Original Article

Intelligent Optimization of OSPF Path Selection Using Machine Learning Models for Adaptive Network Routing

Rebeen Rebwar Hama Amin [a] *

[a] Network Department, Computer Science Institute, Sulaimani Polytechnic University, Sulaymaniyah, Iraq.

Submitted: 25 May 2025
Revised: 1 July 2025
Accepted: 4 August 2025

* **Corresponding Author**:
rebeen.rebwar@spu.edu.iq

Keywords: OSPF, SDN, Machine learning, Routing protocols.

How to cite this paper: R. R. Hama Amin, "Intelligent Optimization of OSPF Path Selection Using Machine Learning Models for Adaptive Network Routing", KJAR, vol. 10, no. 2, pp. 31-42, Dec 2025, doi: 10.24017/science.2025.2.3

Abstract: At the core of enterprise networks lies routing protocols that make forwarding decisions based on a set of rules and metrics. One of the most popular and widely used routing protocols is the Open Shortest Path First (OSPF). Traditional OSPF calculates the cost of the route primarily based on interface bandwidth, without considering real-time factors such as latency, congestion, or link stability. These calculations are static and can lead to deficiencies in adapting to unstable network conditions. This study proposes the integration of multiple machine learning (ML) models and techniques to enhance OSPF routing decisions. Four important ML functions namely traffic forecast, anomaly detection, failure prediction, and dynamic cost optimization—have been used to improve OSPF performance. ML methods such as Random Forest and XGBoost are used to predict and assign costs in traffic utilization and real-time performance assessments. AutoRegressive Integrated Moving Average models and Long Short-Term Memory are applied to enable traffic predictions and route adjustments before potential congestions. Furthermore, link and node failure are common in network routing. Random Forest and logistic regression models are employed to predict these. The simulation took place in Graphical Network Simulator-3 using Cisco routers and Linux servers to allow thorough testing before and after applying the ML models. The results and findings have shown that the integration of ML models reroutes the traffic to enhance latency and throughput by approximately 30%. The findings demonstrate the upside of ML-enhanced OSPF routing as a versatile and scalable solution for high-demand networks.

1. Introduction

The evolution of enterprise-class networks has brought more challenges due to the complexities and performance requirements, making it vital to provide real-time responsiveness and fault tolerance in such environments. To operate such networks, routing protocols such as Open Shortest Path First (OSPF) will be used to provide routing services and efficiently make routing decisions. OSPF is a link-state protocol that mostly relies on bandwidth as the primary cost metric to select a forwarding path [1-4]. Optimizing OSPF path selection can, however, cause issues due to traditional OSPF metric baselines, which are limited [5]. Several routing protocols, including OSPF, use algorithms such as Dijkstra's algorithm to find the best path to the destination. However, even though OSPF is successful in the static context of metrics and path selections, it cannot dynamically adjust parameters to provide improvements in latency, link failure, and congestion [2]. Therefore, routing optimization is a crucial study domain to enhance the overall performance of network routing [4-7].

Software-defined networking (SDN) is one of the recent network advancements that separates control and management planes to enable intelligent deployment through application programming interfaces (APIs). While traditional networks struggle with static sets of metrics, SDN is an effective way to apply new, adapted programming techniques to enhance performance. SDN uses network automation tools to orchestrate network operations to provide smart, programmable, and automated architectures [4, 5]. Ansible is a popular automation and configuration management tool that can apply SDN tasks through APIs. Using these programmability advancements enables the application of machine learning (ML) models for traffic predictions and management [8-11].

To overcome the challenges of static metric routing, this research aims to integrate an ML model into OSPF routing decisions, proposing that predicting traffic and adapting to environmental changes will enhance the overall performance of the network. Real-time networks can reach peak performance if they overcome the challenge caused by latency, packet loss, and congestion. Thus, training ML models to predict and influence OSPF routing decisions can be crucial to provide better performance. This study focuses on four main goals for the ML models to train upon [12]. To optimize cost dynamically, ML models are being used, and the most common techniques are Random Forest and XGBoost regression models. The setup system will analyze real-time collected performance data to assign OSPF cost dynamically instead of using traditional static metrics. Hence, accurate reflections of the network state will lead to better path calculations by the routing protocol [13]. Then, the collected traffic data will be used by time-series models such as Long Short-Term Memory (LSTM) and AutoRegressive Integrated Moving Average (ARIMA) to train and predict patterns. These data will eventually forecast potential future congestions that could impact the overall network performance [10, 11].

Furthermore, routing protocols such as OSPF are well-known for link flapping and failures in large-scale congested networks. Therefore, identifying these issues is crucial to the network performance [14-16]. ML unsupervised methods such as Isolation Forest and Autoencoders can be used to identify anomalies in routed traffic flows, and these models will flag any possible real-time problems by finding anomalies such as packet loss, jitter, or latency variations. Finally, Random Forest and logistic regression classifiers will be employed to forecast possible link or node failures depending on patterns in collected network behavior data and performance degradation indicators. The study also investigated the resilience and reliability of the network in order to improve overall performance.

Using Graphical Network Simulator-3 (GNS3), the experimental environment simulation is set to include a network of 10 Cisco routers linked together to create a feasible topology. End-to-end tests across all router pairs (e.g., from R1 to R10) captured crucial data including latency, real OSPF cost, and link usage. Enhancing resource allocation and routing to meet service requirements is essential in extensive networks. A structured simulation will examine each of these ML-driven methods, with thorough test cases logged in a centralized table. With ML-predicted OSPF costs closely matching real-world behavior and consistently overcoming static OSPF metrics in dynamically changing environments, the findings will reveal significant improvements in routing performance [10, 11]. By enabling the system to reroute traffic before performance declines, traffic forecasting models correctly predict congestion patterns. Anomaly detection and failure prediction also give warnings promptly of possible problems, thereby raising the fault tolerance of the network.

This work proposes the integration of ML models in OSPF path selection, enhancing the decision-making of routing based on real network status. It designs and evaluates a multi-model approach to optimize cost functions, predict traffic patterns, detect anomalies, and predict failures. The contribution also includes developing and simulating a testbed constructed on top of GNS3 with automated data collection and dynamic configuration using Python and Ansible, with measurable enhancements in latency, throughput, and path reliability.

The layout of this paper is represented as follows. Section 2 reviews the existing contemporary related works. Section 3 outlines the methodology employed in applying the ML models within the traditional network structure. Section 4 presents the results of the simulations, followed by a discussion in section 5 that describes the findings. Finally, section 6 concludes the paper.

2. Related Works

As network scale grows, the allocation of network resources has become increasingly critical. The close interconnection between the control plane and data plane in conventional network design prevents routing algorithms from acquiring network status information from a comprehensive viewpoint and subsequently planning forwarding routes accordingly. This may lead to diminished network utilization and significant congestion in extensive scenarios. SDN presents an innovative methodology that employs a programmable control plane to dictate the forwarding of various data flows. The network's control logic is then conveyed from delivery devices, such as routers and switches, to software controllers, effectively separating the data plane from the control plane and streamlining network management [15]. Wei *et al.* [17] propose HEATE, an algorithm combining traditional OSPF optimization with SDN flow adjustments to enhance energy efficiency. Their model dynamically reallocates traffic across links to reduce power consumption while maintaining network performance. Furthermore, an ML-driven SDN network might overcome greater challenges faced in time-sensitive massive network topologies.

Caria and Jukan [18] investigate how to configure link weights in hybrid SDN/OSPF networks to support fault-tolerant operations. Their work proposes mathematical models to balance capacity planning, reduce overprovisioning, and ensure service continuity under failure conditions. Caria *et al.* [19] introduce a hybrid networking strategy where SDN controllers partition OSPF domains into multiple sub-domains. This setup enables centralized fine-grained control of routing paths while preserving OSPF’s decentralized nature, improving routing flexibility and manageability. Networking trends suggest using ML for many network optimization tasks. Extensive efforts have been made to develop ML-based solutions for traffic engineering (TE), a major issue in internet service provider networks. Modern TE optimizers use local search, constraint programming, and linear programming. Multi-Agent Reinforcement Learning and Graph Neural Networks for Distributed TE Optimization (MAGNNETO), is a distributed ML-based system for distributed TE optimization, uses multi-agent reinforcement learning and graph neural networks (GNN). MAGNNETO distributes agents across the network that learn and communicate via message exchanges. This framework optimizes link weights in OSPF to reduce network congestion. MAGNNETO is evaluated against three leading TE optimizers in over 75 topologies with actual traffic volumes and experiments reveal that MAGNNETO executes faster than state-of-the-art TE optimizers due to its distributed nature. The ML-based approach also generalizes well to novel networks uncovered during training [1].

A significant trajectory has emerged in the utilization of intelligence-based routing within programmable networks, especially over the past decade; nonetheless, considerable effort remains necessary to attain thorough comparisons and synergies of methodologies, as well as substantive evaluations grounded in available datasets and topologies, along with comprehensive practical implementations that might be used by industry [20]. The articles addressed examine the application of ML approaches for routing optimization in software-defined networking, categorized into three primary types: supervised learning, unsupervised learning, and reinforcement learning. This survey provides comprehensive summary tables pertaining to these investigations, and a comparative analysis is also addressed, giving an overview of the most exemplary works based on their evaluation [20].

Chen *et al.* [15] formulated the optimization target for load balancing within the network and presented an enhanced population initialization approach that utilizes explicit data regarding network environmental variables to expedite algorithm convergence. Furthermore, an exploration phase was implemented to augment the algorithm's development and enhance its efficacy. Ultimately, they conducted simulation experiments utilizing three network topologies and two traffic intensities, evaluating their suggested model in comparison to OSPF and the original algorithms. The findings and results highlight that better adaptability and metric values can be gained using the proposed model. Therefore, the application of the model enhanced the overall load-balancing efficiency.

Deep learning is an effective way of using ML for enhancing enterprise network performance. Abrol *et al.* [21] conducted research on the use of deep reinforcement learning (DRL) methods for adaptive routing. They successfully developed a Deep Graph Convolutional Neural Network for the DRL framework to analyze traffic behaviors and link failures. The proposed approach utilizes q-value estimates for path selection for each traffic flow request. Various experiments were conducted on the traffic patterns while using the deep learning approach and then compared to the traditional OSPF methods. The experimental results have shown that the proposed framework provides better latency, throughput, and adaptability compared to the conventional routing protocol.

Wireless mesh networks are popular for their low cost, quick transmission time, simple setup, and flexibility. Though older routers may follow set rules even when superior routes are available, routing algorithms are vital in defining data paths between nodes. This study suggests adopting the QL-Feed Forward Routing (QFFR) algorithm to solve this problem; it uses Q-learning and a feed forward neural network to allow smart, adaptive routing choices. By means of the network environment, QFFR chooses the most efficient path, thereby enhancing routing performance. Using measures such as throughput, packet delivery ratio, and delay, this contrasts QFFR with conventional algorithms. Results indicate that the proposed algorithm is best in delay and throughput [22]. Pan *et al.* [23] presented a novel method using inverse coupled simulated annealing to optimize OSPF route convergence in IoT networks and addressed issues like slow fault recovery and dynamic topology instability by enhancing convergence speed and reliability through metaheuristic optimization.

Enabling smarter, more autonomous, and user-centric systems, ML is poised to play a key role in 5G self-organizing networks (SONs). The User Specific-Optimal Capacity Shortest Path (US-OCSP) routing technique presented in this paper finds the optimal route between a source and destination using ML depending on node capacity and distance. Q-learning is used to prevent congestion and guarantee the best throughput by means of simulations evaluating available network resources. In 5G SON settings, the method improves user experience and supports efficient resource allocation [24]. Moreover, traditional management techniques fall behind the data volume and device diversity as 6G networks expand. The Speed-optimized LSTM (SP-LSTM) model, which combines predictive analytics with dynamic routing, is a new ML-driven solution offered in this paper [25]. Fast congestion prediction is provided by SP-LSTM, while adaptive routing is provided by reinforcement learning in the two-tiered system. This proactive, learning-based strategy satisfies 6G requirements for ultra-low latency, high dependability, and effective resource use. SP-LSTM's rapid training and prediction show the strength of ML to surpass conventional methods in next-gen network management, making it perfect for dynamic settings [25].

Çoğay *et al.* [26] proposed a QoS-aware dynamic routing framework for edge routers, using ML methods such as XGBoost and Random Forest to classify network packets and employing a multi-path routing approach to dynamically adapt to changing network conditions. The framework consisted of three main modules: a packet classifier, a load balancer, and a routing engine. The packet classifier classified internet packets using ML and the load balancer marked flows based on priority levels and analyzed link loads to prevent congestion. The routing engine selected the optimal paths based on these analyses, ensuring efficient data transfer with optimized routing cost. In the evaluation phase, the packet classifier used pre-trained ensemble learning methods. Based on the classification results, the introduced ML-based framework achieved better throughput than traditional OSPF.

Although earlier research has provided important contributions to the field by optimizing OSPF and general network routing using deep reinforcement learning and graph-based models, this study stands out for its thorough integration of multiple ML models across other network functions. In contrast to methods that only adjust link weights or that only use reinforcement learning frameworks, this research addresses dynamic cost optimization, traffic prediction, anomaly detection, and failure forecasting at the same time by combining supervised learning (XGBoost, Random Forest), time-series forecasting (LSTM, ARIMA), and unsupervised learning (Isolation Forest, Autoencoders). Additionally, a more grounded, testbed-driven evaluation is made possible by the use of a realistic GNS3-based simulation with Cisco routers and live metric collection using Simple Network Management Protocol (SNMP), NetFlow, Internet Control Message Protocol (ICMP), and syslog. A multilayered improvement to OSPF routing is offered by this comprehensive and modular approach, which permits both proactive and reactive decision-making in actual network scenarios that have not received as much attention in the previous literature.

3. Materials and Methods

A thorough simulation-based approach is created to assess how efficiently ML is incorporated into OSPF routing optimization. Using GNS3, an experimental environment is built with a network of 10 Cisco routers spread out in a single topology to mimic a realistic enterprise-level infrastructure. Network parameters, including latency, bandwidth, and traffic load, can be controlled by this configuration. The approach is to split into four stages: data gathering, model training, real-time ML integration, and performance assessment. From several end-to-end router paths (e.g., R1 to R10) under both normal and overwhelmed circumstances, latency, OSPF-calculated cost, traffic volume, and link status are logged during the data collection phase. These datasets were then used to train and test a suite of ML models customized to particular routing improvement tasks: regression models for cost prediction, time-series models for traffic forecasting, unsupervised models for anomaly detection, and classification models for failure prediction. The trained models were then included into a dynamic routing decision layer that interacted with OSPF's cost assignment system.

Figure 1 is a block diagram illustrating the end-to-end process of the proposed system for ML-based OSPF optimization. It begins with gathering data from network devices by utilizing probes to collect statistics. The data are preprocessed and fed into a ML processing module where ML models perform cost optimization, traffic forecasting, anomaly discovery, and failure prediction. The output from these models is utilized to trigger automated configuration changes to OSPF costs via Ansible or Python scripts. A feedback loop tracks the performance of the network after ML; whenever performance is lower than expected, the system initiates model retraining to ensure constant improvement and flexibility.

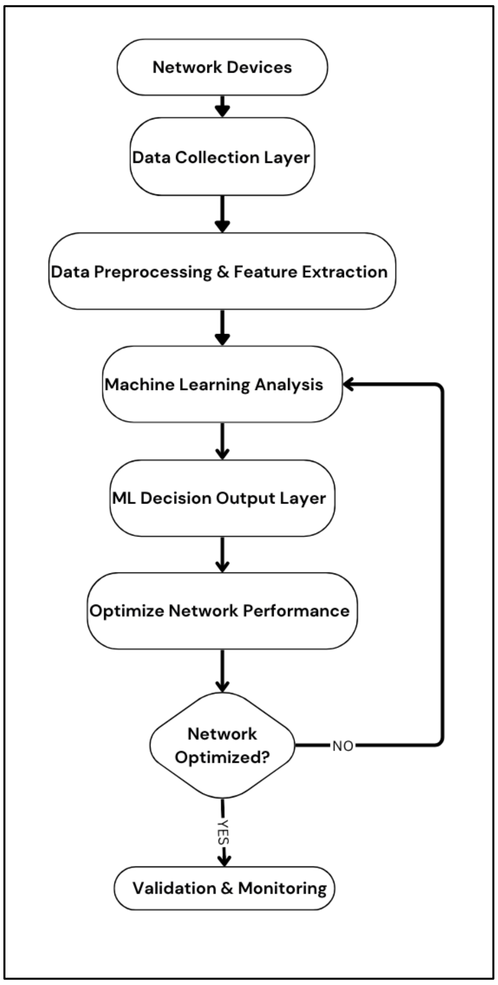

Figure 1: Workflow of ML-driven OSPF path optimization.

The performance of the improved system was compared to baseline OSPF behavior using key indicators such as latency, packet loss, route stability, and failure response time. Figure 2 shows the

GNS3-based testbed topology built of 10 linked Cisco routers set up using OSPF. Two Linux servers and two virtual PCs, one at each end, were used to run real-time statistics gathered by SNMP, NetFlow, and ICMP to dynamically influence traffic flows, path selections, and routing decisions. All simulation phases including baseline measurements, congestion injection, and ML-guided routing optimization rest on this topology.

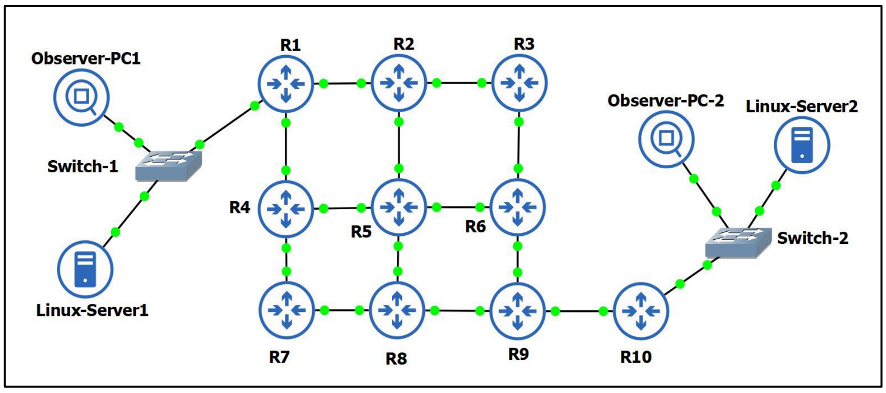

Figure 2: Testbed topology in GNS3 with 10 Cisco routers used for ML-enhanced OSPF simulation.

3.1. Simulation Setup and Measurement Plan

A thorough simulation in a controlled GNS3 environment with 10 linked Cisco routers (R1–R10) was run to assess the performance of ML-enhanced OSPF routing. With several iterations before and after ML integration, the testing approach comprises both baseline and stress conditions. Every stage of the experiment sought to measure variations in network performance and OSPF behavior. Designed to isolate the effect of ML on OSPF behavior, the simulation comprised five main phases as shown in table 1. Under typical load, the native OSPF route between R1 and R10 was seen in the baseline phase. Phase 2 saw the injection of bandwidth-hungry traffic to purposefully overload certain links. Without ML intervention, phase 3 re-evaluated performance under congestion. ML models in phase 4 dynamically predicted and updated OSPF costs depending on real-time latency and interface load data. Phase 5 ran the tests again to measure changes.

Table 1: Simulation phases and objectives.

Phase	Action	Objective
1	Baseline ping/traceroute from R1 to R10	Measure initial path, delay, hops
2	Inject traffic (iperf) to induce congestion	Create routing stress
3	Measure again (ping, traceroute, throughput)	Capture OSPF path decision, delay
4	Apply ML-driven cost changes on routers	Based on real latency measurements
5	Repeat measurement (ping, traceroute)	Compare path, delay, route path, load

Python scripts run on Linux servers are used to automate ML-driven routing decisions and data collection. These scripts collected real-time latency, interface load, and traffic data by means of SSH and SNMP interfacing with network devices. Then, Ansible playbooks are used to coordinate configuration modifications across all routers, then implement ML model outputs (e.g., modified OSPF costs), and launch tests such as ping, traceroute, and iperf. The combined use of Python and Ansible allows for quick retrieval of pre- and post-ML performance metrics across the GNS3-emulated network and repeatable, scalable simulations.

Key network metrics were compared before and after cost changes to assess the effectiveness of the ML improvements. Table 2 lists the measurement tools and their coverage. The comparison underlined how ML-informed routing choices influenced important measures, including load distribution, throughput, route path, and round-trip time (RTT). Every measure was examined over several test cycles to guarantee consistency and dependability in the results.

Table 2: Measurement tools tested pre-ML and post-ML.

Metric	Tool	Pre-ML	Post-ML
RTT (ms)	ping	Successful	Successful
OSPF path/hops	traceroute	Successful	Successful
Interface load	SNMP	Successful	Successful
OSPF cost changes	CLI logs	Successful	Successful
Throughput (Mbps)	iperf	Successful	Successful

3.2. Data Collection

The data collecting phase was run in a simulated network environment built in GNS3 comprising 10 Cisco routers linked in a partial mesh topology. To mimic real-world network dynamics, tools including iperf, traffic generators, and manual link degradation simulated different traffic patterns and link behaviors. Recorded key performance indicators were round-trip latency, OSPF-calculated path costs, interface traffic load, packet loss, and link status. Every end-to-end communication path, including from R1 to R10, was evaluated under various load conditions, including idle, low, moderate, and high utilization. To act as input characteristics for ML models, data were timestamped and labeled with the relevant traffic conditions. The resulting dataset was labeled with traffic conditions, normalized metric ranges, and handled missing values during preprocessing. Subsequent phases' efficient model training and testing were founded on this structured dataset. Various tools and protocols were used to instrument the network to gather important performance data at specified intervals. Table 3 lists the tools employed, the measurements taken, and their corresponding sampling rates.

Table 3: Data collection tools and sampling intervals.

Metric	Tool	Sampling Interval
Latency (ping)	ICMP script	30 seconds
Interface Utilization	SNMP	30 seconds
OSPF LSA Events	Syslog/SNMP traps	Event-driven
Traffic Volume	NetFlow	1-minute summary

These tools gave a real-time, high-fidelity view of network conditions. All router pairs had scripted ICMP-based latency tests; SNMP agents were set up to gather interface counts and usage. NetFlow was used for total traffic volume analysis; syslog and SNMP traps logged OSPF Link State Advertisement (LSA) changes.

3.3. Model Training

During this stage, distinct ML models were deployed for each of the four enhancement goals. Trained on latency, traffic load, and observed performance, supervised regression models such as Random Forest Regressor and XGBoost were used for dynamic cost optimization to forecast ideal OSPF costs. Traffic prediction was done using time-series forecasting models like LSTM and ARIMA, trained on past interface throughput data to project future congestion patterns. Using Isolation Forest and Autoencoder-based deep neural networks, anomaly detection used unsupervised learning to find unusual latency and traffic patterns that might indicate performance degradation or possible failures. Trained on historical data, classification models such as Random Forest Classifier and Logistic Regression forecast binary results suggesting whether a link is likely to fail. To increase model accuracy and generalizability, grid search and cross-validation were used for hyperparameter tuning. The models were validated using 80/20 training/test splits.

3.4. Real-Time ML Integration

The ML models will be integrated into a real-time decision-making system operating in parallel with the OSPF routing process after model training and validation. Using SNMP polling and NetFlow data, this module constantly tracks live network metrics and supplies these inputs to the trained models. Model forecasts serve as the basis for dynamic interface cost changes by means of OSPF configuration file updates via automated scripts and APIs. The system can proactively preconfigure OSPF to avoid congested routes forecasted by the LSTM/ARIMA models for traffic forecasting. Anomalies found by the unsupervised models will set off alerts or temporary path changes to preserve service quality. Preemptive traffic rerouting from vulnerable links using failure prediction results will help to minimize disturbance and downtime. This close integration produces a feedback loop in which ML insights constantly hone routing choices, changing to network conditions in almost real time.

3.5. Performance Evaluation

To show the effectiveness of the ML model applied to the OSPF network, the performance evaluation tests will be done before and subsequent to the ML integrations. Different scenarios will be produced to capture the comprehensive link stability, latency, packet loss, and congestion. All the test cases will be measured from low to high traffic loads between selected routers. Default OSPF metrics will be used as baseline performance. After the experimental simulations, the results are expected to show improvement in latency, convergence, link failure, and path selection after applying ML models. Furthermore, traffic prediction techniques will be implemented to dynamically assign OSPF costs that are correlated with the actual network behavior. The experimental setup is expected to efficiently detect anomalies and failures to avoid loops and performance issues in the future.

3.6. ML-Driven Simulation Cases

Multiple ML models are employed across various simulation scenarios. The experiments are evaluated based on four key performance factors: failure forecasting, anomaly detection, traffic prediction, and cost optimization. Each simulation is conducted using a specific ML model to examine its potential impact on network performance. To facilitate comprehensive evaluation, different routing paths are selected for each simulation, and the resulting data are collected for subsequent analysis. Table 4 below shows the simulation context corresponding to the ML models applied to the OSPF network.

Table 4: ML models and corresponding simulation outcomes.

Simulation ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Technique	Dynamic Cost Optimization	Traffic Prediction	Anomaly Detection	Failure Prediction	Dynamic Cost Optimization	Traffic Prediction	Anomaly Detection	Failure Prediction
Model Used	XGBoost	LSTM	Isolation Forest	Random Forest	Random Forest	ARIMA	Autoencoders	Logistic Regression
Objective	**Optimize interface cost based on real-time data**	**Predict traffic congestion for proactive path selection**	**Detect network irregularities (e.g., abnormal latency)**	**Predict link failure based on historical data**	**Optimize interface cost based on real-time data**	**Forecast traffic load to guide path selection**	**Detect outliers or anomalies in packet loss**	**Predict potential link/node failure based on patterns**
Source	R1	R1	R2	R6	R3	R8	R4	R7
Destination	R3	R5	R4	R7	R9	R10	R6	R8
Latency (ms)	8	40	55	15	22	30	65	10
Actual Cost	1	2	5	3	2	2	4	1
Predicted Cost	0.9	2.1	5.2	3.4	2.3	2.1	4.5	1.1
Traffic Load	Low	High	Medium	Low	Low	High	High	Low
Actual Traffic Prediction	100Mbps	200Mbps	150Mbps	80Mbps	110Mbps	220Mbps	250Mbps	60Mbps
Predicted Traffic	98Mbps	210Mbps	160Mbps	75Mbps	115Mbps	225Mbps	255Mbps	62Mbps
Anomaly Detected	No	No	Yes	No	No	No	Yes	No
Link / Node Failure Forecast	No	No	No	Yes (Link Failure Forecast)	No	No	No	Yes (Node failure forecast)

4. Results

The goal of integrating ML models was to improve OSPF performance by making path selection and optimization more efficient. Data were gathered both prior to and following the application of ML models to the network. The simulation environment comprised routers operating OSPF with traffic injected to replicate both low and heavy network loads. ML models such as XGBoost, LSTM, ARIMA, Isolation Forest, and logistic regression were employed at various phases. The following sections present an analysis of the impact of ML integrations on critical network performance metrics. The data gathered during the simulation tests are analyzed to reveal adjustments in routing efficiency, latency variation, and link reliability.

4.1. Results and Analysis

Since it is set to be an enterprise-class network, multiple different paths between suggested router have been considered for ML-enhancement applications. The simulation recorded values were latency and cost which are crucial for OSPF operated networks. These two values will impact the routing decision-making, which is crucial for the ML models to efficiently predict the traffic and adjust the path accordingly. Table 5 lists five separate simulations with different congestion levels and different routes in which the ML models proficiently predicted the cost and latency.

Table 5: Latency and cost predictions after ML applications.

Simulation ID	Source	Destination	Latency (ms)	Predicted Latency (ms)	Actual Cost	Predicted Cost	Traffic Load
1	R1	R2	5	6	1	1.1	Low
2	R1	R3	20	23	2	2.2	Low
3	R1	R4	50	52	3	3	High
4	R2	R5	10	10	2	2.1	Low
5	R5	R10	60	64	5	5.3	High

The results observed from the table indicated that ML models closely predicted the crucial values required for path selection. Hence, an ML-driven network can take advantage of these procedures to predict link state and make forwarding decisions. It can also be observed that the selected route paths are not always optimal, and in some cases a longer path has been selected that led to higher latency and OSPF costs. These results clearly present the possibility of using ML models for OSPF path selection optimization along with traffic predictions, detecting anomalies and potential link/node failures. Figure 3 presents a clustered chart of four optimized OSPF paths in which, after applying ML models, latency was enhanced compared to the traditional OSPF baselines.

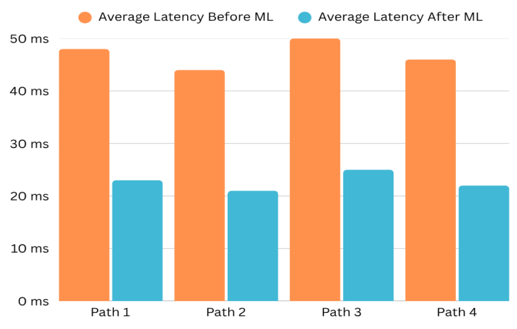

Figure 3: Latency before and after ML applications.

Another important measure is throughput, which the chart in figure 4 uses to compare the network's data speeds (in Mbps) before and after using ML models to improve OSPF routing choices across four different paths. Each pair of bars shows the throughput performance of a specific route. The left bar represents the original throughput, and the right bar shows the better throughput after using ML for cost optimization, anomaly detection, and traffic prediction.

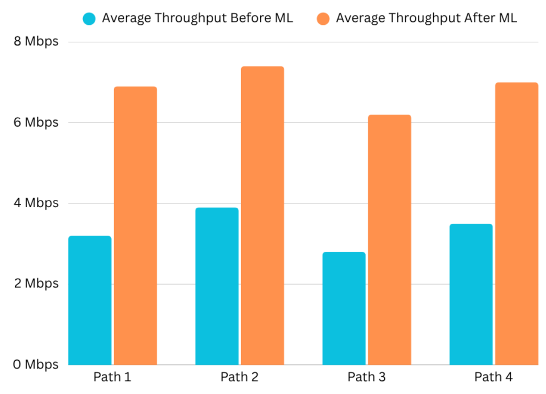

Figure 4: Throughput before and after ML applications.

5. Discussion

The aggregated metric measurement undoubtedly shows the advantages of applying ML models to enhance OSPF cost calculation and traffic predictions. Before the ML intervention, the network had more latency, less throughput, and less efficient routing decisions; under high traffic conditions, OSPF paths were suboptimal. The network showed significant improvements after the application of ML methods, including XGBoost for dynamic cost optimization, LSTM and ARIMA for traffic prediction, and Isolation Forest for anomaly detection. Specifically, the algorithm's capacity to proactively change OSPF interface costs depending on real-time traffic data and collected data caused notable latency reduction. Improving total bandwidth use, throughput also rose as the routing protocol changed to less congested routes. Furthermore, post-ML integration, route paths with low cost and less congestion that were ignored before started to get occupied and used again to obtain better traffic distribution across the entire topology. The simulation results, achieved in a virtual GNS3 platform, sufficiently demonstrate significant performance improvements after integrating ML models, with up to 30% enhancement in latency reduction and throughput enhancement. Over OSPF in its native configuration, which calculates cost statically based on interface bandwidth alone, the dynamic cost adaptation of ML models on pattern learned separates routing intelligence by a wide margin. The absence of real-time adaptability, which is nonexistent in traditional OSPF and only partially addressed in hybrid SDN–OSPF solutions, is ushered in by this work.

Several prior works have studied optimization challenges. For example, Bernárdez *et al.*'s MAGNNETO [1] uses GNNs to optimize OSPF weights with multiple agents. However, while very promising, their research is more focused on distributed learning at a higher conceptual graph level, not including end-to-end simulation outputs or directly addressing anomaly and failure prediction challenges. On the other hand, this approach offers a modular one where different ML models are stacked each to handle a specific aspect of network performance and stability. Similarly, Wei *et al.*'s HEATE algorithm [17] includes heuristic methods coupled with SDN so that energy consumption under OSPF is reduced. While it deals with path optimization from the energy conservation perspective, here the concern is toward performance metrics such as delay, throughput, and resiliency for applications with stringent latency requirements. In addition, the system operates on traditional OSPF networks, making it more relevant to legacy networks that do not have SDN infrastructure.

Caria and Jukan's [18] work on hybrid SDN/OSPF link capacity planning provides an insightful study of the promise of topology-aware planning to increase failure tolerance. Their methodology, however, requires centralized SDN controllers and is not dynamic according to live traffic statistics. This work's methodology, by contrast, utilizes Python automation and Ansible to communicate directly with the GNS3 emulated routers and permits the integration of ML-based decisions almost in real-time. In comparison with Zhang *et al.* [11], who only focused on anomaly detection using boosted decision trees, this work extends beyond isolated detection jobs and integrates these detections into a larger routing decision framework. Along the way, the system demonstrates how an ML-based pipeline can affect real routing decisions like path change and improved throughput not just alerting or logging anomalies. Finally, although Usama *et al*. [8] provide a valuable overview of unsupervised ML in networking, this paper presents a clear, simulated application of these concepts in OSPF-based systems with full-stack implementation and measurable performance.

6. Conclusions

To sum up, this study addresses the challenges that traditional OSPF faces that could be overcome by the use of ML methods in contemporary networks. Research outcomes effectively presented improvements in path selection, latency, and throughput after the employment of models such as XGBoost, LSTM, ARIMA, Isolation Forest, and logistic regression. The simulation observed network behavior in multiple different scenarios with variations in traffic loads. ML models helped in high-traffic situations by dynamically adjusting OSPF costs and providing proactive traffic management. Moreover, predicting anomalies, congestion, and possible link/node failures were included in the ML-provided features that led to a more robust and resilient network. The simulation results have indicated that ML models can support and improve traditional routing protocols by providing intelligent, adaptive decision-making that enhances the network's overall performance. Both latency and throughput were significantly improved, with an average performance gain of approximately 30%. This paper lays the groundwork for future research on applying ML for routing in enterprise-class and real-time networks by introducing new approaches for networks to be more responsive, scalable, efficient, automated, and self-improving.

Data availability: The data are available upon reasonable request by the author.

Conflicts of interest: The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this study.

Funding: The authors did not receive support from any organization for the submitted work.

References

[1] G. Bernárdez et al., "MAGNNETO: A graph neural network-based bulti-agent system for traffic engineering." *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 2, pp. 494–506, 10 Jan. 2023. https://doi.org/10.1109/tccn.2023.3235719

[2] B. Al-Musawi and P. Branch, "Identifying OSPF anomalies using recurrence quantification analysis," arXiv (Cornell University), Jan. 2018. https://doi.org/10.48550/arxiv.1805.08087

[3] H. Saini, and A. K. Garg. "Impact of mathematical optimization on OSPF routing in WDM optical networks", in: H. Malik, S., Srivastava, Y. Sood and A. Ahmad (eds), applications of artificial intelligence techniques in engineering. advances in intelligent systems and computing, vol 697, pp 201-208. Springer, Singapore. https://doi.org/10.1007/978-981-13-1822-1_19

[4] R. Etengu et al. "Deep learning-assisted traffic prediction in hybrid SDN/OSPF backbone networks." *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, 25 Apr. 2022, pp. 1–6. https://doi.org/10.1109/NOMS54207.2022.9789868

[5] A. Ikpe and I. Ekanem, "Engineering wired network performance enhancement in a route redistributed simulation based systems", *Big Data and Computing Visions*, vol, 4, no. 1, pp. 31-48, 2024. https://doi.org/10.22105/bdcv.2024.464144.1181

[6] W. Jiang et al., "Graph neural networks for routing optimization: challenges and opportunities," *Sustainability*, vol. 16, no. 21, p. 9239, Oct. 2024. https://doi.org/10.3390/su16219239

[7] X. Li, J. Li, J. Zhou, and J. Liu, "Towards robust routing: enabling long-range perception with the power of graph transformers and deep reinforcement learning in software-defined networks," *Electronics*, vol. 14, no. 3, p. 476, Jan. 2025. https://doi.org/10.3390/electronics14030476

[8] M. Usama, et al. "unsupervised machine learning for networking: techniques, applications and research challenges." *IEEE Access*, vol. 7, pp. 65579–65615, 2019. https://doi.org/10.1109/access.2019.2916648

[9] Q. Zhou, and D. Pezaros. "A prediction-based model for consistent adaptive routing in back-bone networks at extreme situations." *Electronics*, vol. 9, no. 12, pp. 2146–2146, 15 Dec. 2020. https://doi.org/10.3390/electronics9122146

[10] D. Szostak, A. Włodarczyk and K. Walkowiak, "Machine learning classification and regression approaches for optical network traffic prediction." *Electronics*, vol. 10, no. 13, p. 1578, 30 June 2021. https://doi.org/10.3390/electronics10131578

[11] J. Zhang, R. Gardner and I. Vukotic., "anomaly detection in wide area network meshes using two machine learning algorithms." *Future Generation Computer Systems*, vol. 93, pp. 418–426, Apr. 2019. https://doi.org/10.1016/j.future.2018.07.023

[12] K. Fotiadou, et al., “network traffic anomaly detection via deep learning.” *Information*, vol. 12, no. 5, p. 215, 19 May 2021.https://doi.org/10.3390/info12050215

[13] I. Fosić, et al. “Anomaly detection in NetFlow network traffic using supervised machine learning algorithms.” *Journal of Industrial Information Integration*, vol. 33, p. 100466, 1 June 2023. https://doi.org/10.1016/j.jii.2023.100466

[14] J. Papán et al., “The new label bit repair fast reroute mechanism,” *IEEE Access*, vol.12, pp. 46487 – 46503, 2024. https://doi.org/10.1109/access.2024.3379150

[15] J. Chen et al., “Dynamic routing optimization in software-defined networking based on a metaheuristic algorithm,” *Journal of Cloud Computing*, vol. 13 p.41, Feb. 2024. https://doi.org/10.1186/s13677-024-00603-1

[16] M. Goyal et al. “Improving convergence speed and scalability in OSPF: A survey.” *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 443–463, 2012. https://doi.org/10.1109/surv.2011.011411.00065

[17] W. Wei et al., “Energy-Aware traffic engineering in hybrid SDN/IP backbone networks.” *Journal of Communications and Networks*, vol. 18, no. 4, pp. 559–566, Aug. 2016/ https://doi.org/10.1109/jcn.2016.000079.

[18] M. Caria and A. Jukan. “Link capacity planning for fault tolerant operation in hybrid SDN/OSPF networks.” *2015 IEEE Global Communications Conference (GLOBECOM)*, 1 Dec. 2016, pp. 1–6. https://doi.org/10.1109/glocom.2016.7841957

[19] M. Caria et al., “Divide and conquer: Partitioning OSPF networks with SDN,” *2015 IFIP/IEEE International Symposium on Integrated Network Management* (IM), Ottawa, ON, Canada, 2015, pp. 467-474, https://doi.org/10.1109/inm.2015.7140324

[20] R. Amin et al., “A survey on machine learning techniques for routing optimization in SDN,” *IEEE Access*, vol. 9, pp. 104582–104611, 2021. https://doi.org/10.1109/access.2021.3099092

[21] A. Abrol, P. M. Mohan, and T. Truong-Huu, “A deep reinforcement learning approach for adaptive traffic routing in next-gen networks,” *arXiv.org*, Feb. 2024, https://doi.org/10.48550/arxiv.2402.04515

[22] S. Mahajan, R. Harikrishnan, and K. Kotecha, “Adaptive routing in wireless mesh networks using hybrid reinforcement learning algorithm,” *IEEE Access*, vol. 10, pp. 107961–107979, 2022. https://doi.org/10.1109/ACCESS.2022.3210993

[23] C. Pan et al., “Inverse coupled simulated annealing for enhanced OSPF convergence in IoT networks,” *Electronics*, vol. 13, no. 22, pp. 4332–4332, Nov. 2024. doi: https://doi.org/10.3390/electronics13224332

[24] C. V. Murudkar and R. D. Gitlin, “Optimal-capacity, shortest path routing in self-organizing 5G networks using machine learning,” *Wireless and Microwave Technology Conference*, pp. 1–5, Apr. 2019. https://doi.org/10.1109/WAMICON.2019.8765434

[25] P. Tshakwanda, S. T. Arzo, and M. Devetsikiotis, “Advancing 6G Network performance: AI/ML framework for proactive management and dynamic optimal routing”, *IEEE Open Journal of the Computer Society*, vol 5, pp.303-314, 2024. https://doi.org/10.1109/ojcs.2024.3398540

[26] S. Çoğay, M. Akkoç and G. Secinti, "ML-assisted dynamic multi-path routing for enhanced QoS," *2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Athens, Greece, 2024, pp. 1-6, https://doi.org/10.1109/CAMAD62243.2024.10942960