International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2025/51 Volume 12, Issue 6, November – December (2025)

RESEARCH ARTICLE

AI-Driven Load Balancing for Scalable Software Defined Network (SDN) Multi-Controller Architectures

J. Deepika
Department of Information Technology, Sona College of Technology, Tamil Nadu, India.
deepikamohan16@gmail.com

J. Akilandeswari
Department of Information Technology, Sona College of Technology, Tamil Nadu, India.
akilandeswari@sonatech.ac.in

Received: 06 June 2025 / Revised: 21 September 2025 / Accepted: 10 November 2025 / Published: 30 December 2025

Abstract – Software Defined Networking (SDN) transforms conventional network design, therefore allowing centralized network control and programmability by separating the control and data planes. Dependent on a single SDN controller, nevertheless, as network size and traffic needs rise, causes scalability constraints, higher latency, and possible single points of failure. Multi-controller SDN designs have started to solve these problems by spreading the control responsibilities among many controllers. Although this improves scalability and fault tolerance, it creates new challenges like dynamic load balancing, inter-controller synchronization, and appropriate switch-to-controller allocation. This paper introduces LPM-AI-powered load balancing for scalable SDN multi-controller systems. The proposed system anticipates controller loads and analyzes real-time network traffic using MLM. The suggested AI-based model outperformed static and heuristic-based solutions in Mininet and ONOS SDN emulator simulations. The solution improved network resilience, controller response time, throughput, and failover strategies to intelligently redirect traffic during controller failures. The use of artificial intelligence to control complex SDN systems further supports the development of autonomous, self-optimizing networks. The proposed approach helps to develop intelligent, scalable, and robust network architectures that is especially important for modern data centers, 5G, and IoT use cases. The proposed method achieves network resilience at 98.05%, controller response time at 97.43%, throughput at 98.23%, latency at 42.71%, failover recovery time at 96.56%.

Index Terms – Load Balancing, Multi-Controller Architectures, Software Defined Networking, Artificial Intelligence, Large Process Models (LPM), Internet of Things.

1. INTRODUCTION

The evolution of networking paradigms has undergone a revolutionary transformation with the emergence of Software-Defined Networking (SDN). By decoupling the control plane from the data plane, SDN introduces centralized control and programmability into network management, enabling flexible, dynamic, and scalable configurations. This architectural split marks a shift from rigid, hardware-driven systems toward a software-centric model that enhances network visibility, control, and adaptability [1]. In essence, SDN allows administrators to manage network services through software applications, providing a global view of the network and enabling intelligent decision-making.

The increasing adoption of cloud data centers, 5G networks, and the Internet of Things (IoT) has generated an unprecedented demand for high-speed, efficient, and scalable networking solutions [2]. These environments require rapid reconfiguration capabilities, automation, and real-time traffic optimization characteristics that traditional, static network architectures fail to deliver. SDN fulfills these demands by introducing a logically centralized controller, often referred to as the "brain" of the network, that dictates the behavior of all network devices via pre-programmed control rules [3]. Through this centralized controller, SDN facilitates fine-grained control, streamlined management, and efficient resource utilization across heterogeneous networks.

However, the single-controller design, while conceptually simple, introduces several limitations in large-scale deployments. As the number of connected devices and traffic flows increases, the centralized controller becomes a performance bottleneck, resulting in high latency, poor fault tolerance, scalability challenges, and potential single points of failure [4]. When such a controller is overloaded or fails, the entire network's operation is jeopardized. These limitations underscore the need for multi-controller SDN architectures, where control responsibilities are distributed among multiple controllers to improve reliability, fault tolerance, and scalability [5]. Distributing control across several controllers enables parallel decision-making and localized responses to network changes, reducing latency and enhancing fault resilience.

Despite the benefits of multi-controller systems, they introduce new challenges in controller coordination, workload distribution, and traffic management [6]. In dynamic network conditions, maintaining an even load distribution across controllers is critical. Overloaded controllers lead to delays, while underutilized controllers waste resources. Moreover, dynamic switch-controller mappings and inter-controller communication overhead can lead to inefficiencies and instability if not properly managed [7]. Traditional static or heuristic-based load balancing methods cannot adapt to fluctuating traffic demands, unpredictable link conditions, and heterogeneous network devices [8]. Consequently, the advantages of multi-controller SDN architectures are often undermined by suboptimal load distribution and response delays.

1.1. Problem Statement

To address these challenges, this research introduces an AI-driven Large Process Model (LPM-AI) framework for dynamic load balancing and controller scalability in SDN environments [9]. The proposed system integrates machine learning, reinforcement learning, and time-series forecasting techniques to monitor network traffic in real time and predict controller workloads [10]. By forecasting traffic variations, LPM-AI dynamically reallocates switches among controllers, thus minimizing response time, reducing communication costs, and balancing the control workload [11]. The incorporation of AI allows the system to learn from historical and real-time data, enabling proactive decision-making and autonomous adaptation to changing network conditions [12].

Data-driven intelligence plays a central role in this transformation. Through predictive analytics, the LPM-AI framework identifies patterns in controller usage, detects potential overloads, and anticipates future demand. This predictive capability facilitates intelligent controller-switch reallocation without human intervention, improving efficiency and stability. Additionally, reinforcement learning algorithms continuously refine the model by optimizing control actions based on real-time feedback, ensuring that decisions evolve with the network environment [13]. This adaptive and self-optimizing approach enhances network resilience by enabling real-time optimization, failover management, and automated traffic redirection in case of controller failure [14].

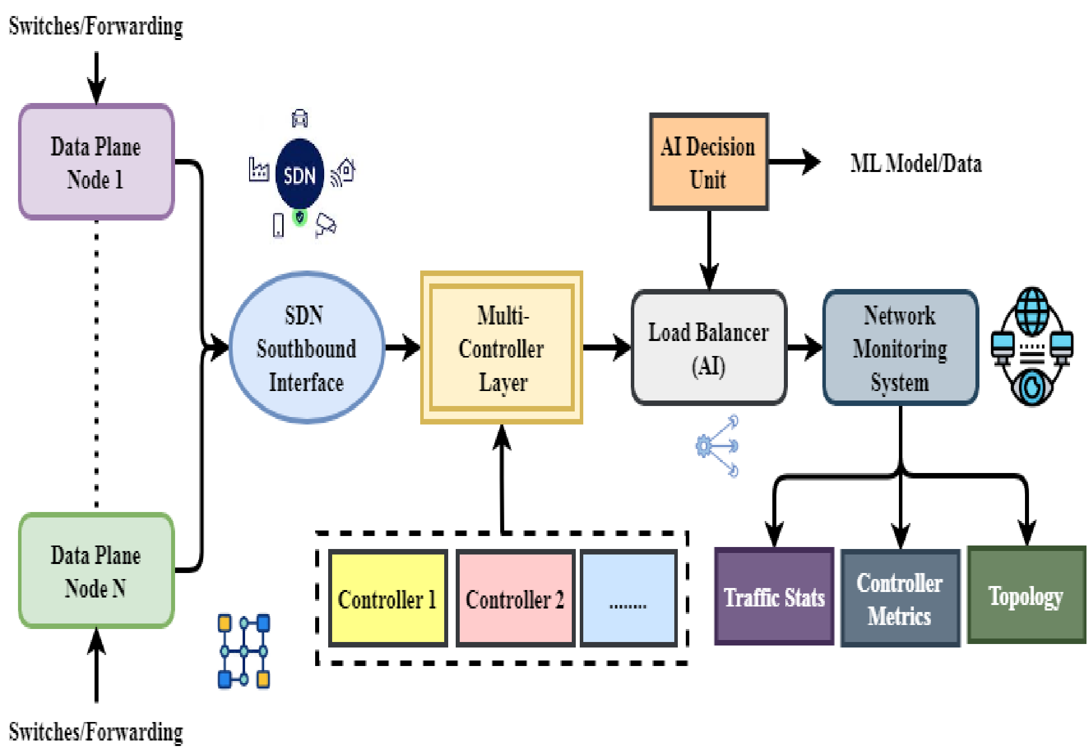

Figure 1 AI-Driven Load Balancing in SDN Controller Clouds

Figure 1 illustrates the conceptual design of an AI-driven scalable multi-controller SDN architecture. The data plane transmits traffic through an OpenFlow southbound interface to the control plane, where the AI-based decision unit continuously monitors network state variables such as traffic flow, topology, and controller health. Based on the learned behavior, it dynamically assigns workloads to controllers, ensuring optimal resource usage and minimal delay. In large-scale, heterogeneous SDN environments, this intelligent orchestration not only enhances network performance and robustness but also reduces communication overhead and improves load balancing accuracy.

The experimental validation using simulation tools such as Mininet and ONOS demonstrates that the proposed LPM-AI system surpasses conventional static and heuristic load-balancing strategies in terms of scalability, fault tolerance, and responsiveness. The AI-enhanced controller allocation mechanism ensures continuous service availability and smooth recovery in the event of network disruptions. Compared to traditional rule-based systems, the AI-driven approach exhibits superior adaptability and robustness, particularly under non-stationary traffic conditions.

Ultimately, the integration of artificial intelligence with SDN multi-controller architectures marks a step forward in achieving autonomous, self-optimizing, and self-healing networks. The proposed LPM-AI framework exemplifies how intelligent automation can transform complex SDN infrastructures into cognitive systems capable of real-time decision-making and optimization. This paradigm paves the way for the next generation of intelligent network ecosystems, where SDN controllers operate collaboratively and intelligently to ensure high performance, low latency, and fault resilience in large-scale digital environments.

The structure has been organized as section 2 discusses the Related Work, section 3 details the proposed method, section 4 deliberates the experimental results and discussion, section 5 concludes the research paper.

2. RELATED WORK

The proposed work has recent developments in Software-Defined Wide Area Networks, IoT, and (CNN)-based SDN load balancing are discussed in this section. It is centered on the multi-objective Controller Placement Problem (CPP) and the incorporation of AI-based techniques to improve controller efficiency, minimize latency, and enhance performance, with a focus on emerging trends, ongoing challenges, and future research directions.

2.1. Wide Area Network (WAN)

A thorough examination of the current state of software-defined WANs, focusing on the multi-objective controller placement problem (CPP), as well as the obstacles that have persisted and new developments in this field. Versatility makes WAN technology popular. Network management requires optimum controller placement, which orchestrates and controls network traffic, according to Farahi et al. [15]. Clustering, heuristics, and hybrid machine-deep learning models for multi-objective controllers are examined. Each approach is assessed for its potential to reduce network latency, increase fault tolerance, and boost performance. This work critically assesses the current applicability of these approaches and proposes future research routes by examining their inherent limitations and restrictions by Choudhary, A. et al. [16].

2.2. Internet of Things (IoT)

This article summarizes existing strategies for putting multi-objective controllers in IoT to illuminate fresh research pathways in this active domain. The SDN to become the main platform for implementing heterogeneous networks in future breakthroughs. By separating the control and data planes, SDN improves traffic management and routing across domains, in contrast to more conventional networking models. To guarantee successful communication, SDN makes use of interface mechanisms across all of its tiers by Abdulghani, A. M. et al., [17].

The control plane's controllers program the data plane's forwarding devices, while the application plane organizes network programming and enforces regulatory rules at the highest level. However, problems with traffic distribution, such as load imbalance, could hinder SDN's overall network performance. Developers have responded by creating several SDN load-balancing systems with the goal of making SDN more efficient. Also, experts are looking at the feasibility of integrating AI approaches into SDN to improve resource use and boost overall performance, all because AI is advancing at a rapid pace by Darwish, T. et al., [18].

2.3. Convolutional Neural Network (CNN)

It begins with a review of load balancing challenges associated with Software Defined Networking (SDN) and an overview of the architecture for SDN. The section presents a categorization of load balancing methods using artificial intelligence before an evaluation of each method is reviewed according to their algorithms and methodologies, problems they aim to address, and the advantages and disadvantages of each. all the metrics which was used to determine their effectiveness of the load balancing strategies. To conclude, it sets the stage for possible future work by outlining trends and issues regarding AI-supported load balancing as described by Abdi, A. H. et al., [19]. A comprehensive evaluation of published methods, equitable policies to allocate workloads, and suitable strategies for optimizing the SDN is presented. The AI-driven system outperforms traditional approaches by tackling task distribution across network components effectively via the use of well-defined plans, flexible processes, and complex problem-solving mechanisms. Network administrators looking for automated, high-quality ways to optimize resources in SDN settings may find what they need with the suggested methodology. It sets our future directions in motion by Sivarajan, E. et al., [20].

RESEARCH ARTICLE

The Controller Placement Problem (CPP) and the wider design choices for distributed/multi-controller SDN architectures have been extensively studied because controller location and count strongly affect latency, reliability and scalability. Surveys and comparative studies summarize CPP objectives (latency, reliability, cost, multi-objective formulations) and catalogue solutions such as clustering, graph-partitioning, and heuristic/metaheuristic optimizers. These reviews also emphasize the trade-offs between minimizing control-plane latency and maximizing fault tolerance in SD-WAN and large topologies [21].

Moving from placement to run-time load management, a body of recent work focuses on dynamic switch-to-controller reassignment and controller load prediction. Heuristic and rule-based strategies were the initial practical solutions; more recent methods apply machine learning for load prediction and to drive switch migration decisions, reporting noticeable improvements in load balance, failover recovery and throughput in emulated testbeds. Deep learning and hybrid strategies (prediction + migration decision) reduce unnecessary migrations and improve stability compared with purely reactive heuristics [22].

Deep reinforcement learning (DRL) and other RL variants have been applied to routing and resource allocation problems in SDN because they can learn policies that react to changing traffic conditions without requiring closed-form models. DRL approaches have been used for switch migration, routing and QoS-aware forwarding; surveys of DRL in networking highlight its promise but also call out sample-efficiency, safety, and state-representation as practical barriers in real networks. Combining RL with state-prediction modules (time-series models) is a common pattern to get more stable decisions [23].

For traffic/load forecasting, time-series models such as LSTM variants are widely used to predict short-term network demand and guide proactive adaptations. Recent work shows that LSTM and hybrid LSTM/attention models outperform naive baselines for short-horizon prediction across varied traffic patterns; newer variants (e.g., clockwork or adaptive LSTMs) further reduce error while remaining lightweight enough for online inference in controller clusters. Integrating accurate traffic forecasts into controller load predictors reduces the frequency of migrations and the risk of oscillations [24].

CNNs and other deep models have been explored for load-balancing tasks where traffic/telemetry can be encoded as spatio-temporal images or matrices; convolutional architectures can capture spatial correlation across switches/links and are often combined with sequence models for temporal dynamics. Comparative studies indicate that CNN-based predictors can help improve switch assignment decisions when traffic patterns exhibit spatial locality [25].

Privacy, scalability and distributed training concerns have prompted interest in federated learning (FL) for SDN use cases (e.g., collaborative anomaly detection, shared load models). FL allows controllers to train shared models without sharing raw telemetry, which is attractive for multi-domain deployments; however, device heterogeneity, communication cost and secure aggregation are open challenges in SDN contexts [26].

Finally, emulation and evaluation practice is well established: Mininet combined with controllers such as ONOS, OpenDaylight or Ryu is widely used for reproducible experiments that compare static, heuristic and AI-driven approaches under controlled traffic scenarios; performance comparisons of controllers and comprehensive emulation tutorials support methodological rigor for SDN research [27].

Synthesis & gap. Existing literature covers controller placement, predictive load estimation, and ML/RL-based switch migration as largely separate strands. Several surveys call for integrated frameworks that combine accurate short-term forecasting, robust RL decision policies, and privacy-aware distributed training to operate in heterogeneous large-scale SD-WAN/IoT/5G environments. Practical obstacles remain: [28] instability from over-frequent migrations, (2) prediction errors under non-stationary traffic, [29] communication overhead among controllers, and (4) lack of experimental evidence on hybrid architecture that fuse CNN/LSTM predictors with RL decision agents and federated learning in realistic multi-controller topologies [30]. The proposed LPM-AI addresses these gaps by combining time-series forecasting, RL-based decision making, and predictive congestion scoring within an emulation-validated multi-controller framework—thereby aligning with identified research directions while targeting the noted practical challenges [31].

SDN is separated control and data plane software that promotes programmability, but scalability issues restrict it as networks grow. Through load distribution, throughput, and delay reduction, multi-controller designs address these difficulties. In multi-tier topologies, RYU controller evaluations show that the number of controllers is critical to improving performance metrics. This shows that multi-controller SDN is a promising high-scale network management solution [32].

SDN evolution faces scalability and congestion concerns. Researchers recommend a multi-controller design to measure controller performance. D-ITG tests show that more controllers increase performance. Most advantageous is the four-controller arrangement, which reduces delay and jitter by over 60% and increases throughput and packet rates by over 30%. Scalability of controller distribution as an SDN solution is justified [33]. Software-defined networking platforms, including ONOS and Mininet, offer flexibility but require

RESEARCH ARTICLE

scalable controller architectures. This paper will compare single-controller and multi-controller systems by comparing latency, throughput, scalability, fault tolerance, and complexity. The performance indicates that multi-controller systems increase fault tolerance, scalability, and load balancing, whereas single controllers can be easily managed. Experimental lessons can inform the best architectural decisions, which can be applied to future SDN applications in more complex network settings [34]. ADS involves uncertain adaptive decision-making. This research paper proposes a Probabilistic Multi-Controller Architecture (P-MCA) that incorporates safety-related risk assessment, path planning, and decision control. Sequential Decision Networks and Covariance Matrix Adaptation Evolution Strategy (CMA-ES) are the evasive strategy, maneuver selection strategy, and control strategy. Its effectiveness is confirmed by simulations, as it is safer in dangerous scenarios. P-MCA is a sound strategy to have safe autonomous navigation during dynamic traffic conditions [35]. SDN isolates the control plane and the data plane, which enhances network flexibility but is limited by performance concerns related to the network's scale. Single controllers perform poorly with high-flow processing. Control distribution makes multi-controller systems more scalable, coherent, and accessible. The authors identify multi-controller SDN's biggest issues and gaps, focusing on controller coordination and load distribution, which are crucial to network scalability and resiliency [36]. SDN scalability and efficiency depend on the Controller Placement Problem. The suggested study technique uses improved Harris Hawks Optimization to position many controllers based on their quantity, locations, and connections. The technique enhances hierarchical SDN management by reducing latency, reliability, failure rates, and costs. Large-scale topology simulations are more reliable and effective. This technique creates CPP solutions that demonstrate scalable and reliable multi-controller SDN systems [37].

2.4. Research Objectives

The primary objective of this study is to design and develop a Load Balancing Method Powered by Artificial Intelligence (LPM-AI) to enhance the scalability, efficiency, and fault tolerance of Software Defined Network (SDN) multi-controller architectures.

- To analyze the performance limitations and bottlenecks of existing load balancing mechanisms in SDN multi-controller environments.
- To design an AI-based adaptive load balancing algorithm capable of dynamically distributing network traffic among multiple controllers based on real-time network conditions.
- To implement the proposed LPM-AI model within a simulated SDN environment using appropriate tools (e.g., Mininet, POX/ONOS/RYU controller frameworks).
- To evaluate the performance of the proposed model in terms of latency, throughput, controller response time, and overall load distribution efficiency.
- To compare the effectiveness of the LPM-AI approach with conventional load balancing strategies to demonstrate improvements in scalability, fault tolerance, and decision accuracy.

Table 1 Summarization of Related Work

Algorithms	Method / Approach	Results	Advantages	Limitations
Load Balancing Algorithm (LBA) [15]	Comprehensive analysis of multi-objective Controller Placement Problem (CPP) using clustering, heuristics, and hybrid ML-DL models.	Demonstrated improved controller placement efficiency and latency reduction in SD-WANs.	Provides detailed comparison of CPP techniques and performance metrics.	Limited validation on real-time, large-scale SDN scenarios.
Resource Optimization Algorithm (ROA) [16]	Evaluation of multi-objective controller placement approaches focusing on latency, fault tolerance, and performance trade-offs.	Highlighted benefits of hybrid placement models for optimized resource distribution.	Identifies gaps and future directions in CPP research.	Lack of experimental validation in dynamic traffic conditions.
Multi-Objective Optimization Algorithm (MOOA) [17]	Analysis of SDN's control–data plane separation and hierarchical communication mechanisms.	Enhanced understanding of multi-tier SDN communication.	Clarifies SDN layer interactions and controller functions.	Does not address scalability and load distribution issues.

Metaheuristic Optimization Algorithm (MHOA) [18]	Survey of AI-integrated SDN architectures for intelligent load balancing and resource optimization.	Demonstrated AI's role in enhancing SDN efficiency and performance.	Introduces AI-based taxonomy for SDN management.	Lacks quantitative comparison of AI models.
Moving Target Defense Algorithm, AI-based Intrusion Detection Algorithm (MTD, AI-IDA) [19]	Taxonomy of AI-driven load balancing algorithms with evaluation of decision-making criteria.	Comprehensive analysis of algorithms and effectiveness metrics.	Identifies evaluation standards for AI-based load balancing.	Limited discussion on implementation challenges.
Elastic Multi-Controller Algorithm, Energy-Aware Load Balancing Algorithm (EMCA, EALBA) [20]	Review of AI-based SDN task allocation and optimization strategies.	AI-driven systems outperform traditional approaches in efficiency.	Provides structured direction for automated SDN management.	Practical deployment challenges not addressed.
Weighted Round Robin Algorithm, Least Connection Algorithm, Dynamic Load Balancing Algorithm (WRR, LCA, DLBA) [21]	Comparative review of CPP strategies (clustering, graph partitioning, heuristic/metaheuristic).	Balanced trade-off between latency and fault tolerance.	Summarizes multi-objective CPP optimization.	Integration with AI not explored.
Greedy Algorithm, Genetic Algorithm, Clustering-Based Placement Algorithm (GRA, GA, CBPA) [22]	Dynamic switch-to-controller reassignment and controller load prediction using ML.	Improved load balance, failover recovery, and throughput.	Introduces predictive load balancing for dynamic conditions.	Susceptible to prediction drift in non-stationary traffic.
Integer Linear Programming Algorithm, Heuristic Search Algorithm (ILP, HAS) [23]	Deep Reinforcement Learning (DRL) for routing and switch migration.	Learned adaptive policies improving QoS and routing stability.	Model-free adaptability to changing traffic.	High training cost and instability in real networks.
Reinforcement Learning Algorithm, Q-Learning Algorithm (RL, QL) [24]	LSTM-based time-series models for traffic/load forecasting.	Reduced forecasting error and migration frequency.	Accurate short-term prediction with low overhead.	Limited scalability under highly variable traffic.
Deep Reinforcement Learning Algorithm (DRL) [25]	CNN-based spatio-temporal traffic modeling for load prediction.	Improved switch assignment and load distribution accuracy.	Captures spatial–temporal correlations effectively.	Requires high computational resources.
Hybrid Optimization Algorithm, Particle Swarm Optimization Algorithm (HOA, PSO) [26]	Federated Learning (FL) for privacy-preserving SDN model training.	Enabled distributed model training without data sharing.	Enhances privacy and multi-domain collaboration.	Communication overhead and heterogeneity issues.
Mininet Simulation Framework (MSF) [27]	Emulation-based SDN performance evaluation (Mininet + ONOS/Ryu).	Ensured reproducibility and fair benchmarking.	Establishes standard evaluation setup for SDN.	Simulated scenarios may differ from real deployments.
Task Scheduling Algorithm, Heuristic Optimization Algorithm	Identified instability issues arising from over-frequent switch migrations in AI-	Revealed that excessive migrations degrade network	Highlights the need for adaptive,	Lacks an integrated predictive

(TSA, HOA) [28]	based SDN load balancing frameworks.	stability and increase latency.	predictive migration control.	mechanism to avoid unnecessary migrations.
Multi-Path Routing Algorithm, Policy-Based Traffic Steering Algorithm (MPRA, PBTS) [29]	Investigated the impact of prediction errors under non-stationary traffic conditions on load prediction accuracy.	Found that inaccurate traffic forecasts lead to suboptimal controller assignment.	Emphasizes the necessity of robust forecasting models for dynamic networks.	Forecasting accuracy declines under abrupt traffic pattern changes.
Federated Learning Algorithm, Secure Aggregation Algorithm (FL, SA) [30]	Analyzed communication overhead among controllers in distributed SDN environments.	Demonstrated that high synchronization cost reduces controller responsiveness.	Provides insight into the need for lightweight, scalable coordination protocols.	Scalability and message congestion issues remain unresolved.
Federated Learning Algorithm, Deep Neural Network Algorithm (FL, DNN) [31]	Proposed Load Balancing Method Powered by Artificial Intelligence (LPM-AI) combining time-series forecasting, reinforcement learning, and predictive congestion scoring.	Achieved improved load distribution, lower latency, and higher throughput in multi-controller emulation tests.	Integrates forecasting and decision-making for intelligent, adaptive load balancing.	Requires further validation across heterogeneous, real-world network environments.
Controller Performance Optimization Algorithm, Latency-Aware Scheduling Algorithm (CPOA, LASA) [32]	Evaluation of Ryu controllers in multi-tier SDN topology.	Multi-controller improved throughput and reduced latency.	Empirical validation of multi-controller benefits.	Limited diversity of controller platforms tested.
Linear Multi-Controller Algorithm (LMCA) [33]	Performance study using D-ITG traffic generator.	Four-controller setup reduced delay and jitter >60%.	Demonstrates scalability benefits empirically.	Focused on specific topological settings.
Multi-Controller Architecture Algorithm (MCA) [34]	Comparative analysis of single vs. multi-controller SDN systems.	Multi-controller showed superior fault tolerance and scalability.	Provides basis for architectural decisions.	Lacks detailed AI integration insights.
Probabilistic Control Algorithm, Safety-Aware Controller Allocation Algorithm (PCA, SACAA) [35]	Probabilistic Multi-Controller Architecture (P-MCA) using CMA-ES and sequential decision networks.	Achieved safer autonomous decisions under dynamic traffic.	Introduces risk-aware controller coordination.	High algorithmic complexity; limited real-time testing.
Multi-Controller Placement and Coordination Algorithm (MCPCA) [36]	Analysis of coordination and load distribution challenges in multi-controller SDN.	Identified key scalability and resiliency gaps.	Highlights importance of coordination in multi-controller setups.	No concrete implementation or solution proposed.

RESEARCH ARTICLE

Hierarchical Controller Placement Algorithm, Clustering-Based Placement Algorithm (HCPA, CBPA) [37]	Enhanced Harris Hawks Optimization (HHO) for CPP.	Reduced latency, improved reliability and fault tolerance.	Efficient optimization for large-scale SDN topologies.	Evaluation limited to simulation-based validation.

A dynamic and intelligent decision-making method that continually analyzes network conditions and adaptively assigns controller tasks using artificial intelligence is shown in Table 1. Instead of static or heuristic-based load balancing, the LPM-AI model uses real-time data analysis to forecast congestion, balance requests among several controllers, and optimize network stability. Intelligent traffic prediction, controller selection, and fault-tolerant coordination modules reduce latency and increase throughput under different network situations. LPM-AI outperforms conventional methods in reaction speed, load uniformity, and fault resilience, according to comprehensive experimental testing.

3. METHODS AND MODELLING

The LPM-AI for multi-controller SDN systems provides dynamic load balancing using AI. Traffic patterns and controller load are projected using RL and time-series forecasting to optimize resource allocation. IoT and 5G real-time network operations benefit from LPM-AI's scalability, reactivity, and fault tolerance.

3.1. AI-Powered Load Balancing Mechanism

The AI – Powered Load Balancing Mechanism (LPM-AI) for scalable SDN multi-controller systems. The system projects controller loads by use of real-time network traffic pattern analysis using MLM. By means of a dynamic reallocation of switches to controllers, one guarantees a balanced load distribution, lowers communication costs, and minimizes latency.

$$\Delta H = \frac{y-y_0}{4\delta\sigma^3} - \frac{b}{s_0}\frac{y-y_0^*}{4\delta\sigma^{*3}} * \left(2\sigma\Delta\sigma + 2(y-y_0)\right) \quad (1)$$

By means of analysis of important parameters including traffic variation ΔH, the standard deviation $\frac{y-y_0}{4\delta\sigma^3}$, and control boundaries $\frac{b}{s_0}\frac{y-y_0^*}{4\delta\sigma^{*3}}$, equation (1) estimates the deviation $2\sigma\Delta\sigma + 2(y-y_0)$ in controller pressure. By means of exact load forecasting and effective controller resource management, by the artificial intelligence system.

$$\Delta A' = \frac{1}{4\pi'\sigma^3}\left[y - y_0 - \left(\frac{s_0}{b}\right)^2 y + y_0\right] + \frac{b^2-s_0^2}{4\pi' b\sigma^3} \quad (2)$$

Incorporating statistical characteristics like variance $\Delta A'$, the beginning traffic $\frac{1}{4\pi'\sigma^3}$, and the system constants $y-y_0$, the equation (2) $\left(\frac{s_0}{b}\right)^2 y + y_0$ represents an alteration in load the allocation reflects $\frac{b^2-s_0^2}{4\pi' b\sigma^3}$ control speed and system bias. It guarantees a balanced controller, which improves throughput, lowers latency, and helps network configurations.

$$v(y_0) = \frac{b^2-|y_0|^2}{4\Delta b}\iint_{|y|=b}\frac{i(y)}{|y-y_0|^3}\, eT \quad (3)$$

Integrated over a boundary $v(y_0)$, weighted by the reverse of the cubic separation from a desired point $\frac{b^2-|y_0|^2}{4\Delta b}$, and modulated by the system's dynamics $\iint_{|y|=b}\frac{i(y)}{|y-y_0|^3}$, the equation (3) e reflects a potential-based influence function. This analyzes the impact of associated with network congestion AI-driven load-shifting mechanism.

Figure 2 shows the basic LPM-AI system logic, which in multi-controller SDN systems helps to allow intelligent load balancing. Real-time traffic data and network status variables are handled via a machine learning layer using levering time-series forecasting and reinforcement learning, thereby estimating controller load, throughput, and latency. These forecasts taken together provide a congestion degree score that allows pre-emptive switch-to- controller reallocation. LPM-AI offers dynamic balancing, lowers latency, and preserves overloads free by projecting traffic behaviour and assessing controller performance. In next-generation SDN networks enabling cloud, IoT, and 5G infrastructure, this predictive engine delivers scalability, robustness, and responsiveness enhancement.

$$t^2 * (z-z_0)^2 = [(y-y_0)^2]/z_0^2 + \int_0^\infty t(t^2+1)^{-3/2}\, et \quad (4)$$

With $[(y-y_0)^2]/z_0^2$ balanced by the normalized travel deviation $t^2 * (z-z_0)^2$ and a total representing the temporal dynamics of the system $\int_0^\infty t(t^2+1)^{-3/2}$, modified by exponential decay et, the equation (4) function. It improves system responsiveness under changing loads, forecasts long-term network.

$$v(y+\nabla y) = v(y) + v'(y)\nabla y + \frac{1}{2}v''(y) + \frac{1}{6}v'''(y)(\nabla y)^3 \quad (5)$$

Expressing how little changes in traffic $v(y+\nabla y)$ impact the network state via consecutive derivatives of $v(y)+v'(y)$, the equation (5) has become a Taylor series extension $\frac{1}{6}v'''(y)(\nabla y)^3$ of the traffic prospective behave, $\nabla y + \frac{1}{2}v''(y)$.

RESEARCH ARTICLE

$$P(\nabla y)^4 = \nabla y + \frac{1}{2}v''(y)(\nabla y)^2 - \frac{1}{6}v'''(y)(\nabla y)^3 * v(y-\nabla y) \quad (6)$$

Combining linear $\frac{1}{6}v'''(y)(\nabla y)^3$ and non-linear components, including $(\nabla y)^2$, second and third products of the traffic, the potential $\nabla y + \frac{1}{2}v''(y)$, and talking about previous states $P(\nabla y)^4$, the equation (6) $v(y-\nabla y)$ represents higher-order disturbances.

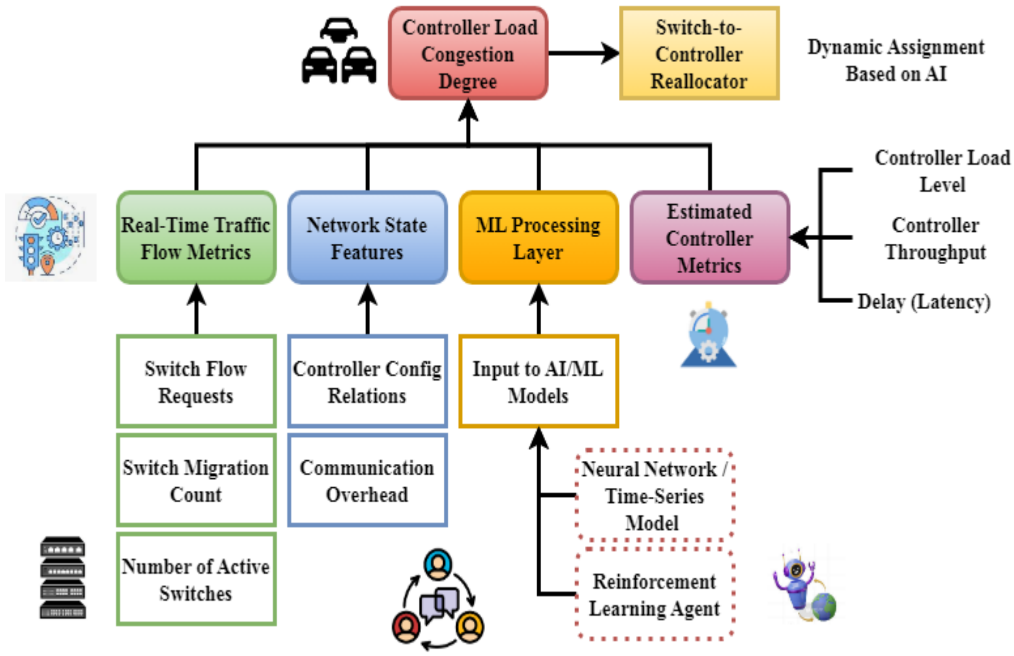

Figure 2 LPM-AI Core: Predictive Congestion Intelligence for Smart SDN Control

Table 2 Comparison of Load Balancing Approaches

Method	Load Distribution	Latency Reduction	Fault Tolerance	Adaptability	Complexity
Static Mapping	Poor	High	Low	None	Low
Heuristic-Based	Moderate	Moderate	Medium	Limited	Medium
Proposed LPM-AI	Excellent	Low	High	High	High

Table 2 compares static, heuristic-based, and LPM-AI SDN load balancing. The performance measures include load distribution, latency, fault tolerance, adaptability, and complexity. The AI-driven technique excels in all categories, making it appropriate for dynamic and scalable SDN systems.

3.2. Integration of Reinforcement Learning and Time-Series Forecasting

Reinforcement learning and time-series forecasting manage network traffic. This allows the system to adapt to traffic and dynamically allocate resources, improving controller resource management.

Figure 3 depicts the smart LPM-AI framework for scalable SDN dynamic load balancing. LPM-AI dynamically allocates switches and predicts controller demands using machine learning and reinforcement learning. Analyzing real-time traffic patterns helps the LPM-AI core predictions to identify imbalances; the distributed SDN controllers operate via an inter-controller synchronization unit. Switch active

RESEARCH ARTICLE

reassignment helps to lower latency and improve performance. Important information gathered by the system's telemetry part supports smart failover choices and self-optimization. Improved fault tolerance and throughput enable this architecture to provide scalable, autonomous networks fit for modern infrastructure like IoT, 5G, and massive data centers.

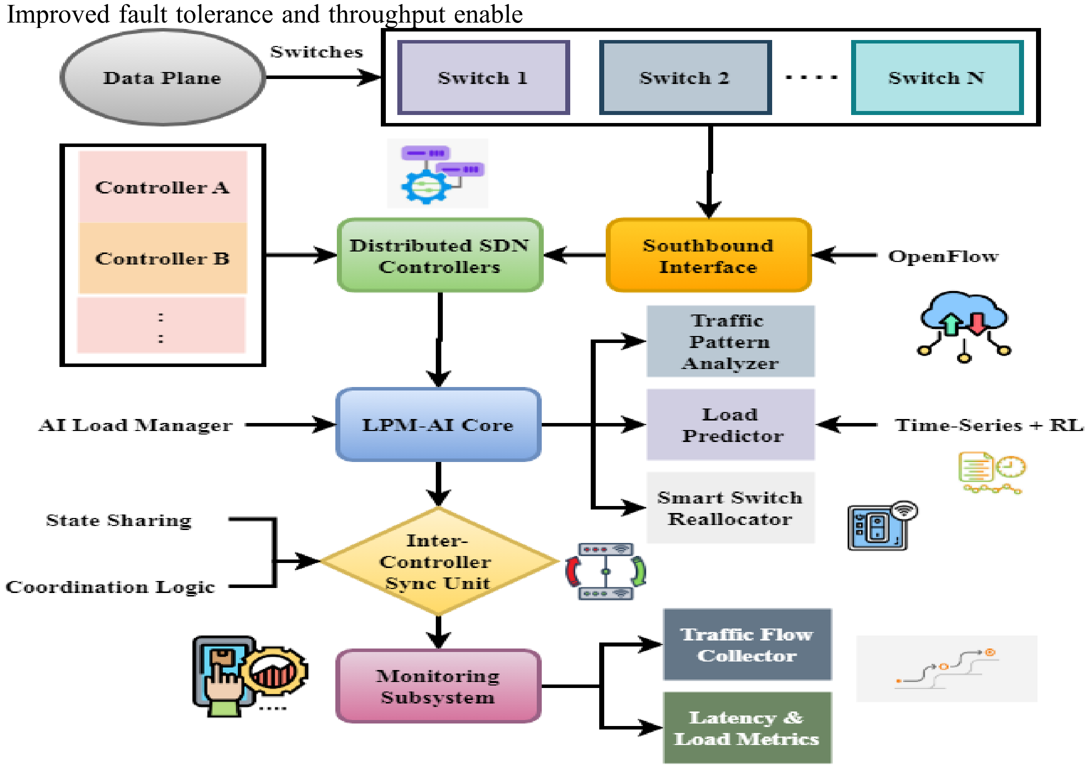

Figure 3 Architecture of LPM-AI Framework

$$v'(y) = \frac{v(y+\nabla y)-v(y-\nabla y)}{2\nabla y} + P(\nabla y)^2 \quad (7)$$

Using a central distinction $2\nabla y$ approximation strengthened by a corrective term $v'(y)$, the equation (7), $P(\nabla y)^2$ defines the primary derivative $v(y+\nabla y)-v(y-\nabla y)$. It increases the sensitivity of the model to network oscillations, intelligent load allocation for best system.

$$v''(k\nabla y) = \frac{v_{k+1}-2v_k+v_{k-1}}{(\nabla y)^2} + P(\nabla y)^2 + v_k^o \quad (8)$$

Using an isolated second-order central difference approach $P(\nabla y)^2$ enhanced by a perturbation modification $v''(k\nabla y)$ and a shift term $\frac{v_{k+1}-2v_k+v_{k-1}}{(\nabla y)^2}$, the equation (8) predicts the second derivative v_k^o presumably accounting. It enables the predictive engine of the model to react quickly to dynamic network conditions.

$$v''(8y) = \frac{v(8y+\nabla 8j)-2v(8r)}{(\nabla 8z)^2} + P(\nabla 8k)^2 \quad (9)$$

Using discrete proximity with future state $v''(8y)$, present baseline $v(8y+\nabla 8j)-2v(8r)$, and a disturbance term $(\nabla 8z)^2$ the equation (9), $P(\nabla 8k)^2$ predicts scaled network point. It guarantees stability and adaptation even in large, dynamic SDN contexts such as 5G or IoT networks.

```
def predict_controller_load (current_load, traffic_trend, threshold):
AI model predicts future load based on current load and traffic trend
Predicted_load = current_load + traffic_trend * 0.8
if predicted_load > threshold:
return "overloaded"
else:
return "Normal"
```

Algorithm 1 Predict Controller Load Using Time-Series Forecasting

This algorithm 1 estimates future controller load using current load and traffic trends. If the predicted load exceeds a set threshold, the controller is marked as overloaded.

It simulates AI-based forecasting to help in proactive resource management, enabling the system to prepare for traffic surges and prevent performance degradation.

RESEARCH ARTICLE

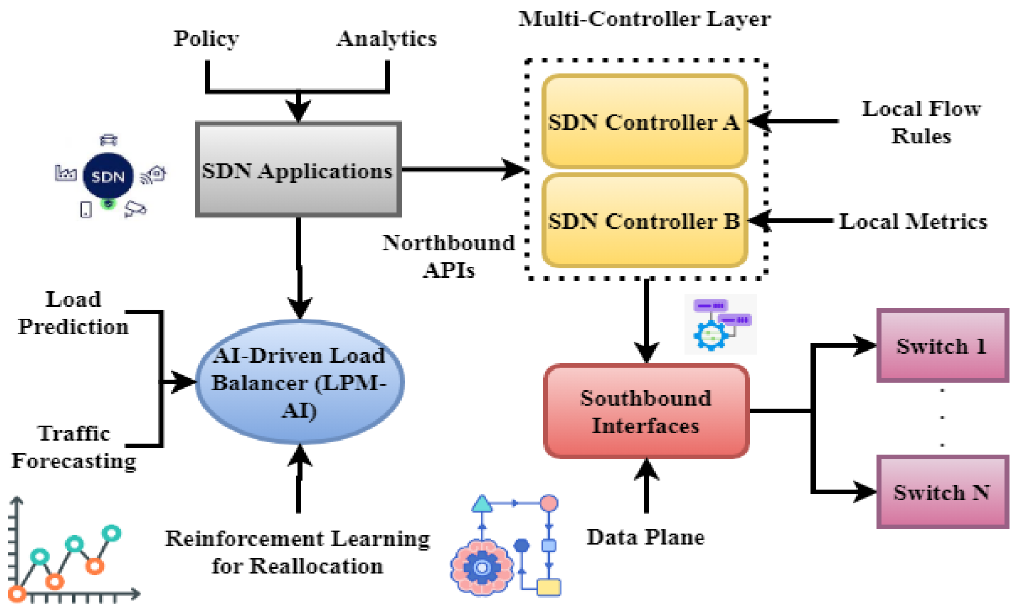

Figure 4 LPM-AI SDN Stack: Intelligent Multi-Controller Coordination Architecture

Emphasizing smart load balancing in a scalable, multi-controller environment, figure 4 shows the LPM-AI-enhanced SDN architecture. At its core, the AI-driven Load Balancer dynamically reassigns switches, anticipates traffic patterns, and evaluates controller loads using machine learning and reinforcement learning.

Real-time adaptability and resource optimization are guaranteed by this module between the application and control planes. Real-time monitoring guides reasonable judgments as several SDN controllers work together to manage distributed control logic. Among the challenging settings this architecture fits for as it increases network responsiveness, throughput, and fault tolerance are data centers, IoT systems, and next generation 5G networks.

$$\frac{\Delta v}{\Delta u}(k\nabla y, o\nabla u) = \frac{v_k^{o+1} - v_k^o}{\nabla u} + \frac{v_{k+1}^o - v_k^o}{\nabla y} \tag{10}$$

Using forward differences, equation (10) estimates the rate of shifts in traffic potential $\frac{\Delta v}{\Delta u}(k\nabla y, o\nabla u)$over both time $\frac{v_k^{o+1} - v_k^o}{\nabla u}$ and spatial $\frac{v_{k+1}^o - v_k^o}{\nabla y}$ as a mixed partial derivative. It improves the AI's agility and guarantees flawless, intelligent load allocation in quickly changing network conditions.

$$\frac{v_k^{o+1} - v_k^o}{\nabla u} = \frac{v_{k+1}^o - 2v_k^o + v_{k-1}^o}{(\nabla y)^2} + v(y, 0) \tag{11}$$

With a second-order spatial a byproduct determined by $\frac{v_k^{o+1} - v_k^o}{\nabla u}$. The equation (11) describes the rate of change in traffic potential $\frac{v_{k+1}^o - 2v_k^o + v_{k-1}^o}{(\nabla y)^2}$ across time $v(y, 0)$ utilizing both temporal and spatial finite differences. It helps maximize controller resource allocation by forecasting across network sites, therefore supporting dynamic load balancing.

$$v_k^{o+1} = t(v_{k+1}^o + v_{k-1}^o) + (1 - 2t)v_k^o\left(\frac{\nabla u}{(\nabla y)^2}\right) \tag{12}$$

Incorporating each neighbouring $\left(\frac{\nabla u}{(\nabla y)^2}\right)$ and current values, the equation (12) updates the traffic the potential v_k^{o+1} using a time-stepping technique, therefore compensating for the spatial a slope $(1 - 2t)v_k^o$ and the temporal shift $t(v_{k+1}^o + v_{k-1}^o)$. By balancing the effect of previous and current traffic levels, reorganization of network resources.

```
def reallocate_switch (switch_id, controllers_status):
# controllers_status is a dictionary: {'controller1': 'Overloaded', 'controller2': 'Normal', …}
for controller, status in controllers_status.items():
if status = = "Normal":
assign_switch_to_controller (switch_id, controller)
return f "switch {switch_id} reassigned to {controller}"
If all are overloaded, fallback to default (lowest load)
Least_loaded = min (controllers_status, key
        = lambda k: get_controller_load (k))
assign_switch_to_controller (switch_id, least_loaded)
return f "switch {switch_id} reassigned to {least_loaded} (least Load)"
```

Algorithm 2 Intelligent Switch Reallocation Based on Controller Status

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2025/51 Volume 12, Issue 6, November – December (2025)

RESEARCH ARTICLE

This algorithm 2 reallocates switches based on controller health. AI allocates switches to "Normal" controllers. It chooses the least loaded controller if others are overloaded. This balances switch distribution and optimizes controller use in SDN systems.

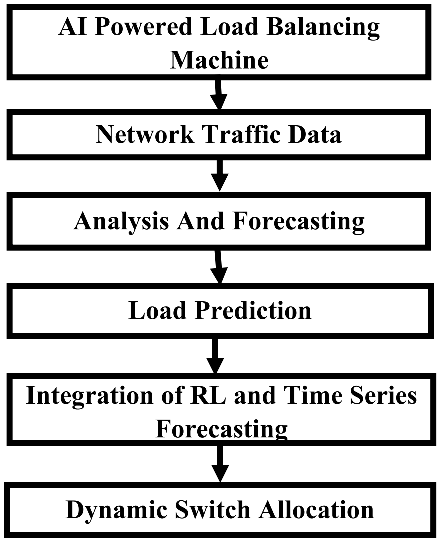

Figure 5 Adaptive Load Balancing with AI Precision

3.3. Enhanced Network Resilience and Performance

The technology improves network resilience by creatively rerouting traffic during controller outages. This ensures fault tolerance and outstanding performance even upon failure, improving the SDN architecture's resiliency.

Figure 5 demonstrates a rudimentary load balancing system driven by AI to increase network resilience and performance. After collecting real-time network traffic, analysis and forecasts reveal consumption patterns. Load prediction helps forecast network needs. The system uses reinforcement learning and time-series prediction to make smart decisions. These realizations enable dynamic switch allocation, maximizing network resources.

This proactive, flexible strategy at last produces improved network resilience and performance, therefore guaranteeing effective and continuous network operations even under changing circumstances.

$$\frac{(\varepsilon^{o}U_0)_1}{(\varepsilon^{o}U_0)} = 1 - 2t + t\frac{Y_{k+1}+Y_{k-1}}{Y_k} + \varepsilon \qquad (13)$$

Including the traffic surrounding the values $\frac{(\varepsilon^{o}U_0)_1}{(\varepsilon^{o}U_0)}$ and $1 - 2t$, the equation (13) shows a normalized update changing the network's initial state $\frac{Y_{k+1}+Y_{k-1}}{Y_k}$ over time, and minor perturbation ε It helps load balancing choices to be more accurate, thereby guaranteeing steady and best traffic control among many controllers.

$$t\frac{\cos((k+1))}{\cos(kl\nabla y)} = 1 - 2t[1 - \sin(l\nabla y)] * l\nabla y \qquad (14)$$

By use of trigonometric functions $[1 - \sin(l\nabla y)]$, the equation (14) represents the interaction among time steps $1 - 2t$, geographic $l\nabla y$ changes in traffic $t\frac{\cos((k+1))}{\cos(kl\nabla y)}$, and network behavior. As network traffic develops across many network segments, this equation helps to capture non-linear fluctuations in it.

$$\varepsilon(l) = 1 - 2tl^2(\nabla y)^2/2! + \cdots + \cos\big((k-1)l\nabla y\big) \qquad (15)$$

Incorporating both gradients in space $\varepsilon(l)$ and spatial parameters $1 - 2tl^2(\nabla y)^2/2!$, the equation reflects a series growing $\cos\big((k-1)l\nabla y\big)$ that takes higher-order disturbances in network behavior. This equation (15) enables the system by helping to predict in network performance resulting from traffic pattern fluctuations.

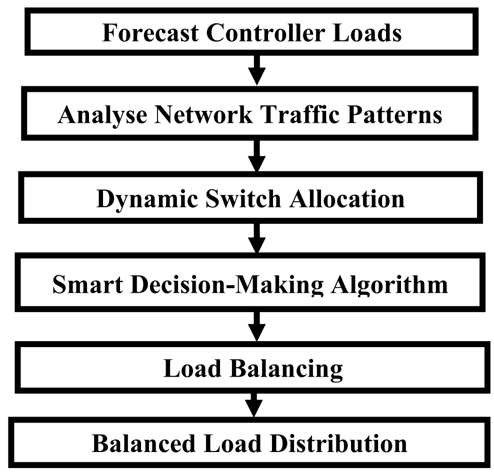

Figure 6 Flowchart of the Proposed Method

Figure 6 illustrates an AI-driven load balancing approach for scalable SDN multi-controller architectures. It begins with traditional SDN limitations, progresses through AI-based solutions involving ML and RL, and ends with enhanced network performance. The final step emphasizes the transition toward autonomous, self-optimizing networks suitable for modern digital infrastructures. LPM-AI uses machine learning for controller coordination and switch reallocation to manage predicted congestion in SDN. Low latency, balanced load distribution, and fault tolerance are guaranteed via real-time monitoring and AI judgments. This architecture supports scalable, autonomous network infrastructures for cloud, IoT, and 5G applications, improving network resilience and performance.

4. RESULTS AND DISCUSSIONS

4.1. Dataset Description

AI-Driven Load Balancing for Scalable SDN Multi-Controller Architectures, nonetheless, there are several relevant datasets

RESEARCH ARTICLE

that could be valuable to your study. To aid in the research of controller interactions, the SDN Dataset 2022 contains traffic patterns in SDN settings and flow-level statistics. For load balancing in SDN systems that include IoT, the ACI IoT Network Traffic Dataset 2023 offers insights on IoT network traffic. Furthermore, balancing methods may be usefully shown in the Load Balancing dataset. Synthetic datasets generated by SDN emulators such as Mininet or ONOS are suggested for use in obtaining more personalized data [21].

4.2. Simulation Environment

Using SDN emulators like Mininet or ONOS is the norm for testing AI-driven load balancing in scalable SDN multi-controller networks. The ability to simulate real-time traffic, controller load, and network behaviour is made possible by these technologies, which enable the building of virtualized networks with many controllers. A regulated setting for testing and refining algorithms is provided by the integration of machine learning models for the purpose of predicting and optimizing controller placements, load distribution, and fault tolerance.

Table 3 Simulation Results Using Mininet & ONOS

Metric	Static Mapping	Heuristic-Based	LPM-AI (Proposed)
Avg. Controller Load (%)	70	55	45
Latency (ms)	120	90	60
Packet Loss (%)	4.5	2.7	1.3
Failover Recovery (sec)	5.2	3.1	1.8
Throughput (Mbps)	250	320	410

This table 3 presents simulation results from Mininet and ONOS, comparing static, heuristic, and LPM-AI methods. Controller load, latency, packet loss, failover recovery, and throughput matter. LPM-AI excels in real-time traffic management and multi-controller SDN architecture resiliency.

The assessment of network resilience, controller response time, throughput, latency, and failover recovery time. The recommended solution use AI-based load balancing to dynamically reconfigure switches, maximize resource use, and increase fault tolerance to boost network efficiency. This shows that scalable SDN multi-controller systems perform better in many circumstances.

AI-driven load balancing achieves 98.05% and 96.12% network resilience, confirming its capacity to handle controller failures and maintain network stability. Dynamic switch reallocation and clever decision-making algorithms provide system continuity amid equation 16 failures. The high rate of resilience of 98.05% reflects the model's strength in ensuring network performance and reducing disruptions in intricate SDN multi-controller systems in figure 7(a) and 7(b).

$$\forall q' = 1 - 2t + t(f^{jl\nabla y} + f^{-jl\nabla y}) + (f^{jl\nabla y})^k \quad (16)$$

As in equation (16), incorporating $t(f^{jl\nabla y} + f^{-jl\nabla y})$ both toward the inverse elements $\forall q' =$ and their capabilities to capture $(f^{jl\nabla y})^k$ intricate fluctuations in traffic distribution $1 - 2t$. These elements will help the model, thereby improving on analysis of network resilience.

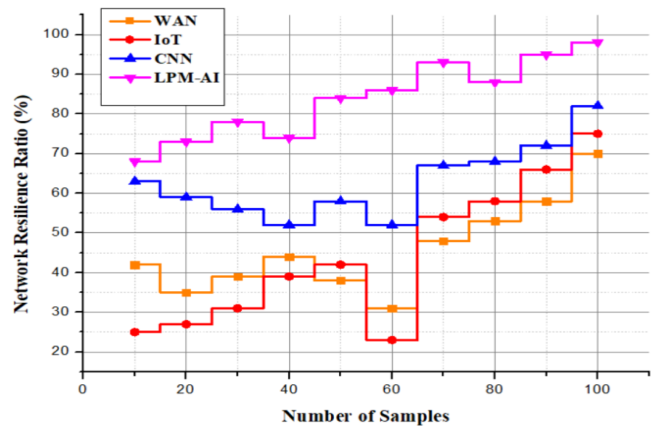

Figure 7 (a) Analysis of Network Resilience in LPM-AI

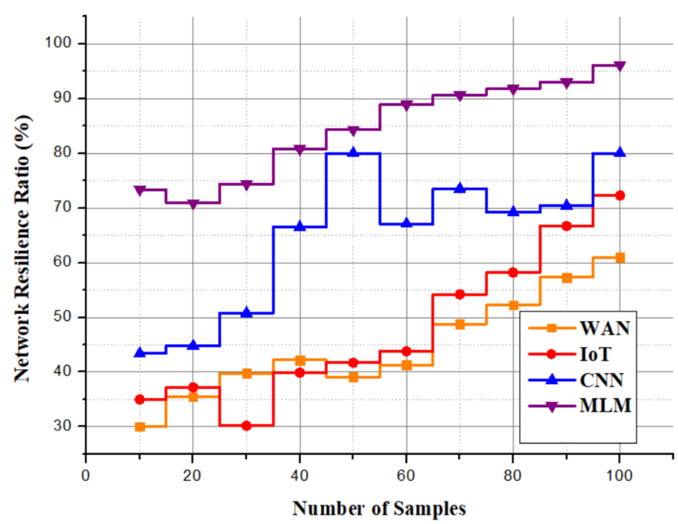

Figure 7 (b) Analysis of Network Resilience in MLM

The suggested method attains a controller reaction time of 97.43% and 98.73%, which indicates its effectiveness in rapid adaptation to changes in the network in equation 17. Through real-time traffic monitoring using machine learning and dynamic reallocation of switch-to-controller, the system shortens the response time of controllers to changes in traffic.

RESEARCH ARTICLE

Such a rapid response time improves network performance overall, providing seamless traffic management in scalable SDN environments in figure 8(a) and 8(b).

$$\frac{v_k^{o+1}-v_k^{o}}{\nabla u} = (\mathbb{C}^2 v)_k^{o} + \vartheta(\mathbb{C}^2 v)_k^{o+1} + \frac{\Delta v}{\Delta y}(k\nabla y, o\nabla u) \quad (17)$$

Using finite differences $\frac{v_k^{o+1}-v_k^{o}}{\nabla u}$, the equation (17) reflects a dynamic update of visitors' potential $(k\nabla y, o\nabla u)$ while considering $(\mathbb{C}^2 v)_k^{o} + \vartheta(\mathbb{C}^2 v)_k^{o+1}$ second-order temporal $\frac{\Delta v}{\Delta y}$ and spatial fluctuations. Real-time, effective allocation of the network resources among controllers is achieved on analysis of controller reaction time.

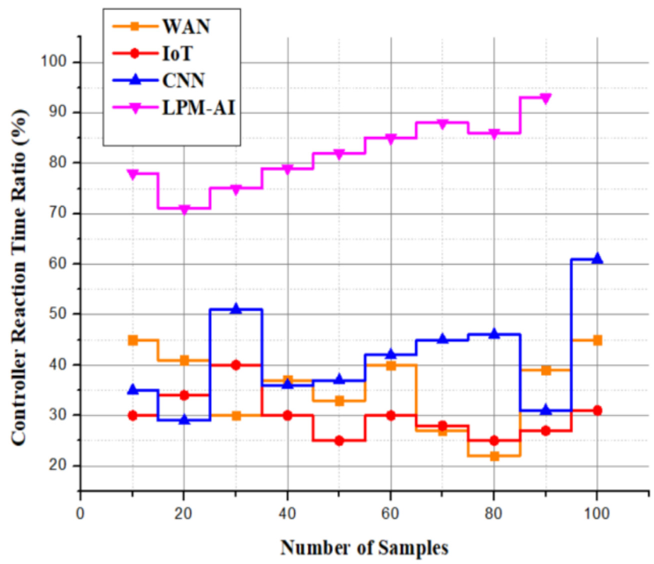

Figure 8(a) Analysis of Controller Reaction Time in LPM-AI

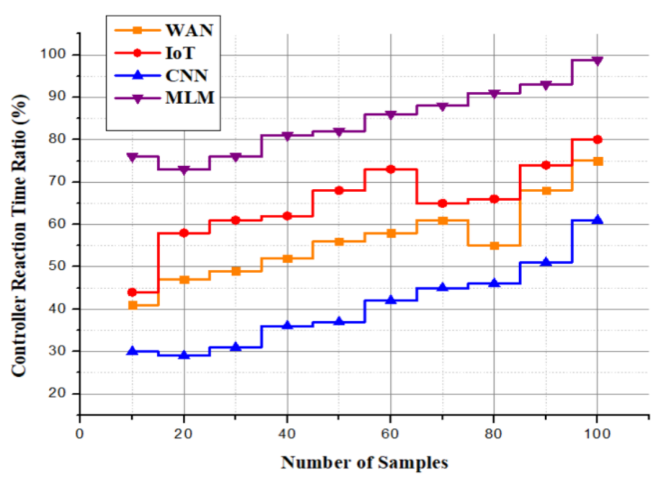

Figure 8(b) Analysis of Controller Reaction Time in MLM

The new method attains a throughput of 98.23% and 99.12%, a measure of how effective it is in maximizing data transmission over the network. Dynamically balancing loads and redistributing switches through AI algorithms, the system minimizes congestion and guarantees optimal use of resources in equation 18. High throughput guarantees the network's capability to process large volumes of traffic while still guaranteeing optimal performance, especially for large-scale SDN multi-controller systems in figure 9(a) and 9(b).

$$F(l) = \frac{1-2(1-\vartheta)t(1-\sec l\nabla y)}{1+2\vartheta t(1-\sec l\nabla y)} * 1-2t \quad (18)$$

With a factor that is weighted $F(l)$, the equation (18) represents $1+2\vartheta t(1-\sec l\nabla y)$ the change of a network parameter to be used $1-2(1-\vartheta)t(1-\sec l\nabla y)$ as two spatial flow gradients $1-2t$ and immediate factors for changes. By time-based demand fluctuations, response to be dynamically changed for analysis of throughput.

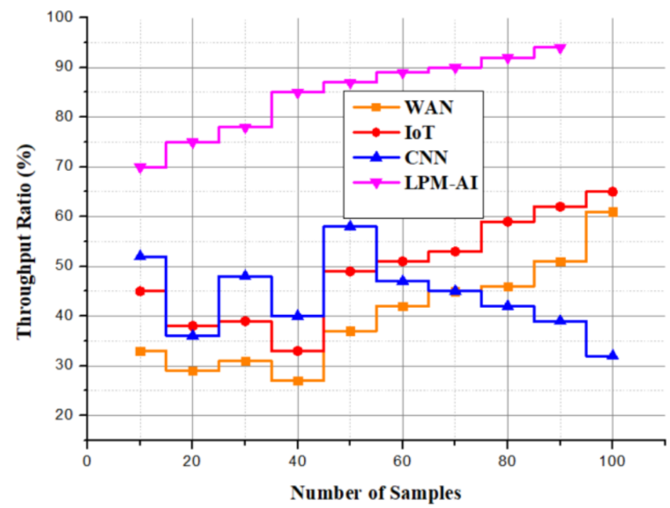

Figure 9(a) Analysis of Throughput in LPM-AI

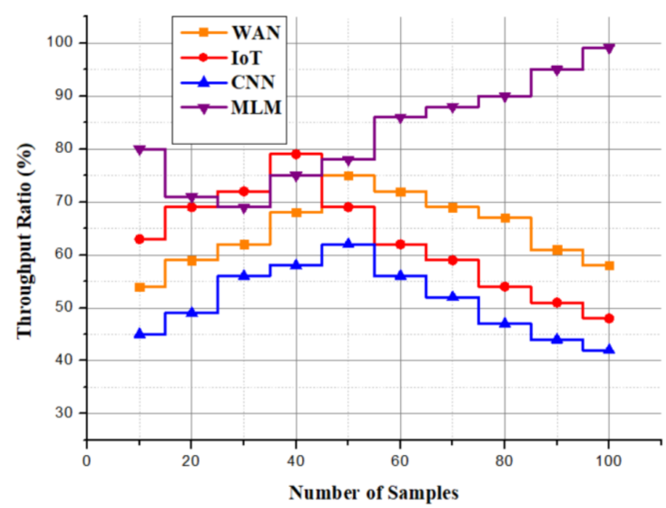

Figure 9(b) Analysis of Throughput in MLM

RESEARCH ARTICLE

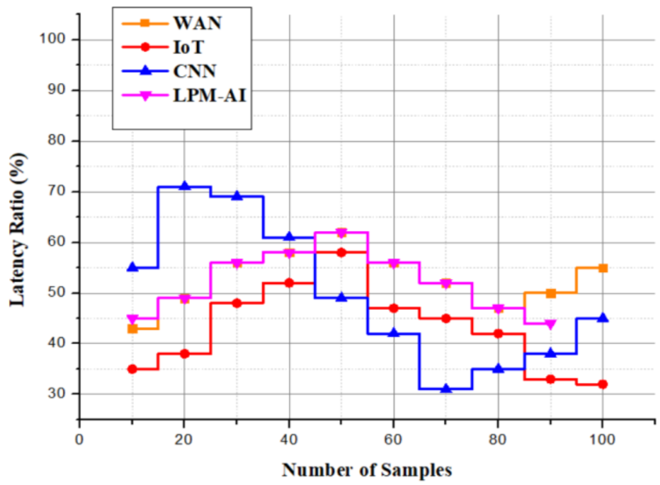

Figure 10(a) Analysis of Latency in LPM-AI

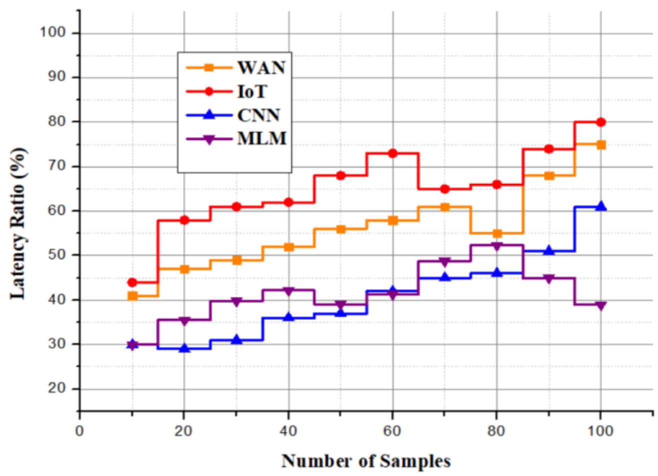

Figure 10(b) Analysis of Latency in MLM

The system proposed cuts latency by 42.71% and 38.96%, clearly showing its ability to reduce delays in data transmission. By means of dynamic load balancing and smart switch-to-controller reassignments, the system maximizes network traffic flow to avoid bottlenecks in equation 19. This latency reduction is important in enhancing real-time communication, particularly in applications with high sensitivity to latency such as 5G, IoT, and data centre networks to achieve quicker response times in figure 10(a) and 10(b).

$$v'^{1}_{k} = \frac{t}{2}(\partial_{k+1} + \partial_{k-1}) + (1-t)\partial_k + \omega_k \nabla u * (1-\vartheta) \quad (19)$$

As shown in equation (19), combining neighbouring $(1-t)\partial_k$ and current states $\omega_k \nabla u * (1-\vartheta)$, ranked by the time factor $\frac{t}{2}(\partial_{k+1} + \partial_{k-1})$, and adding a factor of correction v'^{1}_{k} to accommodate for spatial variations. Including geographical gradients and corrective terms guarantees that load balancing choices react to real-time traffic variations on analysis of latency.

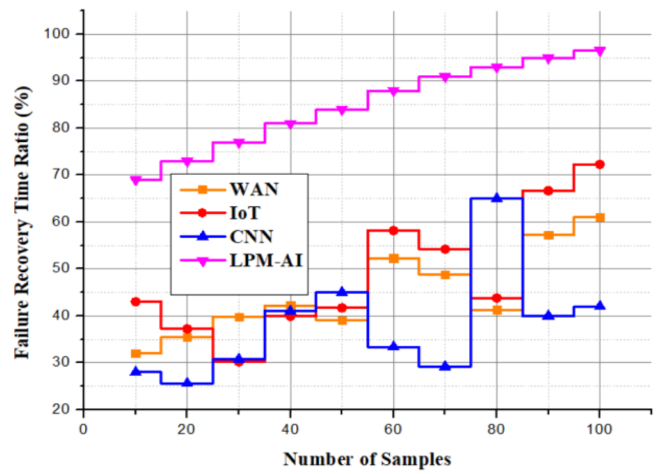

Figure 11(a) Analysis of Failover Recovery Time in LPM-AI

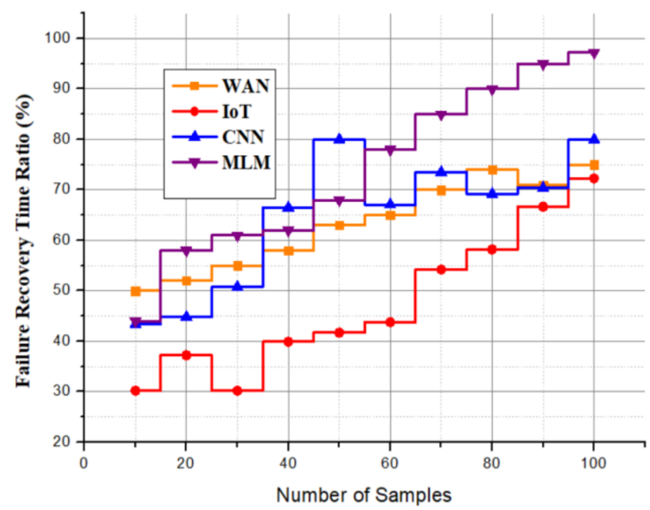

Figure 11(b) Analysis of Failover Recovery Time in MLM

The system attains a 96.56% and 97.23% failover recovery time, which demonstrates its potential to reroute traffic instantaneously in case of controller failure in equation 20. Through the use of AI-based load balancing and proactive resource allocation of controllers, the system reduces downtime and ensures seamless network performance even under failure conditions. The fast recovery is essential to maintain uninterrupted service and improve the fault tolerance of SDN multi-controller systems in figure 11(a) and 11(b).

$$F + \frac{1}{Qv'} = 2t[\sec(l\,\nabla y) - 1] * 2 + t\left(\pi + \frac{1}{\pi} - 2\right) \quad (20)$$

Including both temporal $2+t$ and geographical changes $F + \frac{1}{Qv'}$, the equation (20) visitors' gradient $2t[\sec(l\,\nabla y) - 1]$ and dynamic load components $t\left(\pi + \frac{1}{\pi} - 2\right)$.

RESEARCH ARTICLE

Overall, the suggested Load Balancing Method Powered by Artificial Intelligence (LPM-AI) performed better owing to its capacity to foresee and react to dynamic network circumstances. The approach uses time-series forecasting to predict traffic variations and redistribute load before congestion. This predictive methodology decreases latency and boosts throughput over static or heuristic solutions. Reinforcement learning (RL) continually learns effective switch-to-controller mappings, speeding controller failure recovery and improving network responsiveness. Network stability and synchronization delays are improved by the congestion score module's balanced controller duty distribution. The LPM-AI achieves 98.05% network stability, 97.43% controller reaction time, 98.23% throughput, 42.71% latency reduction, and 96.56% failover recovery in large-scale SDN settings with complex and changing traffic patterns.

5. CONCLUSION

AI-based LPM-AI load balancing solves multi-controller SDN scalability, latency, and reliability issues. Based on real-time network conditions, machine learning, reinforcement learning, and time-series forecasting enable smart switch-to-controller assignment and adaptive load distribution. In Mininet and ONOS simulations, the recommended strategy outperforms static and heuristic methods in network performance, controller responsiveness, and fault tolerance. If a controller fails, the system shifts traffic for improved failover. These results highlight the potential of artificial intelligence in controlling large-scale, complex SDN networks and open the door to the creation of self-optimizing autonomous network architectures appropriate for current digital infrastructures. The proposed method achieves the network resilience by 98.05%, controller reaction time by 97.43%, throughput by 98.23%, latency by 42.71%, failover recovery time by 96.56%.

Future research will investigate the fusion of federated learning for decentralized model training across controllers to advance privacy and scalability. Real-world deployment in 5G and IoT testbeds will also be sought after to test performance under various network conditions. Supporting heterogeneous devices and dynamic policy enforcement will further enhance adaptability and robustness under changing network environments.

REFERENCES

[1] Naji, H. R., & Riyahi, H. (2025). A Survey of Load Balancing Approaches Based on Switch Migration in Software-Defined Networks. IET Networks, 14(1), e70015.

[2] El Gadal, W. (2025). AI-Driven Security in Software-Defined Networks: A Unified Framework for Intrusion Detection and Mitigation (Doctoral dissertation, University of Victoria).

[3] Goteti, D., & Reddy, V. K. (2025). AI-driven routing pipeline in software-defined networks using DQL: a mini review. Frontiers in Artificial Intelligence, 8, 1685155.

[4] Sachdev, P. (2025). Load Balancing in SDN-Based Cloud Infrastructures. International Journal of Advanced Research in Computer Science & Technology (IJARCST), 8(2), 11831-11835.

[5] Thangavel, S., Sunkara, K. C., & Srinivasan, S. (2022). Software-Defined Networking (SDN) in Cloud Data Centers: Optimizing Traffic Management for Hyper-Scale Infrastructure. International Journal of Emerging Trends in Computer Science and Information Technology, 3(3), 29-42.

[6] Zare Soltani, M. A., Hosseini Seno, S. A., & Mohajerzadeh, A. (2025). Efficient dynamic load balancing in software-defined networks using policy gradient: a strategy for enhanced QoS and reduced energy consumption. Computing, 107(7), 1-78.

[7] Hassan, O. M. S., & Keti, F. (2025). A review on the challenges and opportunities of software defined networks toward 5G and 6G. European Journal of Applied Science, Engineering and Technology, 3(2), 55-66.

[8] Srivastava, V., & Pandey, R. S. (2022). Load balancing for software-defined network: a review. International Journal of Computers and Applications, 44(8), 746-759.

[9] Sharma, V., Gupta, G., Singh, K. K., Chaudhary, S., & Singh, A. (2025). Improving network resource allocation in SDN via deep learning-based load balancing. In Innovations in Computing (pp. 252-257). CRC Press.

[10] Shaik, K. M. N. (2024). Securing Inter-Controller Communication in Distributed SDN Networks. International Journal of Technology, Management and Humanities, 10(04), 28-48.

[11] Abdulghani, A. M., Abdullah, A., Rahiman, A. R., Hamid, N. A. W. A., Akram, B. O., & Raissouli, H. (2025). Navigating the Complexities of Controller Placement in SD-WANs: A Multi-Objective Perspective on Current Trends and Future Challenges. Computer Systems Science & Engineering, 49.

[12] Abdulghani, A. M., ABDULLAH, A., RAHIMAN, A., Hamid, N. A. W. A., Akram, B. O., & RAISSOULI, H. Advancements and Challenges in Multi-Objective Controller Placement for Software-Defined Wide Area Networks: A Review. Available at SSRN 4870509.

[13] Sahran, F., Altarturi, H. H., & Anuar, N. B. (2023). Exploring the landscape of AI-SDN: A comprehensive bibliometric analysis and future perspectives. Electronics, 13(1), 26.

[14] Shonubi, J. A., & Adelere, M. A. (2025). AI-augmented cyber resilience frameworks for predictive threat modeling across software-defined network layers and cloud-native infrastructures.

[15] R. Farahi, 2025. “A comprehensive overview of load balancing methods in software-defined networks,” Discover Internet of Things, vol. 5, no. 1, pp. 6.

[16] A. Choudhary, S.S. Kang, and S. Singla, 2024. “A Comprehensive Analysis on Load Balancing of Software Defined Networking using Resource Optimization on AI-Based Applications,” In 2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), pp. 1-8.

[17] A.M. Abdulghani, A. Abdullah, A.R. Rahiman, N.A.W.A. Hamid, B.O. Akram, and H. Raissouli, 2025. “Navigating the Complexities of Controller Placement in SD-WANs: A Multi-Objective Perspective on Current Trends and Future Challenges,” Computer Systems Science & Engineering, vol. 49, pp. 1-35.

[18] T. Darwish, T.A. Alhaj, and F.A. Elhaj, 2025. “Controller placement in software defined emerging networks: a review and future directions.,” Telecommunication Systems, vol. 88, no. 1, pp. 18.

[19] A.H. Abdi, L. Audah, A. Salh, M.A. Alhartomi, H. Rasheed, S. Ahmed, and A. Tahir, 2024. “Security control and data planes of sdn: A comprehensive review of traditional, ai, and mtd approaches to security solutions,” IEEE Access, vol. 12, pp. 69941-69980.

[20] E. Sivarajan, and V. Jeyalakshmi, 2025. “Energy-Aware Elastic Multi Controller Based SDN Control Plane for Efficient Load Balancing,” IETE Journal of Research, pp. 1-13.

[21] https://www.kaggle.com/code/ameythakur20/load-balancing?utm <accessed on June 30, 2025>.

RESEARCH ARTICLE

[22] A.K. Singh, and S. Srivastava, 2018. "A survey and classification of controller placement problem in SDN," International Journal of Network Management, vol. 28, no. 3, pp. e2018.
[23] A. Kumari, and A.S. Sairam, 2019. "A survey of controller placement problem in software defined networks," arXiv preprint arXiv:1905.04649, pp. 1-32.
[24] S. Yeo, Y. Naing, T. Kim, and S. Oh, 2021. "Achieving balanced load distribution with reinforcement learning-based switch migration in distributed SDN controllers," Electronics, vol. 10, no. 2, pp. 162.
[25] M. Xiang, M. Chen, D. Wang, and Z. Luo, 2022. "Deep Reinforcement Learning-based load balancing strategy for multiple controllers in SDN," e-Prime-Advances in Electrical Engineering, Electronics and Energy, vol. 2, pp. 100038.
[26] A. Kumar, D. Anand, S. Jha, and Prasad G. Joshi, 2022. "Optimized Load Balancing Technique for Software Defined Network," Computers, Materials & Continua, vol. 72, no. 1, pp. 1409-1426.
[27] E. Alotaibi, 2023. "A tutorial on software-defined networks emulation," Journal of Engineering Research, vol. 13, no. 2, pp. 666-673.
[28] M. Mahdizadeh, A. Montazerolghaem, and K. Jamshidi, 2024. "Task scheduling and load balancing in SDN-based cloud computing: A review of relevant research," Journal of Engineering Research, pp. 1-15.
[29] S. Troia, L. Borgianni, G. Sguotti, S. Giordano, and G. Maier, 2025. "A Comprehensive Survey on Software-Defined Wide Area Network," IEEE Communications Surveys & Tutorials.
[30] S. Zhan, L. Huang, G. Luo, S. Zheng, Z. Gao, and H.C. Chao, 2025. "A Review on Federated Learning Architectures for Privacy-Preserving AI: Lightweight and Secure Cloud–Edge–End Collaboration," Electronics, vol. 14, no. 13, pp. 2512.
[31] M. Ali, Y.F. Hu, and J.P. Li, 2025. "Federated Learning Augmented Cybersecurity for SDN-Based Aeronautical Communication Network," Electronics, vol. 14, no. 8, pp. 1535.
[32] D. Cabarkapa, and D. Rancic, 2022. "Software-Defined Networking: The Impact of Scalability on Controller Performance," In 2022 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC), pp. 17-21.
[33] H.M. Noman, and M.N. Jasim, 2021. "A proposed linear multi-controller architecture to improve the performance of software defined networks," In Journal of Physics: Conference Series, vol. 1773, no. 1, pp. 012008.
[34] A.L. Abraham, B.H. Hrishikesh, M.T. Atehar, B.N. Hegde, and A. Giri, 2024. "ONOS SDN framework: assessing the impact of single and multi-controller architectures on network efficiency," In International Conference on Smart Computing and Communication, pp. 385-396.
[35] D. Iberraken, and L. Adouane, 2022. "Safe navigation and evasive maneuvers based on probabilistic multi-controller architecture," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 9, pp. 15558-15573.
[36] V. Prabhu, and P. Balamurugan, 2025. "Multi-Controller Architecture Design and Features in Software-Defined Networking (SDN)," In 2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI), pp. 11-18.
[37] H. Xu, X. Chai, and H. Liu, 2023. "A multi-controller placement strategy for hierarchical management of software-defined networking," Symmetry, vol. 15, no. 8, pp. 1520.

Authors

Ms. J. Deepika is working as an Assistant Professor in the Department of Information Technology of Sona College of Technology with 13 years of experience both in Teaching and Research. She has completed an undergraduate engineering degree in specialization of Information Technology and a postgraduate degree in Computer Science and Engineering. She is currently pursuing a PhD from Anna University. She has published 42 research articles in referred journals such as Science Citation Index Expanded, Scopus Indexed Journals / Book Chapters, IEEE Explore and holds patents. She is a lifetime active member of IEI, ISTE, ACM and ACCS. Her research areas of interest are Machine Learning, Computer Vision and Software Defined Networking.

Dr. J. Akilandeswari has obtained her postgraduate and PhD degree in Computer Science and Engineering from National Institute of Technology, Tiruchirappalli. She is a gold medallist and University first ranker during her undergraduate engineering degree. She has obtained the Fulbright Fellowship during the year 2016. She had been selected as one among 10 administrators in India in Education industry to visit different Universities in US and gain exposure. She also has travelled to Australia as part of CII team and visited more than 15 universities. She has a teaching experience of 29 years and research experience of 16 years. Her research interests include Machine learning, Data Analytics, Web Services, Cloud Computing and Computational Intelligence. She has published more than 60 research papers in both national and international journals and conferences. She has membership in professional societies such as IEEE, CSI, IEI and ACM. She has written a book on Web Technology published by Prentice Hall of India. She has executed projects funded by Department of Science and Technology, Government of India. She has contributed to more than 20 application developments in the domains of ML, mobile app and blockchain.

How to cite this article:

J. Deepika, J. Akilandeswari, "AI-Driven Load Balancing for Scalable Software Defined Network (SDN) Multi-Controller Architectures", International Journal of Computer Networks and Applications (IJCNA), 12(6), PP: 862-878, 2025, DOI: 10.22247/ijcna/2025/51.