

AI-Based Load Balancer for Software Defined Networking (SDN)

1. Jivan Kamble, 2. Sanjana Mane, 3. Ranjeet Mandake, 4. Yash Nangre, 5. Dhanashri Sutar (Student of Department of Computer Science and Engineering) 6. Prof - A.V. Patil (Assistant Professor, Dept. of Computer Science & Engineering)

Ashokrao Mane Group of Institutions, Vathar (416 112)

2. RELATED WORK

Several studies have explored the integration of SDN and load balancing techniques to improve network efficiency. Traditional SDN-based load balancing methods mainly utilize static algorithms such as Round Robin and Weighted Routing. Although these techniques provide basic traffic distribution, they fail to adapt dynamically under varying traffic conditions. Researchers have introduced machine learning-based approaches for predicting network congestion and optimizing routing paths. Some studies implemented reinforcement learning models for adaptive path selection, while others used supervised learning algorithms for traffic classification and congestion prediction. These systems demonstrated improvements in throughput and latency reduction compared to traditional methods. The proposed AI-based SDN Load Balancer improves upon previous approaches by integrating intelligent traffic prediction, dynamic routing decisions, congestion-aware load balancing, and dashboard-based monitoring within a single framework. The use of Mininet-based emulation further enables cost-effective experimentation and performance validation.

Methodology used:

The proposed methodology integrates Artificial Intelligence with Software Defined Networking to achieve intelligent and adaptive load balancing. A Mininet-based virtual SDN environment is created to simulate network traffic conditions. The SDN controller collects real-time traffic statistics from OpenFlow-enabled switches and forwards the data to the machine learning module for traffic prediction and congestion analysis. Based on AI-generated predictions, the controller dynamically updates routing policies and distributes traffic across available network paths. A monitoring dashboard is incorporated for visualization and performance evaluation using metrics such as throughput, delay, and packet loss.

Outcomes:

The experimental results obtained from Mininet-based simulations demonstrate that the proposed AI-integrated SDN

Abstract:

The rapid growth of cloud computing, IoT devices, and large-scale internet services has increased the demand for efficient network traffic management techniques. Traditional networking approaches suffer from static configurations and inefficient resource utilization under dynamic traffic conditions. To overcome these limitations, this project proposes an **Artificial Intelligence-Based Load Balancer using Software Defined Networking (SDN)**. The system integrates machine learning techniques with SDN architecture to intelligently distribute network traffic across multiple servers or network paths. The proposed framework improves network performance, minimizes congestion, reduces latency, and enhances resource utilization through centralized and programmable traffic control.

Keywords:

Network Emulator – Mininet, SDN (Software Defined Network) Controller – Ryu, Machine Learning – PyTorch, DQN – Deep-Q-Network, TFT – Temporal Fusion Transformer.

1. INTRODUCTION

Software Defined Networking (SDN) separates the control plane from the data plane, enabling centralized network management and programmability. Although SDN provides flexibility and dynamic control, conventional load balancing algorithms such as Round Robin and static routing fail to adapt efficiently to rapidly changing traffic conditions. The proposed system introduces an AI-driven load balancing mechanism capable of analyzing real-time network statistics and making intelligent routing decisions. By combining SDN with machine learning algorithms, the system dynamically predicts traffic conditions and distributes loads to optimize overall network performance.

framework performs more efficiently than conventional static load balancing approaches in terms of delay reduction, traffic optimization, and network stability.

3. LITERATURE REVIEW

1. In 2024, a study published in the *Journal of Cloud Computing* discussed QoS-aware load balancing techniques in SDN-based IoT networks and highlighted that intelligent traffic management significantly improves bandwidth utilization and network efficiency.
2. Another 2024 research published by *MDPI Applied Sciences* focused on optimization algorithms for routing and load balancing in SDN environments. The study demonstrated that dynamic optimization techniques effectively reduce network delay and congestion.
3. A 2024 paper in *Sensors (MDPI)* proposed a Temporal Deep Q-Learning approach for optimal load balancing in SDN. The research showed that reinforcement learning-based routing improves adaptive traffic distribution and enhances Quality of Service (QoS) under dynamic network conditions.
4. A 2024 study by **Mohammad Rostami and Salman Goli-Bidgoli** titled “*An overview of QoS-aware load balancing techniques in SDN-based IoT networks*” discussed intelligent SDN traffic management methods for IoT environments. The research concluded that AI-based load balancing improves bandwidth utilization, reduces congestion, and enhances Quality of Service (QoS).
5. In another 2024 paper, **Maria Daniela Tache, Ovidiu Păscuțoiu, and Eugen Borcoci** proposed “*Optimization Algorithms in SDN: Routing, Load Balancing, and Delay Optimization*”. The study focused on dynamic routing and optimization techniques to minimize delay and improve network performance in SDN architectures.
6. A 2024 research work by **Sonam Sharma, Dambarudhar Seth, and Manoj Kapil** titled “*Combined optimization strategy: CUBW for load balancing in software defined network*” introduced an efficient controller load balancing mechanism considering multiple network parameters. The proposed approach achieved better traffic distribution and controller performance.
7. In 2024, **L. Leo Prasanth and E. Uma** presented “*A computationally intelligent framework for traffic engineering and congestion management in software-defined network (SDN)*”. The paper proposed intelligent traffic engineering methods to reduce congestion and improve throughput in SDN environments.
8. Another 2024 study by **Junyan Chen, Wei Xiao, and Xinmei Li** titled “*Dynamic routing optimization in*

software-defined networking based on a metaheuristic algorithm” focused on intelligent routing optimization using SDN programmability. The proposed algorithm improved traffic allocation and reduced network delay.

9. A 2024 review paper by **Qin Du, Xin Cui, Haoyao Tang, and Xiangxiao Chen** titled “*Review of Load Balancing Mechanisms in SDN-Based Data Centers*” analyzed different SDN load balancing approaches used in data center networks. The study emphasized the importance of adaptive traffic scheduling and centralized network management.

4. PROPOSED WORK

1. Data Preparation:

The proposed system collects real-time traffic statistics from OpenFlow-enabled switches in the Mininet-based SDN environment. Network parameters such as bandwidth utilization, packet transmission rate, delay, and port statistics are continuously monitored by the Ryu controller using `OFPPortStatsRequest`. The collected traffic data is aggregated and stored in JSON format (`live_metrics.json`) through the `stats_collector` module. This dataset is then preprocessed and used for intelligent traffic analysis and load prediction.

2. Model Design:

The proposed model integrates Artificial Intelligence with Software Defined Networking for adaptive load balancing. The system uses a hybrid AI approach consisting of a Temporal Fusion Transformer (TFT) for time-series traffic forecasting and a Deep Q-Network (DQN) for reinforcement learning-based optimal path selection. The routing strategies are dynamically managed through the `routing_modes.py` module, enabling intelligent switching between traffic management algorithms based on real-time network conditions.

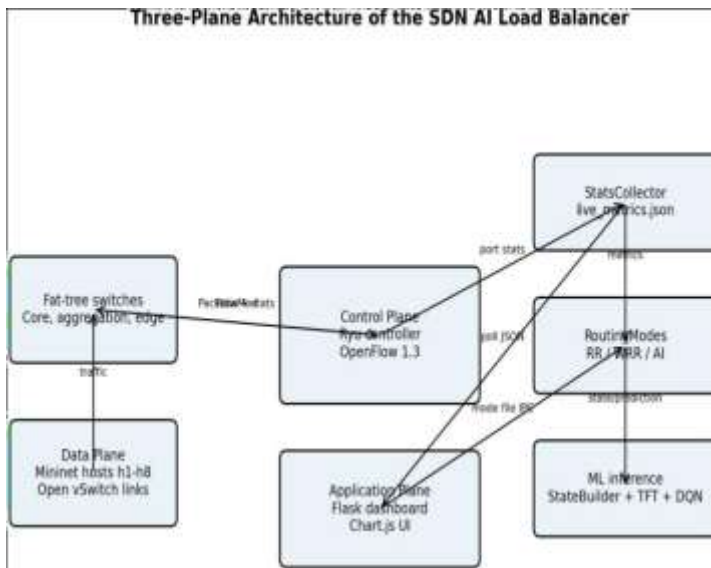
3. Model Training:

The AI model is trained using historical and real-time traffic statistics collected from the SDN environment. The TFT model learns traffic patterns and predicts future link load conditions, while the DQN agent is trained to select optimal network paths that minimize congestion and improve throughput. During training, the system evaluates performance metrics such as delay, packet loss, and bandwidth utilization to optimize routing decisions and enhance Quality of Service (QoS).

4. Model Architecture:

The architecture of the proposed system consists of three major planes: Data Plane, Control Plane, and Application Plane. The Data Plane uses Mininet with Open vSwitch (OVS) switches

for network emulation and traffic generation. The Control Plane utilizes the Ryu SDN Controller for traffic monitoring, flow management, and OpenFlow communication. The Application Plane contains the AI modules, including the TFT forecasting model and DQN-based routing agent, along with a Flask-based dashboard for real-time visualization using Recharts.js. The overall architecture enables intelligent traffic prediction, adaptive routing, centralized network control, and real-time performance monitoring.

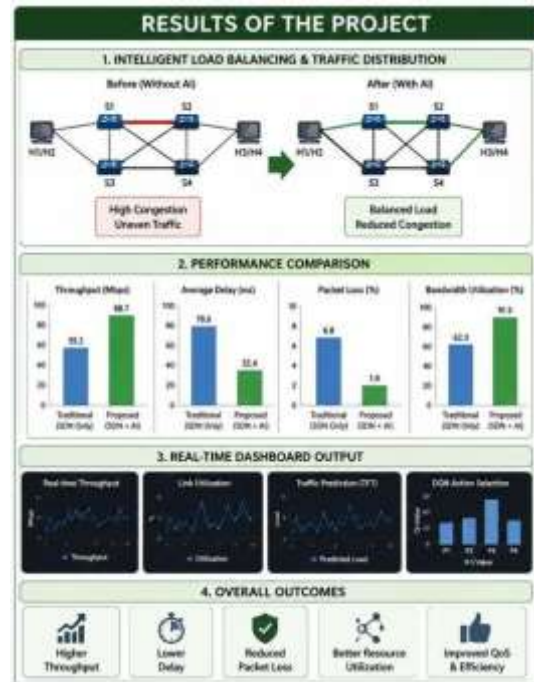


5. ENVIRONMENT OF THE PROJECT

The proposed AI-based SDN Load Balancer was implemented using Ubuntu 22.04 LTS on Oracle VirtualBox. Mininet and Open vSwitch (OVS) were used to simulate the SDN network environment, while the Ryu Controller managed traffic flow using the OpenFlow protocol. Python 3.10 was used for implementing AI models, controller logic, and automation scripts. The system integrates a TFT model for traffic prediction, a DQN model for intelligent routing, and a Flask-based dashboard with Recharts.js for real-time network monitoring and visualization.

6. RESULT

The implemented prototype demonstrates successful integration of the main architectural blocks: Mininet topology, Ryu controller, routing-mode selection, metric export, dashboard polling, and AI inference scaffolding. The current output confirms the selected routing mode and path index, while full path enforcement is identified as the next required enhancement.



7. CONCLUSION

The proposed AI-based SDN Load Balancer provides an intelligent and adaptive solution for modern network traffic management. By integrating Software Defined Networking with Artificial Intelligence techniques such as TFT-based traffic prediction and DQN-based routing, the system efficiently distributes network traffic and reduces congestion. The implementation using Mininet, OpenFlow, and the Ryu Controller demonstrates improved throughput, reduced delay, lower packet loss, and better resource utilization compared to traditional load balancing methods. The real-time monitoring dashboard further enhances network visibility and management, making the proposed system suitable for next-generation cloud, IoT, and data center networks.

8. REFERENCES

1. IBM Think, "What Is Software-Defined Networking (SDN)?", <https://www.ibm.com/think/topics/sdn>
2. Open Networking Foundation, "OpenFlow Switch Specification Version 1.3.0", <https://www.cs.yale.edu/homes/yu-minlan/teach/csci599-fall12/papers/openflow-spec-v1.3.0.pdf>
3. Ryu SDN Framework, <https://ryu-sdn.org/>
4. Mininet Overview, <https://mininet.org/overview/>
5. PyTorch Tutorials, "Saving and Loading Models", https://docs.pytorch.org/tutorials/beginner/saving_loading_models.html
6. PyTorch Tutorials, "Save and Load the Model", https://docs.pytorch.org/tutorials/beginner/basics/saveload_un_tutorial.html

7. Flask Documentation, "Blueprints and Views",
<https://flask.palletsprojects.com/en/stable/tutorial/views/>
8. Chart.js Documentation, "Line Chart" and "Bar Chart",
<https://www.chartjs.org/docs/latest/charts/line.html> and
<https://www.chartjs.org/docs/latest/charts/bar.html>
9. Lim, Arik, Loeff, Pfister, "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting",
<https://arxiv.org/abs/1912.09363>
10. Mnih et al., "Human-level control through deep reinforcement learning", Nature,
<https://www.nature.com/articles/nature14236>
11. Faezi and Shirmarz, "A Comprehensive Survey on Machine Learning using in Software Defined Networks (SDN)",
<https://link.springer.com/article/10.1007/s44230-023-00025-3>