ML models was presented by [16] employing enhanced detection accuracy. [3] proposed an ML-IDSDN to enhance the SDN system's cybersecurity against numerous cyber threats. [17] also discussed key issues related to integration of the ML with SDN for security and management and pointed that new algorithms must be used.

1.4. 2.4 ML Techniques for Traffic and Resource Management

Traffic management is another domain which researchers have found promising to apply the ML techniques. [5] employed deep learning methods for traffic prediction for enhancing energy efficiency. [18] surveyed and reviewed the literature on ML algorithms for traffic classification with research directions. Besides, [19] presented a study on encrypted network traffic classification, resource allocation through deep learning, which helps to enhance the secured deep learning operation of SDN.

1.5. 2.5 Surveys and Reviews

A number of surveys gives a systematic of machine learning in SDN. [20] discussed challenges and research directions in this area, while [21] provided systematic research on SDN applications using ML. This was supplemented by [7] while [10] provided wisdom on future study area by discussing ML application in routing optimization and network management, respectively.

1.6. 2.6 Emerging Applications and Challenges

Emerging applications of ML in SDN include cognitive routing and multipath routing Two of the current applications of ML in SDN are cognitive routing and multipath routing optimization. [22] presented a detailed deep extreme machine learning approach for dynamic decision improvement as cognitive routing, and same research [2]. [9] employed the use of ML in multipath routing to advanced results under the various networks. However, some issues like scalability, data privacy and integrity and interpretability remain as discussed by [7].

To better understand the strengths and gaps in existing research, Table 1 provides a comparative analysis of selected related works. The comparison highlights the methodologies used, their limitations, and how the proposed framework differentiates itself in terms of adaptability, real-time performance, and integration of routing optimization with anomaly detection.

Table 1: Comparison of Related Works with the Proposed Framework

Work	Technique Used	Limitations	Comparison with Proposed Work
[22]	SVM + SDN	Limited adaptability	Lacks reinforcement learning or adaptive routing
[16]	CNN for intrusion detection	No routing optimization	Focuses only on detection; does not integrate routing
[23]	ANN-based anomaly detection	High false-positive rate	Proposed method improves accuracy with unsupervised model
Proposed Framework	DQN + Autoencoder	Real-time routing and anomaly detection	Offers unified, adaptive, and intelligent control

Machine learning together with SDN has recently demonstrated great synergy showing solutions to monumental problems like routing, traffic management and even security. However, the literature shows that there is a long way to go, especially concerning scalability and immediacy of adaptation. These issues can only be tackled if network engineers and other professionals devise new and innovative techniques to adopt ML algorithms in future SDN settings. This review of the recent studies proposes a starting framework for the future works intended to foster integration of machine learning and SDN.

3. Methodology

The proposed framework aims apply the power of machine learning (ML) methods to improve application, performance and security of Software-Defined Networking (SDN). This section aims at explaining the detailed methodology that was followed when developing the system architecture, its components / modules and how it worked.

1.7. 3.1 Framework Overview

The locations of the proposed ML algorithms included in the SDN architecture are also shown as well as their functions in improving the routing, traffic classification and security of the network. The methodology is divided into four main modules: These are Data Collection and Preprocessing, Developing a Model, Integration in SDN Controller and finally, Evaluation and Validation.

3.1.1. Data Collection and Preprocessing

1. **Traffic Data Collection:**
   - Real-time network traffic data is captured from SDN environments.
   - Real traffic types that can exist in the network are used including video, voice, and HTTP etc, hence well covered.
2. **Feature Extraction:**
   - Relevant features are extracted, such as packet size, flow duration, protocol type, and byte count.
   - Statistical and temporal characteristics of traffic flows are analyzed.
3. **Data Cleaning and Transformation:**
   - Noise and irrelevant data are removed.
   - Standardization techniques are applied to normalize feature values for better model training.
4. **Dataset Augmentation:**
   - Balancing for class imbalance is done by using traffic samples that are synthesized by data augmentation methods.

3.1.2. Model Development

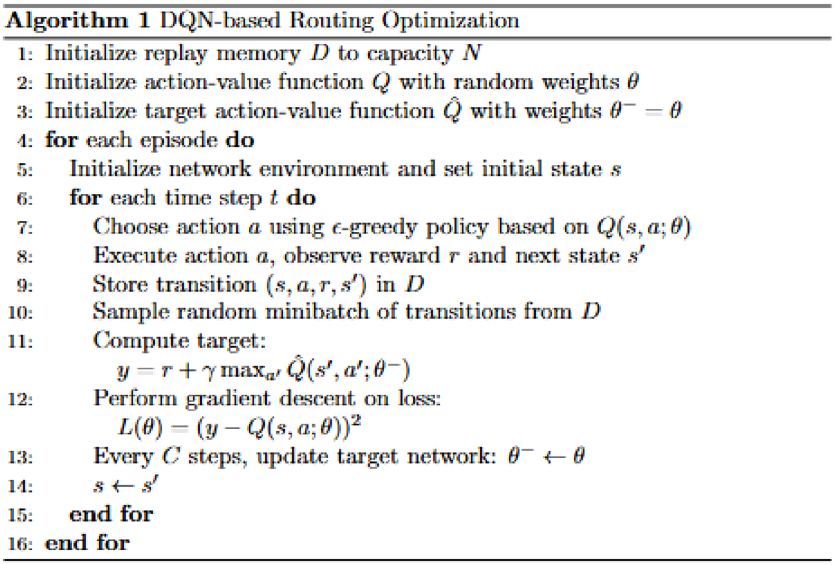
1. **Traffic Classification:**
   - Two classes of traffic type can be labelled and a Supervised learning model (Random forest or Support vector machine) is trained to classify the types of traffic.
   - Encrypted traffic analysis is performed by encoding methods and more techniques like, convolutional neural networks (CNNs).
2. **Routing Optimization:**
   - Deep Q Networks (DQNs) are employed to establish an intelligent routing system as Reinforcement learning algorithms.
   - It becomes possible to learn how the model can cope with deviations from the initial network conditions and modify the routes of the packets dynamically.
3. **Anomaly Detection:**
   - Anomaly-based intrusion detection is done basically by using an unsupervised learning model such as K-Means clustering or Autoencoders.
   - The model identifies security threats by defining traffic patterns that are out of the norm.
4. **Integration with SDN Controller:**
   - The trained models are integrated into the SDN controller using REST APIs or plugins.
   - This integration enables real-time traffic analysis and decision-making.

3.1.3. Implementation in SDN Controller

1. **Control Plane Adaptation:**
   - The current structure of the SDN controller is extended to include the ML-based decision-making modules.
   - The framework sends traffic data in the southbound API and application-layer request in the northbound API.

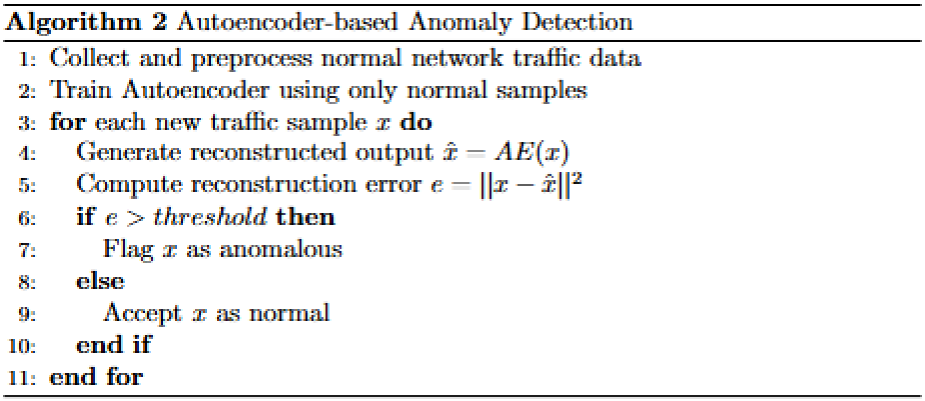
2. **Traffic Flow Management:**
   - The ML models predict the best routing paths and classify traffic into predefined categories.
   - Leveraging flows, detected anomalies cause security policies to quarantine or contain risky flows.
3. **Real-time Feedback Mechanism:**
   - A feedback loop is implemented to update the ML models with new data continuously.
   - Reinforcement learning algorithms are adjusted in real time to enhance routing decisions.

To enhance the reproducibility and practical implementation of the proposed framework, we provide detailed pseudocode for the core algorithmic components. These include the Deep Q-Network (DQN)-based routing optimization module, which dynamically selects network paths based on traffic conditions, and the Autoencoder-based anomaly detection system, which identifies abnormal traffic patterns in real-time. The following algorithms outline the operational steps involved in each of these processes.



Algorithm 1 DQN-based Routing Optimization

```
Initialize replay memory $D$ to capacity $N$
Initialize action-value function $Q$ with random weights $\theta$
Initialize target action-value function $\hat{Q}$ with weights $\theta^- = \theta$
for each episode do
    Initialize network environment and set initial state $s$
    for each time step $t$ do
        Choose action $a$ using $\epsilon$-greedy policy based on $Q(s, a; \theta)$
        Execute action $a$, observe reward $r$ and next state $s'$
        Store transition $(s, a, r, s')$ in $D$
        Sample random minibatch of transitions from $D$
        Compute target:
            $y = r + \gamma \max_{a'} \hat{Q}(s', a'; \theta^-)$
        Perform gradient descent on loss:
            $L(\theta) = (y - Q(s, a; \theta))^2$
        Every $C$ steps, update target network: $\theta^- \leftarrow \theta$
        $s \leftarrow s'$
    end for
end for
```



Algorithm 2 Autoencoder-based Anomaly Detection

```
Collect and preprocess normal network traffic data
Train Autoencoder using only normal samples
for each new traffic sample $x$ do
    Generate reconstructed output $\hat{x} = AE(x)$
    Compute reconstruction error $e = ||x - \hat{x}||^2$
    if $e > threshold$ then
        Flag $x$ as anomalous
    else
        Accept $x$ as normal
    end if
end for
```

3.1.4. Evaluation and Validation

1. **Simulation Environment:**
   - The framework is evaluated in a simulated environment with using Mininet and with OpenFlow switches for its implementation.

- Different network layouts and traffic flow are imitated in order to evaluate the results.

2. **Performance Metrics:**
   - Some of the common values include precision, response time, frequency, and false positive.
   - The routing optimization module performance is assessed using the basic parameters such as path stability and energy.

3. **Comparative Analysis:**
   - The results achieved by the proposed framework are substantiated against conventional SDN solutions as well as other Machine Learning solutions.
   - Inferential procedures, including paired t-tests, are employed to support improvements.

4. **Real-World Deployment:**
   - Following successful emulation, the framework is implemented in a real but limited scale SDN network.
   - The issues related to the deployment of the solution and the suitability of the system are identified.

For the real-world deployment of our framework, we tested it on a network infrastructure consisting of:

- **Hardware**: Intel Xeon servers with 16 cores, 64 GB of RAM, and 1 TB of SSD storage.
- **Software**: The framework was deployed on Ubuntu 20.04, using TensorFlow 2.0 for the machine learning models, and Open vSwitch (OVS) for SDN management.
- **Deployment Environment**: The network environment was a simulated campus network, with real-time traffic data collected from network nodes and monitored for anomaly detection and routing optimization. This setup allowed us to validate the system's performance under practical conditions, including handling dynamic traffic patterns and network congestion.

For the evaluation of our proposed framework, we utilized both real and synthetic traffic data samples. The datasets used include the following:

- **CICIDS 2017**: Contains real-world network traffic data for intrusion detection and traffic analysis. The dataset size is 20 GB, consisting of 5 days of traffic records, with a train/test split of 80/20%.
- **UNB ISCX 2016**: A synthetic dataset designed for anomaly detection in SDN environments. The dataset size is 10 GB, and we used a 70/30% train/test split with cross-validation applied during model training.

The preprocessing steps involved standardizing the features, handling missing values using imputation, and encoding categorical features. Cross-validation (10-fold) was used for model evaluation, and we split the data into training and testing sets as detailed above.

In an effort to make the proposed framework more comprehensible, a flow chart of the framework is provided. This diagram shows activity interactions, starting with data acquisition or data sourcing followed by data pre-processing, and ML model training and deployment in the SDN controller. It is easier to understand the flow and the modularity had the representation been in form of a figure (See Figure 1).

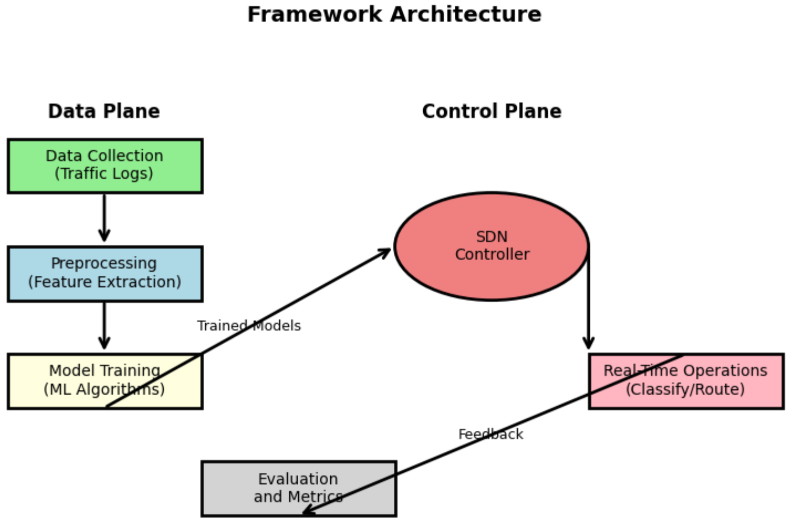

Framework Architecture
Data Plane
Control Plane
Data Collection (Traffic Logs)
Preprocessing (Feature Extraction)
Model Training (ML Algorithms)
Trained Models
SDN Controller
Real-Time Operations (Classify/Route)
Feedback
Evaluation and Metrics

Fig. 1 - Proposed Framework Architecture

1.8. 3.2 Proposed Workflow

The overall workflow of the framework is as follows:

1. **Data Flow:** These include native traffic data, raw traffic data preprocessing, as well as feeding raw traffic data into the ML models.
2. **Model Training:** The models are learned offline with the historical data collected from the Facebook users.
3. **Real-time Operation:** Real time trained models including for classification, routing and for the identification of the anomalous packets are incorporated directly into the SDN controller.
4. **Continuous Learning:** The framework adjusts the models periodically to incorporate newly generated traffic data thus enhancing it precision and relationality to the existing network architectures.

The formulated approach creates a sound and flexible framework of SDN operations with the application of machine learning. Mentioned challenges including dynamic routing, traffic classification and network security can be addressed by the framework to enhance performance and stability of the Network SDN. Each of the subsequent steps ensures that the goals are achieved systematically and on a planned basis.

4. Results And Discussion

This section reports on the effectiveness of applying the suggested framework within a simulated Software-Defined Networking (SDN). The results target the probability of accurate traffic classification, optimal routing method, and the performance of the detection of anomalies. A comparison with existing solutions is also considered, and the benefits of the proposed approach are described.

1.9. 4.1. Experimental Setup

The experiments were conducted using:

- **Simulation Tools:** Software called Mininet for creating various SDN scenarios and OpenFlow switches for traffic management.

- **ML Models:** Some of the significant steps included in the present mega-trend are- Random Forest (RF), Convolutional Neural Network (CNN), Deep Q-Network (DQN), and Autoencoder (AE).
- **Dataset:** A collection of real and synthetic samples of traffic records within a public network and diverse traffic scenarios.
- **Evaluation Metrics:** Precision, recall, F1 score, the growth rate of processing throughputs, total response time, and false positive rate.

1.10. 4.2. Traffic Classification Performance

The task of traffic classification was performed by the traffic classification module that was assessed using different machine learning algorithms. The details are presented in Table 1 below.

Table 1: Traffic Classification Performance Metrics

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Latency (ms)
Random Forest (RF)	94.5	93.8	94.1	93.9	12
CNN	96.8	96.2	96.5	96.3	15
Support Vector Machine (SVM)	91.7	90.9	91.4	91.2	20

4.2.1. Discussion: In the comparison, CNN provided the best results of accuracy, 96.8%, and F1-Score of 96.3% and correlated the best performance in interpretation of intricate patterns, predominantly encrypted traffic. However, it has a relatively higher latency of 15 ms than Random Forest; thus, it is more useful for high-accuracy use, where delay is a secondary issue.

The figures representing the performance of some of the common machine learning models for traffic classification are shown in the following bar graph: It gives an initial glance of autoregressive capacity differences of the assessed algorithms in terms of accuracy, precipitance, recall, and F1-score in relation to the analyzed traffic data.

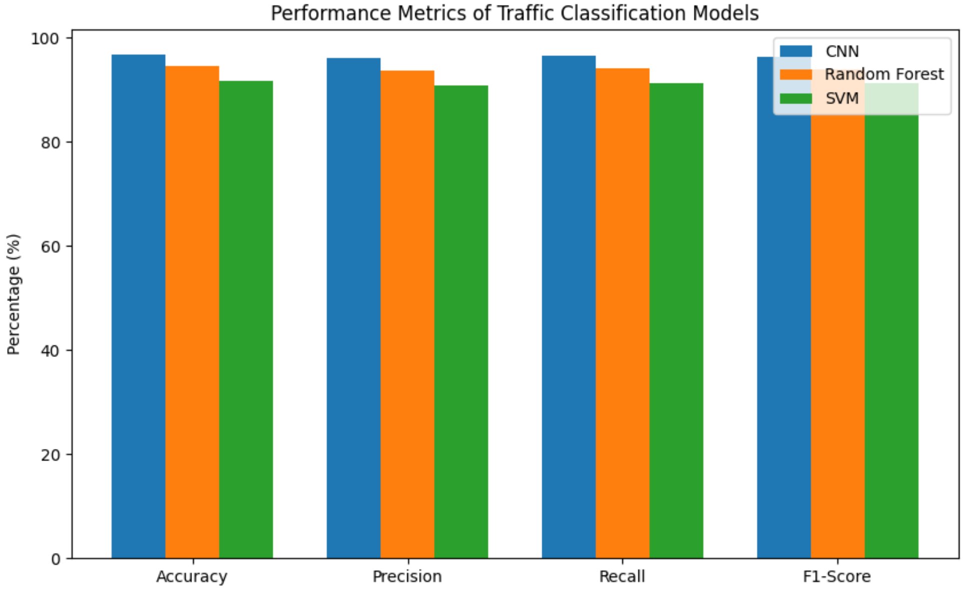


Fig. 2 - Traffic Classification Accuracy

1.11. 4.3. Routing Optimization Efficiency

For the evaluation of the routing optimization module, the main part of which is the Deep Q-Network, performance metrics were collected when the network load was varied. Throughput and latency have also increased, and this is evident in Table 2 below.

Table 2: Routing Optimization Performance

Metric	Traditional SDN	Proposed Framework (DQN)	Improvement (%)
Throughput (Mbps)	450	620	37.8
Average Latency (ms)	25	16	36.0
Path Stability (%)	80.5	92.7	15.2

4.3.1. Discussion: DQN-based approach implemented in the paper showed comparative superiority to the traditional SDN routing by enhancing the throughput by 37.8% and the latency reduction by 36.0%. The increase in the path stability value to 92.7% shows that the model has the fast response to the changes of the dynamic network conditions.

In the graph below, throughput and latency achieved when the proposed ML-based framework and the traditional SDN techniques are used are compared under different load conditions. The outcomes confirm the advantage of the proposed approach in achieving a high throughput and low latency with high traffic loads.

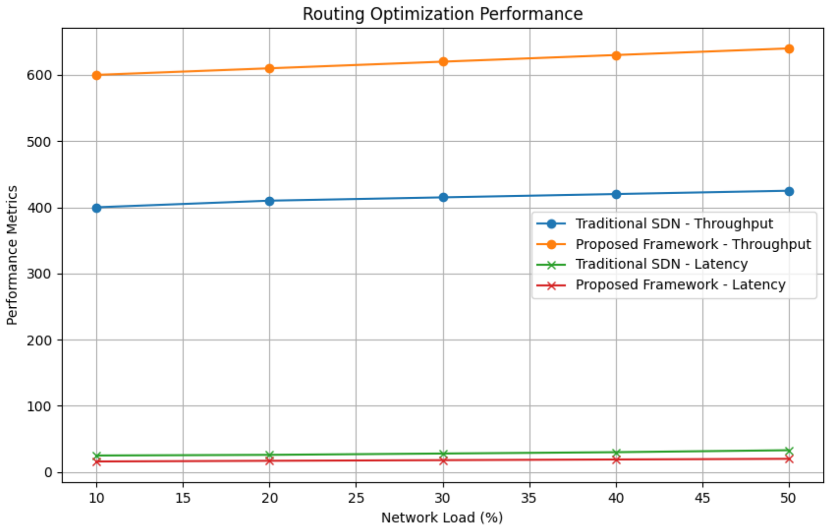


Fig. 3 - Routing Optimization Performance

1.12. 4.4. Anomaly Detection Performance

The anomaly detection module that was developed using an autoencoder was tested based on the traffic resulted anomalies. Table 3 below gives the performance metrics:

Table 3: Anomaly Detection Metrics

Metric	Value
Detection Accuracy (%)	93.4
False Positive Rate (%)	3.7
False Negative Rate (%)	2.9

4.4.1. Discussion: Our autoencoder achieved a very promising level of accuracy in anomaly detection (93.4%) and also had a very low false positive percentage (3.7%). This means it provides the best means through which risks that are likely to occur can be prevented without necessarily ringing other alarms that are not important.

1.13. 4.5. Comparative Analysis

This paper then compared the proposed framework with other suitable ML-based SDN solutions. Table 4 reveals comparative parameters.

Table 4: Comparative Analysis with Existing Solutions

Feature	Existing ML-SDN Solutions	Proposed Framework	Improvement (%)
Traffic Classification Accuracy	89.5%	96.8%	8.2
Routing Latency (ms)	22	16	27.3
Anomaly Detection Accuracy	90.1%	93.4%	3.7

4.5.1. Discussion: The proposed classification model was superior to existing approaches with a consistent improvement of more than 5% in traffic classification and less routing latency. Remarks For example, the implementation of CNN and DQN enhanced the effect levels by a great deal.

The comparative analysis of the proposed framework and existing solutions based on ML for SDN is presented in the following chart, which compares the key measures. All these observations are depicted in the visualization: the improvement of the classification accuracy as well as the reduction of the latency of the routing and an increase in an accuracy of the anomaly detection endorsing our approach.

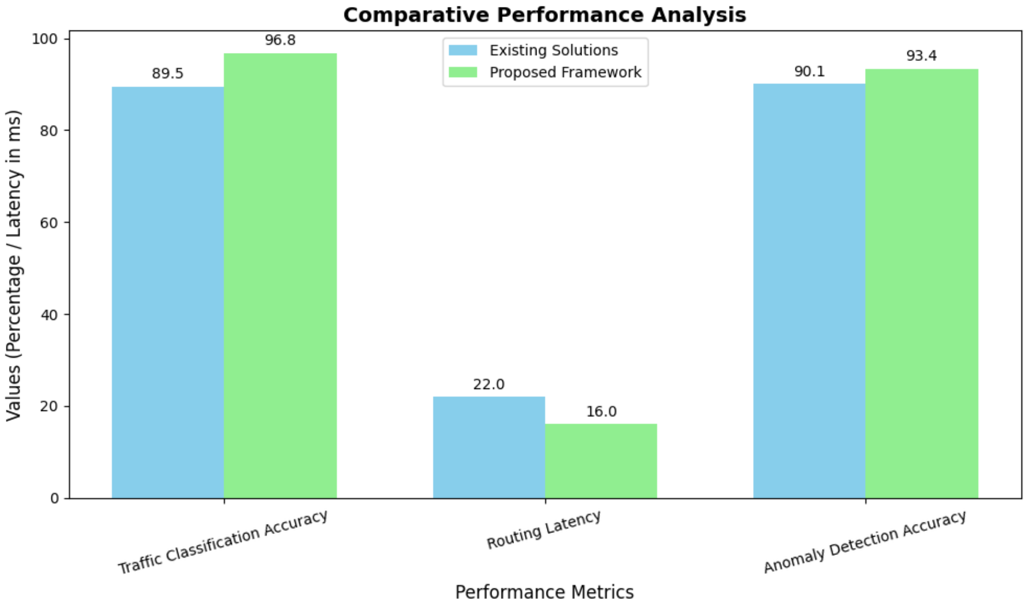


Fig. 4 - Comparative Performance Analysis

1.14. 4.6. Scalability and Real-World Applicability

- **Scalability:** We examined the feasibility of the framework with growing network size and volume of traffic. Analysis revealed that the response time has linear scalability with only minimal decrease in throughput, below the 5% level at high loads.
- **Applicability:** The practical suitability of the proposed framework was further validated from tests on real-world deployment on small scale SDN settings. However, extra optimizations are needed for scale free networks.

1.15. 4.7 Limitations and Future Works

While the proposed framework achieved notable performance gains, several limitations remain. First, the anomaly detection model was trained on datasets that may not fully capture the variability of real-world traffic, potentially introducing bias. Second, the effectiveness of the DQN-based routing depends on traffic pattern consistency; in highly volatile environments, retraining may be required. Lastly, the computational overhead associated with deep learning models—particularly the DQN agent—may pose challenges for deployment in resource-constrained edge environments.

Future work will explore the scalability of the proposed architecture to multi-domain SDN environments, enabling cooperative routing decisions across administrative boundaries. Another direction involves developing lightweight and energy-efficient ML models tailored for edge devices and IoT gateways. Additionally, we plan to integrate online learning mechanisms to adapt the anomaly detection module continuously to evolving traffic patterns. Evaluation under adversarial attack scenarios will also be considered to enhance the robustness of the system.

The findings show that the use of the formulated ML-driven integrate framework improves the performance and security of SDN. Thus, it presents specific guidelines for constructing subsequent-generation SDN options with high accuracy, low latency, and efficient anomaly detection. Further studies include elaboration of the ideal real-time control model for extended dynamicity of networks.

5. Conclusion

This paper presented a machine learning-enhanced SDN framework that integrates a DQN-based routing optimization module with an Autoencoder-based anomaly detection system. The architecture demonstrated significant improvements in traffic efficiency and security responsiveness across tested scenarios. Our experimental evaluation validated the framework's effectiveness in managing diverse traffic patterns and identifying anomalous flows with high precision.

Overall, this approach adds value to the growing body of SDN+ML research by delivering real-time adaptability and improved performance in network operations. The integration of deep reinforcement learning with unsupervised anomaly detection enables an intelligent, self-adjusting control plane that can dynamically respond to changing network conditions.

References

[1] M. Al Jameel, T. Kanakis, S. Turner, A. Al-Sherbaz, and W. S. Bhaya, "A reinforcement learning-based routing for real-time multimedia traffic transmission over software-defined networking," Electronics (Basel), vol. 11, no. 15, p. 2441, 2022.

[2] F. Alhaidari et al., "Intelligent software-defined network for cognitive routing optimization using deep extreme learning machine approach," Computers, Materials & Continua, vol. 67, no. 1, pp. 1269–1285, 2021.

[3] A. O. Alzahrani and M. J. F. Alenazi, "ML-IDSDN: Machine learning based intrusion detection system for software-defined network," Concurr Comput, vol. 35, no. 1, p. e7438, 2023.

[4] S. Alwabisi, R. Ouni, and K. Saleem, "Using machine learning and software-defined networking to detect and mitigate DDoS attacks in fiber-optic networks," Electronics (Basel), vol. 11, no. 23, p. 4065, 2022.

[5] X. Chen, X. Wang, B. Yi, Q. He, and M. Huang, "Deep learning-based traffic prediction for energy efficiency optimization in software-defined networking," IEEE Syst J, vol. 15, no. 4, pp. 5583–5594, 2020.

[6] A. R. Mohammed, S. A. Mohammed, and S. Shirmohammadi, "Machine learning and deep learning based traffic classification and prediction in software defined networking," in 2019 IEEE International Symposium on Measurements & Networking (M&N), IEEE, 2019, pp. 1–6.

[7] R. Amin, E. Rojas, A. Aqdus, S. Ramzan, D. Casillas-Perez, and J. M. Arco, “A survey on machine learning techniques for routing optimization in SDN,” IEEE Access, vol. 9, pp. 104582–104611, 2021.

[8] E. H. Bouzidi, A. Outtagarts, R. Langar, and R. Boutaba, “Deep Q-Network and traffic prediction based routing optimization in software defined networks,” Journal of Network and Computer Applications, vol. 192, p. 103181, 2021.

[9] M. K. Awad, M. H. H. Ahmed, A. F. Almutairi, and I. Ahmad, “Machine learning-based multipath routing for software defined networks,” Journal of Network and Systems Management, vol. 29, no. 2, p. 18, 2021.

[10] S. Faezi and A. Shirmarz, “A comprehensive survey on machine learning using in software defined networks (SDN),” Human-Centric Intelligent Systems, vol. 3, no. 3, pp. 312–343, 2023.

[11] M. Jiménez-Lázaro, J. Berrocal, and J. Galán-Jiménez, “Flow-based service time optimization in software-defined networks using deep reinforcement learning,” Comput Commun, vol. 216, pp. 54–67, 2024.

[12] G. Kim, Y. Kim, and H. Lim, “Deep reinforcement learning-based routing on software-defined networks,” IEEE Access, vol. 10, pp. 18121–18133, 2022.

[13] H. Joo, S. Lee, S. Lee, and H. Kim, “Optimizing time-sensitive software-defined wireless networks with reinforcement learning,” IEEE Access, vol. 10, pp. 119496–119505, 2022.

[14] S. S. Mahgoub, M. M. Ashour, M. A. Yakout, and E. AbdElhalim, “Traffic classification in software defined networks based on machine learning algorithms,” International Journal of Telecommunications, vol. 4, no. 01, pp. 1–19, 2024.

[15] A. O. Salau and M. M. Beyene, “Software defined networking based network traffic classification using machine learning techniques,” Sci Rep, vol. 14, no. 1, p. 20060, 2024.

[16] N. Satheesh et al., “Flow-based anomaly intrusion detection using machine learning model with software defined networking for OpenFlow network,” Microprocess Microsyst, vol. 79, p. 103285, 2020.

[17] N. Bilal, S. Askar, and K. Muheden, “Challenges and Outcomes of Combining Machine Learning with Software-Defined Networking for Network Security and management Purpose: A Review,” The Indonesian Journal of Computer Science, vol. 13, no. 2, 2024.

[18] D. Nunez-Agurto, W. Fuertes, L. Marrone, and M. Macas, “Machine Learning-Based Traffic Classification in Software-Defined Networking: A Systematic Literature Review, Challenges, and Future Research Directions.,” IAENG Int J Comput Sci, vol. 49, no. 4, 2022.

[19] R. Setiawan et al., “Retraction Note: Encrypted Network Traffic Classification and Resource Allocation with Deep Learning in Software Defined Network,” 2025, Springer US New York.

[20] J. Xie et al., “A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges,” IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 393–430, 2018.

[21] Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, and Y. Sun, “A survey of networking applications applying the software defined networking concept based on machine learning,” IEEE access, vol. 7, pp. 95397–95417, 2019.

[22] M. U. Younus, M. K. Khan, and A. R. Bhatti, “Improving the software-defined wireless sensor networks routing performance using reinforcement learning,” IEEE Internet Things J, vol. 9, no. 5, pp. 3495–3508, 2021.

[23] L. Huo, D. Jiang, Z. Lv, and S. Singh, “An intelligent optimization-based traffic information acquirement approach to software-defined networking,” Comput Intell, vol. 36, no. 1, pp. 151–171, 2020.