A Survey on the Integration of Machine Learning Techniques in Software-Defined Networking

[1]N. SENTHILKUMARAN, [2]Dr. R. SANKARASUBRAMANIAN

[1]Assistant Professor, [2]Principal
[1]Department of Computer Applications
[1]Vellalar College for Women, Erode, India
[1]senthilkumaran@vcw.ac.in , [2]rsankarprofessor@gmail.com

Abstract—**Software-Defined Networking (SDN) represents a fundamental paradigm shift in modern communication networks, offering extraordinary capabilities through its principles of programmability, centralized control, and abstraction. This innovative architecture fundamentally transforms how network infrastructures are managed and operated, moving away from static, hardware-centric configurations towards dynamic, software-driven orchestration. However, the inherently dynamic, complex, and evolving nature of contemporary SDN environments, coupled with ever-increasing demands for enhanced efficiency, resilience, and adaptability, has led to a increasing interest in applying advanced Machine Learning techniques. This powerful integration leverages ML's analytical prowess to address critical networking challenges such as intelligent traffic classification, robust intrusion detection and prevention, optimal resource allocation, and proactive fault management. This survey provides a comprehensive overview of recent ML approaches rigorously applied within SDN paradigms. It meticulously categorizes various proposed solutions based on their primary application domains (e.g., security, QoS), the families of Machine Learning models employed (e.g., supervised, unsupervised, deep learning), the datasets utilized for training and evaluation, and the key performance metrics reported in academic literature. Furthermore, this paper critically examines the prevalent challenges and open issues that impede the widespread and successful integration of ML in SDN, including concerns related to scalability in large-scale deployments, the scarcity of realistic and labelled network datasets, and the inherent security vulnerabilities of ML models themselves (e.g., adversarial attacks). Finally, it outlines several promising potential research directions, such as federated learning, lightweight ML models, and Explainable AI, that aim to overcome these obstacles and foster the development of smarter, more resilient, and truly self-optimizing network architectures for the future.**

Index Terms—**SDN, paradigm, QoS, Security, ML and self-optimizing, supervised and unsupervised learning**

I. Introduction

The advent of Software-Defined Networking marks a pivotal transformation in network architecture, fundamentally altering the landscape of how modern digital infrastructures are designed, deployed, and managed. This paradigm shift moves away from traditional, vertically integrated networking devices with tightly coupled control and data planes towards a more programmable, centralized, and abstract model Martín et al., 2019, p. 872. By decoupling the control logic from the underlying packet forwarding functions, SDN empowers network administrators with unprecedented flexibility and granular control over network resources through centralized, software-based controllers like ONOS, Ryu, and OpenDaylight. This architectural innovation facilitates dynamic network reconfigurations, significantly simplifies network management tasks, and enables the rapid deployment of new services and applications, all contributing to more agile, responsive, and cost-effective network infrastructures Alhilali & Montazerolghaem, 2023.

Despite these transformative advantages, the modern network landscape is characterized by its continuous evolution, presenting a constant stream of challenges that necessitate innovative and intelligent solutions. The exponential increase in network scale and complexity, driven by factors such as the proliferation of IoT devices, the widespread adoption of cloud computing, and the surging demand for high-bandwidth applications, exerts immense pressure on existing network management capabilities. Concurrently, the omnipresent threat of sophisticated and constantly evolving cyberattacks demands security mechanisms that can adapt and respond intelligently in real-time. Traditional network management paradigms, often relying on static configurations and manual interventions, frequently struggle to keep pace with these dynamic requirements, leading to inefficiencies, increased operational costs, performance bottlenecks, and heightened security vulnerabilities. This continuous evolution and the emergence of novel threats mandate novel and robust approaches to improve both security and performance across contemporary network infrastructures Khekare et al., 2023, p. 2.

In this critical context, the convergence of Machine Learning and Software-Defined Networking has rapidly emerged as a powerful and promising paradigm for significantly enhancing network management capabilities. ML's inherent ability to analyze vast amounts of network data, learn complex behavioral patterns, predict future network states, and automate intricate decision-making processes aligns perfectly with the dynamic, data-rich requirements of modern SDN environments. This profound synergy enables intelligent automation and optimized resource allocation, thereby proactively addressing multifaceted challenges related to Quality of Service, bolstering network security, and optimizing overall network performance Martín et al., 2019, p. 872.

The integration of ML within SDN architectures is particularly beneficial for developing robust and proactive defence mechanisms against sophisticated cyber threats, such as Distributed Denial of Service attacks Akinyele & Olaoye, 2024. ML and deep learning techniques have demonstrably proven highly effective in identifying and neutralizing various network threats by discerning intricate patterns and subtle anomalies that frequently elude traditional signature-based detection methods Ali et al., 2023; Bahashwan et al., 2023; Restuccia et al., 2018, p. 4836. For instance, a range of supervised machine learning models, including decision trees (e.g., J48, Random Tree, REP Tree, Random Forest), neural networks (e.g., Multi-Layer Perceptron), and Support Vector Machines, have showcased remarkable efficacy when deployed in intrusion detection systems. These models achieve high detection rates even against complex, low-rate DDoS attacks that are notoriously difficult to identify using conventional means Khekare et al., 2023, p. 4. The adaptive nature of ML algorithms further ensures continuous learning and improvement from new data, maintaining effective defense mechanisms that can stay ahead of rapidly evolving threat landscapes and the persistent emergence of novel attack vectors Yasarathna & Le-Khac, 2025.

Beyond the crucial domain of security, the multifaceted application of ML in SDNs extends to various other critical aspects of network management. These include intelligent resource allocation, proactive traffic engineering, and predictive maintenance capabilities Faezi & Shirmarz, 2023, p. 316. This capability is particularly vital given the rapid expansion of interconnected network devices (e.g., IoT, mobile endpoints) and the resultant exponential increase in administrative complexity, which traditional network architectures demonstrably struggle to manage efficiently Bahashwan et al., 2023. The inherently programmable nature of SDN, seamlessly coupled with the powerful analytical capabilities of machine learning, thus provides a flexible, intelligent, and scalable framework for effectively addressing these complexities, thereby significantly enhancing network resilience, optimizing operational efficiency, and reducing human intervention Akinyele & Olaoye, 2024; Khekare et al., 2023, p. 3. This survey aims to provide a comprehensive and up-to-date review of the current academic literature, specifically focusing on how various machine learning techniques contribute to improving Quality of Service metrics, bolstering network security postures, and optimizing overall performance within diverse SDN environments.

Table 1 Difference between Traditional and SDN

Feature	Traditional Networking	Software-Defined Networking (SDN)
Architecture	Vertically integrated devices; **Tightly coupled** control and data planes.	**Decoupled** control logic from packet forwarding functions; **Centralized** control.

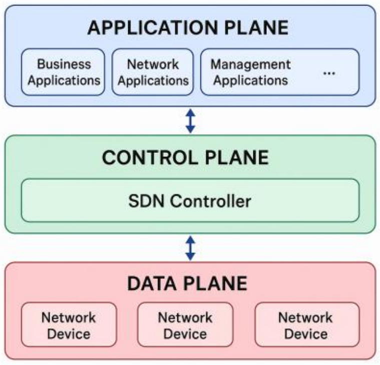
2. Background

To fully understand the transformative potential that emerges from the integration of Machine Learning and Software-Defined Networking, it is important to first build a strong grasp of the fundamental ideas that define both fields. By examining the core principles behind SDN and the major Machine Learning paradigms, we can better appreciate how their combination offers effective solutions to the growing complexity and dynamic challenges of modern network environments. This section outlines these foundational concepts and prepares the groundwork for a deeper exploration of their collaborative applications.

2.1 Software-Defined Networking Architecture

Software-Defined Networking represents a fundamental architectural shift that redefines how network control and data forwarding functions are structured and managed. Its core innovation lies in explicitly separating the network's control logic (the "control plane") from the underlying packet forwarding functions (the "data plane"). This deliberate separation centralizes network intelligence, allowing for programmatic and global control over the entire network infrastructure from a single, unified point. The typical SDN architecture is conventionally understood as comprising three distinct, yet intrinsically interconnected, layers: the Application Layer, the Control Layer, and the Data Layer.

Fig. 1 SDN Architecture



- ***Application Layer***: Positioned at the topmost level of the SDN architecture, this layer is home to various network applications and high-level services. These applications, which can range from sophisticated traffic engineering tools, advanced security services (such as firewalls and intrusion prevention systems), dynamic load balancers, to comprehensive network virtualization solutions, communicate their specific requirements and policies to the control plane. They crucially leverage the abstracted, network-wide view provided by the SDN controller to implement complex network policies, achieve specific service-level objectives, and deliver diverse high-level network services. This abstraction empowers developers to create custom, innovative applications that can interact with and influence network behavior dynamically and programmatically, fostering rapid innovation in network services.
- ***Control Layer***: Often revered as the "brain" or "nervous system" of the SDN architecture, the control layer typically consists of one or more centralized SDN controllers (e.g., Open Network Operating System, Ryu, OpenDaylight). These controllers maintain a global, holistic, and up-to-date view of the entire network topology, actively monitor traffic flows, and track the real-time operational states of all connected network devices. Their primary function is to translate the high-level service requests and abstract policies received from the application layer into concrete, granular forwarding rules. These rules are then meticulously pushed down to the forwarding elements in the data plane. The controller-network interface, most commonly implemented using protocols like OpenFlow, is fundamental for this bidirectional communication, enabling the controller to dynamically configure, manage, and orchestrate the behavior of forwarding devices. The centralized nature of this layer is pivotal for enabling intelligent, network-wide optimization, efficient resource allocation, and advanced automation, providing a unified point of control for complex network operations.
- ***Data Layer***: Situated at the lowest level, this layer comprises the actual network hardware devices, such as programmable switches and routers. These devices, often referred to as "forwarding elements" or "data plane elements," are solely responsible for high-speed forwarding of data packets based on the rules and instructions programmed into them by the SDN controller. They expose a programmatic interface (e.g., OpenFlow, P4) to the control layer, allowing the controller to dynamically modify their forwarding tables, port configurations, and overall behavior in real-time. Critically, the data plane is designed for maximum efficiency in packet processing and is typically stateless, meaning it relies entirely on the control plane for any intelligence, policy decisions, or stateful operations.

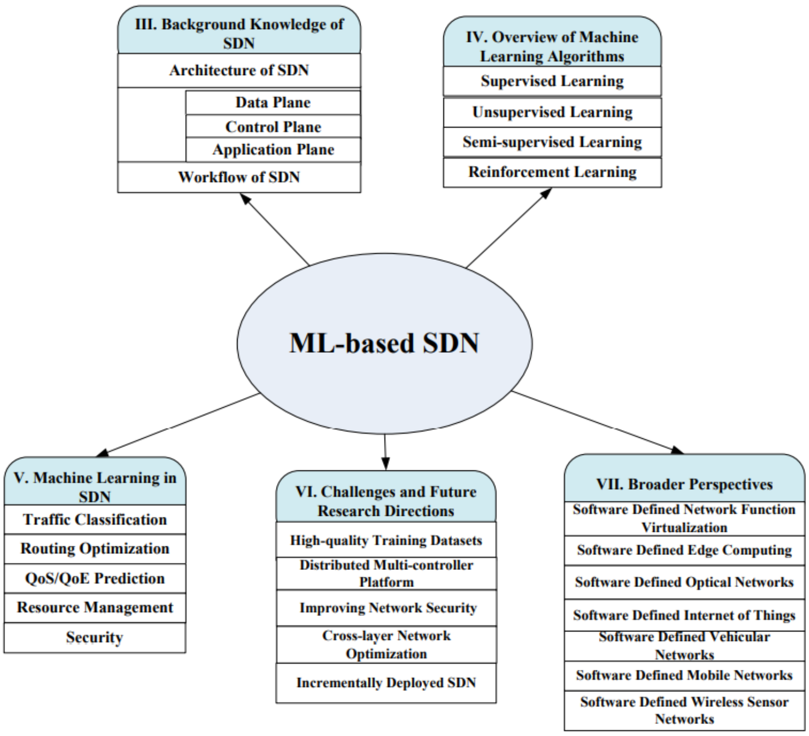
This elegant, layered architecture provides unparalleled flexibility, allowing network administrators to manage, secure, and optimize network resources with a level of agility previously unattainable in traditional networking paradigms. However, even with these profound architectural advantages, SDN environments inherently face a new set of challenges. These include ensuring scalability for massive, geographically distributed deployments, mitigating novel security vulnerabilities inherent in centralized control (e.g., single point of failure, controller compromise), addressing potential controller bottlenecks under extremely heavy traffic loads, and adapting intelligently to highly dynamic and unpredictable traffic patterns. These challenges underscore the profound need for advanced intelligence and autonomous decision-making capabilities, making Machine Learning a natural, powerful, and increasingly indispensable complement to SDN.

2.2 Machine Learning in Networking

Machine Learning encompasses a diverse set of algorithms and statistical models that empower computer systems to perform specific tasks without explicit programming instructions. Instead, these systems learn from data, identify patterns, make predictions, and adapt their behavior over time. In the context of networking, ML offers a sophisticated suite of tools to analyze vast quantities of complex network data, learn intricate behavioral patterns, detect subtle anomalies, predict future network states, and automate complex decision-making processes. The application of ML in SDN environments can be broadly categorized based on the underlying learning paradigm employed: supervised, unsupervised, reinforcement, and deep learning.

- ***Supervised Learning***: This paradigm is characterized by training models on *labeled datasets*, where each input example is explicitly associated with a known, correct output. The fundamental goal of supervised learning is for the model to learn an accurate mapping function from input features to output labels. Once trained, this function can then be used to predict outputs for new, previously unseen data. In SDN, supervised learning is extensively utilized for a wide array of tasks:

Fig. 2. ML based SDN


III. Background Knowledge of SDN
Architecture of SDN
Data Plane
Control Plane
Application Plane
Workflow of SDN
IV. Overview of Machine Learning Algorithms
Supervised Learning
Unsupervised Learning
Semi-supervised Learning
Reinforcement Learning
ML-based SDN
V. Machine Learning in SDN
Traffic Classification
Routing Optimization
QoS/QoE Prediction
Resource Management
Security
VI. Challenges and Future Research Directions
High-quality Training Datasets
Distributed Multi-controller Platform
Improving Network Security
Cross-layer Network Optimization
Incrementally Deployed SDN
VII. Broader Perspectives
Software Defined Network Function Virtualization
Software Defined Edge Computing
Software Defined Optical Networks
Software Defined Internet of Things
Software Defined Vehicular Networks
Software Defined Mobile Networks
Software Defined Wireless Sensor Networks

  - ***Classification***: This involves categorizing network entities or events into predefined classes. Examples include identifying different traffic types (e.g., VoIP, video streaming, HTTP, P2P), detecting specific types of known cyberattacks (e.g., DoS, probing, malware propagation), or classifying network events (e.g., link failure, congestion onset). Popular algorithms include Support Vector Machines, Decision Trees (e.g., C4.5, CART), K-Nearest Neighbors, Logistic Regression, and Naive Bayes classifiers.
  - ***Regression***: This involves predicting continuous numerical values. In SDN, regression models can be employed to predict metrics such as future bandwidth demand, anticipated network latency, or projected traffic loads over a given time horizon.
- ***Unsupervised Learning***: In contrast to supervised learning, unsupervised learning deals with *unlabeled data*. The primary objective of unsupervised learning algorithms is to discover inherent patterns, underlying structures, or hidden relationships within the data without any prior knowledge of output labels. This paradigm is particularly valuable in networking scenarios where obtaining labeled data can be resource-intensive, time-consuming, or practically impossible. Key applications in SDN environments include:
  - ***Clustering***: This involves grouping similar network flows, devices, user behaviors, or events together based on their intrinsic characteristics. Clustering can help identify different user groups, application clusters, or even distinguish between normal and abnormal network behaviors by forming distinct clusters. Algorithms like K-means, DBSCAN, and Gaussian Mixture Models are commonly used.
  - ***Anomaly Detection***: This is a critical application for identifying unusual or suspicious network activities that deviate significantly from established normal behavior. Anomaly detection is invaluable for proactively detecting novel cyber threats (e.g., zero-day attacks), identifying subtle performance degradations, or pinpointing misconfigurations. Autoencoders, Isolation Forests, and One-Class SVMs are popular choices for identifying outliers or anomalies within network traffic and telemetry data.
- ***Reinforcement Learning***: Reinforcement Learning is a unique learning paradigm where an "agent" learns to make a sequence of optimal decisions in an "environment" to maximize a cumulative "reward" signal. The agent learns through a process of trial and error, taking actions, observing the resultant state changes, and receiving feedback in the form of rewards or penalties from the environment. This paradigm is exceptionally well-suited for dynamic network scenarios where decisions need to be made continuously and adaptively in response to changing network conditions. In SDN, RL is being applied for:
  - ***Optimal Path Selection***: Dynamically adjusting routing paths and forwarding policies to minimize metrics like latency, maximize throughput, or balance load across diverse network segments.
  - ***Adaptive Resource Management***: Intelligently allocating bandwidth, buffer space, CPU cycles, or other critical network resources in real-time based on the current network state, application requirements, and dynamic QoS demands.
  - ***Adaptive Security Policies***: Developing intelligent security policies that can learn to respond to evolving attack strategies and autonomously apply mitigation actions. Algorithms like Q-learning, SARSA, and Deep Q-Networks are prominent in this rapidly evolving area.
- ***Deep Learning***: A specialized subfield of machine learning, deep learning leverages artificial neural networks with multiple hidden layers (often referred to as "deep neural networks") to learn complex, hierarchical patterns and abstract representations directly from raw input data. DL has achieved groundbreaking success in tasks involving large-scale, high-

dimensional, and often unstructured datasets, frequently outperforming classical ML methods. Its applications in SDN are extensive and continuously expanding, including:

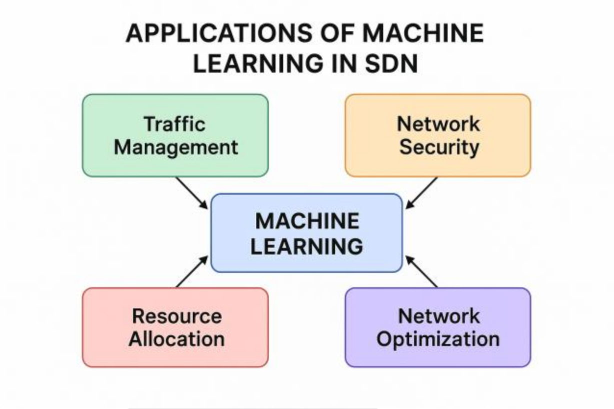
- ***Complex Traffic Classification:*** DL models, particularly Convolutional Neural Networks for spatial feature extraction and Recurrent Neural Networks or Long Short-Term Memory networks for sequential data analysis, are exceptionally effective for classifying intricate traffic patterns, including encrypted traffic where traditional feature engineering proves challenging.
- ***Advanced Intrusion Detection*:** Deep learning models can learn highly complex and subtle attack signatures and behavioral patterns from massive network flow data, leading to more accurate and resilient intrusion detection systems.
- ***Network State Prediction and Forecasting*:** Leveraging LSTM networks, deep learning can accurately predict future network states, traffic loads, and potential congestion points based on historical time-series data, enabling proactive network management.
- ***Automated Policy Generation*:** DL can be used to develop models that learn to generate optimal operational policies for the SDN controller, moving towards truly autonomous network management.

The synergistic combination of SDN's programmable and centralized control capabilities with ML's powerful analytical, predictive, and adaptive capabilities aims to achieve smarter, more resilient, and ultimately self-optimizing networks. This integration facilitates dynamic network adjustments, enables predictive management strategies, and fosters autonomous responses to the complex and ever-changing demands of modern network infrastructures, paving the way for next-generation network intelligence.

3. Applications of Machine Learning in SDN

The integration of Machine Learning within Software-Defined Networking has opened up numerous avenues for significantly enhancing network operations across various critical domains. This section delves into specific application areas where ML techniques are profoundly impacting SDN environments, offering intelligent solutions to longstanding challenges.

Fig. 3 Applications of Machine Learning in SDN



3.1 Traffic Classification

Accurate and real-time traffic classification is a cornerstone of effective network management. It is indispensable for enabling fine-grained Quality of Service differentiation, enforcing precise security policies, optimizing resource allocation, and performing efficient traffic engineering. Traditional traffic classification methods, which often rely on well-known port numbers, IP addresses, or Deep Packet Inspection of packet payloads, face growing challenges in modern networks. These challenges stem from the widespread use of encryption (e.g., HTTPS, VPNs), dynamic port assignments by applications, and the sheer volume and velocity of network data, which make byte-level inspection computationally intensive and privacy-intrusive. Machine Learning models offer a powerful and adaptive alternative by learning intricate patterns from flow-level features, even in scenarios involving encrypted traffic where payload content is obscured.

ML models such as Support Vector Machines, Random Forest, and Convolutional Neural Networks are widely applied for accurately classifying diverse application traffic types, including Voice over IP, HTTP, video streaming services, and peer-to-peer (P2P) file sharing. These models can utilize a rich set of flow-level features extracted directly from OpenFlow switches, including statistical properties of packet sizes, inter-arrival times between packets, overall flow duration, byte counts, and connection states, to build robust and highly accurate classification capabilities. For instance, SVMs are effective due to their ability to find optimal separating hyperplanes in high-dimensional feature spaces, thereby maximizing the margin between different traffic classes. Random Forests, leveraging the power of ensemble learning, combine multiple decision trees to improve overall accuracy, reduce overfitting, and enhance generalization capabilities across varied traffic patterns.

Notably, Deep Learning models, especially Convolutional Neural Networks for their ability to learn spatial hierarchies in data and Recurrent Neural Networks or their advanced variants like Long Short-Term Memory for sequential data processing, have demonstrated superior performance in classifying encrypted traffic. This exceptional capability arises because DL models can automatically extract complex, abstract, and subtle features directly from raw or minimally pre-processed flow data, thereby sidestepping the inherent limitations of manual feature engineering when dealing with encrypted contexts. The ability to accurately identify application types regardless of encryption is critically important for maintaining comprehensive network visibility, enforcing appropriate policies, and ensuring effective control over network resources in an era increasingly dominated by secure protocols like HTTPS and encrypted tunnels. Furthermore, DL can handle the massive scale of data generated in modern SDNs, providing efficient classification even with high-dimensional feature sets.

3.2 Intrusion Detection and Security

The centralized intelligence of SDN controllers, while offering unprecedented flexibility and programmability, concurrently presents an attractive and potentially high-impact target for sophisticated cyber adversaries. A compromise of the SDN controller can have catastrophic network-wide consequences, underscoring that robust and proactive security mechanisms, particularly for intrusion detection and prevention, are absolutely paramount in SDN environments. Machine Learning-based Intrusion Detection Systems leverage the inherent pattern recognition and anomaly detection capabilities of ML to identify malicious activities that often bypass or evade traditional signature-based security tools.

ML-based IDSs in SDN encompass a broad range of techniques, tailored to address different aspects of network security:

- ***Supervised ML for Known Attack Detection*****:** Algorithms such as Logistic Regression, K-Nearest Neighbours, Decision Trees, and ensemble methods like Random Forest and XGBoost are extensively used for detecting well-known and previously characterized attack types, including Denial of Service and Distributed Denial of Service attacks. These models are meticulously trained on labeled datasets containing both normal network traffic and traffic exhibiting known attack patterns, allowing them to learn to accurately distinguish between benign and malicious network behaviours. For example, studies have consistently shown that XGBoost offers exceptionally high accuracy and remarkably low error rates in DDoS detection and mitigation within SDN environments, frequently outperforming other classical ML techniques like Logistic Regression and Naive Bayes Arvind & Radhika, 2023. A comprehensive comparison of various classification methods, including Support Vector Machines, K-Nearest Neighbours, Decision Trees, Multi-Layer Perceptron, and Convolutional Neural Networks, has highlighted their differing efficacies in detecting DDoS attacks specifically within SDN environments, providing valuable insights into algorithm selection Ali et al., 2023; Singh, 2021.
- ***Unsupervised ML for Anomaly Detection*****:** For detecting novel, sophisticated, or zero-day attacks—threats for which no prior signatures or labels exist—unsupervised learning methods are highly effective. Techniques such as K-means clustering, DBSCAN, and Autoencoders are prominently employed to identify anomalies: network activities or patterns that deviate significantly from an established 'normal' baseline. K-means can group similar traffic flows or behaviors, allowing data points that do not fit into any cluster, or form very small, distinct clusters, to be flagged as anomalous. Autoencoders, a type of neural network, learn a compressed, low-dimensional representation of normal network traffic; any data point that cannot be accurately reconstructed by the autoencoder (i.e., has a high reconstruction error) is likely an anomaly and a potential indicator of an attack.
- ***Deep Reinforcement Learning for Adaptive Mitigation*****:** Deep Reinforcement Learning approaches enable the SDN controller to not only detect but also to learn and execute highly adaptive mitigation policies in response to ongoing and evolving attacks. In this paradigm, an RL agent, typically integrated with the SDN controller, continuously observes the network state (e.g., real-time traffic volume, detected attack signatures, resource utilization). Based on these observations, it takes dynamic routing actions (e.g., programming new forwarding rules to isolate malicious flows, rerouting traffic away from compromised segments, blocking specific source IPs) and receives immediate feedback or "rewards" based on the effectiveness of its actions in mitigating the threat and maintaining network performance. This continuous learning cycle allows for dynamic, intelligent, and autonomous responses to evolving attack strategies without requiring extensive human intervention, moving towards self-healing networks.

Common datasets utilized for training and evaluating these SDN-specific IDSs include well-established public datasets like NSL-KDD and CICIDS2017, alongside an increasingly vital category of SDN-specific synthetic traffic datasets. These synthetic datasets are meticulously crafted to more accurately reflect the unique characteristics, protocols, and traffic patterns found within SDN environments, thereby enhancing the realism and applicability of the trained models. These advanced techniques, particularly machine learning and deep learning, have collectively proven highly effective in identifying and neutralizing a wide spectrum of network threats, from volumetric DDoS attacks to subtle intrusions Ali et al., 2023.

3.3 Routing Optimization

Efficient and adaptive routing is a critical determinant of overall network performance, directly impacting key metrics such as end-to-end latency, aggregate throughput, and the efficient utilization of network resources. In the highly dynamic and constantly evolving SDN environments, where traffic patterns, network topology, and application demands can change rapidly, traditional static routing protocols are often sub-optimal. These conventional protocols lack the agility to dynamically adapt to real-time network conditions, frequently leading to congestion, underutilized links, and increased service degradation. Machine Learning, and particularly Reinforcement Learning, offers a powerful and intelligent mechanism for adaptive, real-time routing optimization.

Reinforcement Learning approaches such as Q-learning, SARSA, and Deep Q-Networks are extensively used to optimize path selection and traffic forwarding policies under dynamic network conditions. In this setup, an RL agent, typically integrated within or closely connected to the SDN controller, learns to choose optimal paths by actively interacting with the network environment. The agent continuously observes the current network state, gathering real-time telemetry data such as link loads, queue lengths at switches, propagation delays, and packet loss rates. Based on these observations, the agent takes routing actions, which involve instructing the SDN controller to program new forwarding rules for specific traffic flows or reconfigure existing paths. It then receives immediate or delayed feedback in the form of "rewards" based on the resultant network performance (e.g., a positive reward for reduced latency or increased throughput, a negative reward for congestion or packet loss). Through this continuous process of interaction, learning, and refinement, the RL agent gradually develops an optimal routing policy that maximizes long-term network performance objectives.

The benefits of ML-driven routing optimization in SDN are substantial and far-reaching:

- ***Lower Latency***: By dynamically identifying and selecting paths with minimal current and predicted congestion, RL-based routing can significantly reduce end-to-end latency, which is critical for real-time applications such as video conferencing, online gaming, and industrial control systems.
- ***Improved Load Balancing***: RL agents possess the intelligence to distribute traffic more evenly across all available network paths and links, effectively preventing the formation of "hot spots" (congested areas) and maximizing the overall utilization of network capacity. This dynamic load balancing ensures that no single link or device becomes a bottleneck.
- ***Adaptive Congestion Avoidance***: Unlike reactive traditional routing, RL can proactively identify nascent or impending congestion based on learned patterns and real-time network telemetry. It can then intelligently reroute traffic away from these potentially congested areas before they severely impact network performance, thereby maintaining smooth and reliable network operation.
- ***Enhanced Resilience***: By continuously monitoring network health and adapting routing decisions, RL can quickly respond to link failures or device outages, finding alternative paths and minimizing service disruption without human intervention.
- ***Multi-objective Optimization***: RL can be designed to optimize for multiple conflicting objectives simultaneously, such as minimizing latency while maximizing throughput and ensuring fairness among different traffic classes, providing a more holistic routing solution.

Table 2. ML Driven Routing Optimization

Benefit	**Description**
Lower Latency	Dynamically selects paths with minimal real-time and predicted congestion, reducing end-to-end delay—crucial for applications like video conferencing, gaming, and industrial control.
Improved Load Balancing	RL agents distribute traffic evenly across network paths, preventing hot spots and maximizing overall network capacity utilization.
Adaptive Congestion Avoidance	Proactively detects emerging congestion using learned patterns and telemetry, rerouting traffic before performance degradation occurs.
Enhanced Resilience	Continuously monitors network health and adapts routing to handle link failures or outages, ensuring minimal service disruption.
Multi-objective Optimization	Optimizes multiple goals (e.g., low latency, fairness, high throughput) simultaneously for a balanced and efficient routing strategy.

3.4 Resource Allocation and QoS Management

Ensuring robust Quality of Service and efficiently allocating network resources are paramount for supporting the increasingly diverse and stringent requirements of modern applications, especially for latency-sensitive, bandwidth-intensive, or mission-critical services. Machine Learning plays a pivotal role in these areas by enabling predictive capabilities, intelligent identification of congestion patterns, and adaptive allocation of crucial resources such as bandwidth, buffer space, and CPU cycles within the programmable framework of SDN.

ML models significantly enhance resource allocation through their ability to accurately predict future bandwidth demand and identify potential congestion patterns before they manifest as performance degradation. By analysing extensive historical traffic data, real-time network telemetry (e.g., flow statistics, port statistics, CPU utilization), and contextual information (e.g., time of day, day of week, scheduled events), ML models can forecast network resource requirements with high precision. Long Short-Term Memory networks, a specialized type of Recurrent Neural Network, are particularly effective for traffic forecasting due to their unique architecture that allows them to capture long-term dependencies and temporal dynamics inherent in time-series data. By accurately predicting future traffic loads and resource demands, the SDN controller can proactively adjust resource allocations, dynamically reserve bandwidth, or provision virtual network slices, thereby preventing performance bottlenecks and ensuring consistent QoS levels before any actual service degradation occurs.

Regression models are widely employed for tasks such as bandwidth prediction, where the objective is to forecast a continuous numerical value (e.g., network throughput in Mbps or Gbps). These models can incorporate a multitude of input features, including but not limited to the time of day, day of the week, observed user activity patterns, application types, and even external events, to make highly accurate predictions of future resource needs.

Furthermore, Reinforcement Learning is increasingly being applied for sophisticated QoS-aware flow scheduling and dynamic resource provisioning. An RL agent, operating within or alongside the SDN controller, can learn optimal policies for scheduling different traffic flows based on their specific QoS requirements (e.g., priority levels, latency tolerance, guaranteed bandwidth). The agent makes decisions regarding which flows to prioritize, how much bandwidth to allocate, or which path to use, receiving feedback in the form of rewards for successfully meeting QoS objectives and penalties for any violations. This continuous process of learning through interaction allows for highly adaptive and intelligent resource management decisions, ensuring that critical applications consistently receive the necessary resources while simultaneously maximizing overall network efficiency and utilization. RL can also dynamically adjust power consumption by turning off unused ports or links based on predicted traffic, contributing to green networking initiatives.

3.5 Fault and Failure Detection

Network faults and failures, ranging from subtle link degradations to catastrophic device outages or controller malfunctions, can severely disrupt network services, lead to significant downtime, and result in substantial financial losses and reputational damage. Machine Learning techniques are proving instrumental in enabling early prediction, rapid and accurate diagnosis, and proactive mitigation of these issues, thereby minimizing their impact on network operations and significantly enhancing network resilience.

Various ML techniques are effectively employed for robust fault and failure detection in SDN:

- **Classification Models for Failure Diagnosis:** Supervised classification models such as Decision Trees, Support Vector Machines, and Random Forests are widely used for accurately classifying the type and precise location of network failures based on comprehensive network monitoring data. By analyzing a rich set of metrics and alarms, including link status changes, packet loss rates, error counts on interfaces, device CPU and memory utilization, and log messages from switches and controllers, these models can pinpoint the root cause of an issue. For example, a decision tree might learn that a sudden increase in packet loss on a specific port, coupled with a decrease in throughput, indicates a faulty transceiver or a physical cable issue.
- **Anomaly Detection for Proactive Identification:** Unsupervised learning methods, such as Autoencoders, Isolation Forests, One-Class SVMs, and clustering algorithms like K-means, are highly effective for anomaly detection in network flow statistics and device telemetry. These techniques identify subtle deviations from established normal operational patterns that can indicate an impending fault or a nascent failure that might otherwise go unnoticed by rule-based systems. For instance, an unexpected, but minor, spike in retransmission rates on a particular link, or a gradual increase in latency that doesn't trigger traditional thresholds, could signal a degrading connection, an overloaded buffer, or a hardware malfunction in its early stages. Autoencoders, by learning the normal compressed representation of network behavior, can flag data points with high reconstruction errors as anomalies indicative of potential problems.
- **Graph Neural Networks for Topology-Aware Diagnosis:** Graph Neural Networks are emerging as an exceptionally powerful and advanced tool for fault diagnosis in complex and dynamic network topologies characteristic of SDN. GNNs inherently model the network as a graph, where nodes represent network devices (e.g., switches, hosts, controllers) and edges represent the physical or logical links between them. By processing information across this rich graph structure, GNNs can capture complex spatial dependencies and propagate failure signals across the entire topology. This capability enables more accurate, contextual, and topology-aware fault diagnosis, even in dynamic SDN environments where the network graph can frequently change due to reconfigurations or link failures. This allows for a holistic understanding of how localized failures or anomalies might propagate and impact global network performance, leading to more precise root cause analysis and faster recovery.

The capabilities for early and accurate fault prediction and detection provided by Machine Learning are instrumental in enabling proactive maintenance strategies, facilitating rapid fault isolation and recovery, and ultimately, significantly enhancing overall network resilience, availability, and service continuity. This predictive power minimizes downtime and optimizes operational expenditures.

4. Open Challenges

While the integration of Machine Learning within Software-Defined Networking offers transformative potential, its widespread adoption and optimal implementation are currently hindered by several significant open challenges. Addressing these challenges is crucial for realizing the full promise of intelligent and autonomous SDN.

4.1 Scalability of ML Models in Large SDNs

One of the most pressing challenges pertains to the scalability of ML models, particularly in the context of large-scale Software-Defined Networks. Modern networks, ranging from hyperscale data centres to vast enterprise and service provider networks, generate colossal volumes of data from millions of flows, devices, and events every second. This "big data" characteristic of network telemetry poses several issues for ML:

- ***Data Volume and Velocity*:** Processing and analysing such massive datasets in real-time, or near real-time, can overwhelm conventional ML architectures. The sheer volume makes data storage and retrieval challenging, while the high velocity demands extremely efficient processing pipelines.
- ***Computational Overhead*:** Training complex ML models, especially deep learning architectures, on these vast datasets requires immense computational resources. Furthermore, even during inference (making predictions), the computational demands can be significant. If ML inference engines are integrated directly into the SDN control plane, they risk becoming a critical bottleneck, delaying flow decisions and hindering the controller's ability to maintain real-time network control.
- ***Memory Constraints*:** Storing model parameters and feature sets for highly complex models that monitor millions of network entities can exceed the memory capacities of typical SDN controller platforms.

- ***Distributed Learning* Needs:** Centralized ML processing for massive SDNs is often impractical. There is a need for distributed or federated learning approaches that can process data closer to its source (e.g., on switches or regional controllers) to reduce data transfer overhead and improve latency.

Addressing scalability requires innovation in lightweight ML models, distributed learning architectures, and efficient data processing frameworks that can operate effectively under extreme data volumes and real-time constraints.

4.2 Lack of Realistic SDN Datasets

The development and evaluation of robust ML models heavily rely on the availability of high-quality, representative datasets. Unfortunately, a significant challenge in the ML-for-SDN domain is the pronounced scarcity of realistic and publicly available SDN-native datasets.

- ***Legacy Datasets*:** Many research efforts still rely on older, generalized network intrusion datasets like KDD99 or CICIDS2017. While valuable, these datasets often do not accurately reflect the unique characteristics, protocols, traffic patterns, and attack vectors prevalent in modern SDN environments. For instance, they may lack specific OpenFlow messages, controller-to-switch interactions, or SDN-specific vulnerabilities.
- ***Synthetic vs. Real-world Data*:** While synthetic datasets can be generated in SDN testbeds to simulate certain scenarios, they often fail to capture the full complexity, variability, and nuanced behaviors of real-world operational networks. The diversity of applications, user behaviors, and unpredictable anomalies in live networks are difficult to replicate accurately.
- ***Privacy and Proprietary Concerns*:** Real-world network traffic data from operational SDNs is often proprietary and contains sensitive user information, making it difficult for organizations to share publicly dueg to privacy regulations (e.g., GDPR, HIPAA) and competitive concerns. This severely limits the ability of researchers to train and validate ML models on truly representative data.
- ***Labelling Challenges*:** Even when raw data is available, the process of accurately labelling network traffic for specific ML tasks (e.g., classifying attack types, identifying performance anomalies) is labour-intensive, requires deep domain expertise, and is prone to errors.

The absence of rich, diverse, labelled, and publicly accessible SDN datasets hinders fair comparison of different ML techniques, slows down research progress, and limits the development of truly robust and deployable solutions.

4.3 Security and Adversarial Attacks on ML Models

While ML is a powerful tool for enhancing network security, the ML models themselves are not infallible and introduce new security vulnerabilities. The field of adversarial machine learning highlights how ML models can be manipulated by malicious actors:

- ***Adversarial Examples*:** Attackers can craft "adversarial examples"—slightly perturbed inputs that are imperceptible to humans but cause ML models to misclassify. For an SDN IDS, an attacker could subtly modify benign traffic to appear malicious, or conversely, modify malicious traffic to be classified as benign, thereby bypassing detection.
- ***Model Poisoning*:** During the training phase, attackers could inject malicious data into the training set, poisoning the model and causing it to learn incorrect or biased behaviours. This could lead to a compromised IDS that intentionally ignores certain attacks or misdirects legitimate traffic.
- ***Model Evasion*:** After a model is deployed, attackers can systematically probe the model to understand its decision boundaries and then craft inputs that evade detection. This is particularly relevant for anomaly detection systems where attackers might try to mimic normal behaviour to avoid being flagged.
- ***Data Privacy in Training*:** When ML models are trained on sensitive network data, there are risks of membership inference attacks (determining if a particular record was part of the training data) or model inversion attacks (reconstructing training data from the model parameters).

These vulnerabilities are critical because compromised ML models in SDN could mislead IDS, misroute traffic, or misallocate resources, leading to severe network disruptions or security breaches. Developing robust, verifiable, and adversarial-resilient ML models is a significant challenge.

4.4 Controller Overhead and Latency

The centralized nature of the SDN controller, while advantageous for global visibility, introduces the potential for bottlenecks when integrating complex ML tasks. Running resource-intensive ML models directly in the control plane can impose significant overhead and introduce unacceptable latency:

- ***Processing Latency*:** ML inference, especially for deep learning models, can be computationally intensive. If every flow or significant network event requires an ML prediction before a forwarding decision can be made, this processing latency can severely impact the real-time responsiveness of the controller. This is critical for high-speed networks and low-latency applications.

- ***Resource Contention*****:** Running ML algorithms alongside core controller functions (e.g., topology discovery, flow rule installation) can lead to resource contention. This can degrade the performance of the controller itself, making it slow to respond to network changes or install new flow rules.
- ***Scalability Limitations*****:** As network size and traffic volume increase, the number of ML inferences required also scales, exacerbating the overhead and latency issues.

Solutions often involve offloading ML tasks to dedicated processing units, distributing ML functionality across multiple controllers, or employing lightweight ML models and efficient inference techniques that can execute rapidly within the controller's operational constraints.

4.5 Interoperability and Standardization

The proliferation of diverse ML techniques and SDN platforms creates challenges related to interoperability and standardization.

- ***Heterogeneous ML Frameworks*****:** Researchers and developers often use different ML frameworks (e.g., TensorFlow, PyTorch, Scikit-learn), programming languages, and data formats. Integrating these disparate ML solutions with various SDN controllers (e.g., OpenDaylight, Ryu, ONOS, OVN) can be complex and labor-intensive.
- ***Lack of Standardized Interfaces*****:** There is a general lack of standardized APIs and protocols for seamless interaction between ML components and SDN controllers or data plane elements. This often results in ad-hoc integrations that are difficult to scale, maintain, and port across different SDN deployments.
- ***Vendor Lock-in*****:** Proprietary ML-driven SDN solutions can lead to vendor lock-in, limiting flexibility and innovation.
- ***Evaluation Metrics*****:** A lack of universally accepted benchmarks and standardized evaluation metrics for ML-for-SDN solutions makes it challenging to objectively compare the performance and effectiveness of different approaches.

Achieving greater interoperability and establishing industry standards for ML integration in SDN are crucial for accelerating deployment, fostering a broader ecosystem, and ensuring the long-term viability of these intelligent network architectures. This includes standardizing data formats, model deployment interfaces, and performance evaluation methodologies.

5. Future Research Directions

The field of Machine Learning in Software-Defined Networking is still in its nascent stages, with significant potential for innovation and advancement. Addressing the aforementioned challenges and exploring novel paradigms will be crucial for realizing truly intelligent, autonomous, and resilient networks. This section outlines several promising future research directions.

5.1 Federated and Distributed Learning for SDN

As network scale grows and data privacy concerns intensify, centralized ML models become increasingly impractical. Federated Learning and Distributed Learning offer compelling solutions. FL enables multiple decentralized edge devices or local controllers to collaboratively train a shared ML model without exchanging raw data, thus preserving data privacy and reducing communication overhead. Each local entity trains a model on its own data, and only model updates (e.g., weights or gradients) are aggregated at a central server.

- ***Benefits*****:** This approach mitigates centralized bottlenecks, enhances data privacy by keeping sensitive network traffic data localized, and reduces the bandwidth required for data transfer. It also allows for continuous learning from diverse network segments.
- ***Research Focus*****:** Future research will explore robust aggregation mechanisms, secure multi-party computation for privacy-preserving model updates, strategies for handling heterogeneous local datasets (non-IID data), and efficient deployment of FL in hierarchical SDN architectures. The resilience of FL to adversarial attacks and node failures also requires deeper investigation.

5.2 Lightweight ML Models for Real-Time SDN Control

The computational overhead and latency introduced by complex ML models pose significant challenges for real-time decision-making in high-speed SDN environments. Future research will focus on developing and deploying lightweight ML models that can operate efficiently with minimal computational resources and provide ultra-low latency inference.

- ***Techniques*****:** This includes techniques such as model compression (e.g., pruning, quantization, knowledge distillation), TinyML approaches, and hardware-accelerated inference (e.g., using FPGAs or specialized AI chips at network edges). The goal is to develop models that are small in size, consume less power, and execute quickly, making them suitable for deployment on resource-constrained network devices or within the critical path of the SDN controller.
- ***Research Focus*****:** Exploring new model architectures inherently designed for efficiency, investigating hardware-software co-design for ML acceleration in SDN switches, and developing methods for dynamically offloading model inference to appropriate compute resources (e.g., edge servers, dedicated ML accelerators) will be key. The trade-off between model accuracy and computational footprint will be a central theme.

5.3 GNN-Based Modeling for Dynamic Network Topologies

Traditional ML models often struggle to effectively capture the inherent graph structure and dynamic nature of network topologies. Graph Neural Networks offer a powerful paradigm for processing graph-structured data and are particularly well-suited for SDN. GNNs can model network entities (e.g., switches, hosts) as nodes and connections as edges, learning representations that incorporate both node features and topological relationships.

- ***Advantages*:** GNNs can intrinsically understand spatial relationships within the network, capture the impact of local changes on global network state, and adapt to dynamic topology changes. This makes them ideal for tasks like fault diagnosis, routing optimization, traffic prediction, and security analysis in dynamic SDN environments.
- ***Research Focus*:** Future work will involve developing novel GNN architectures specifically optimized for large-scale, evolving network graphs, exploring how GNNs can integrate real-time network telemetry (e.g., link loads, packet queues) as dynamic edge or node features, and applying GNNs to more complex tasks such as network synthesis and policy learning. Investigating the interpretability and explainability of GNN decisions in networking contexts will also be crucial.

5.4 Explainable AI for SDN Decision Transparency

The "black-box" nature of many advanced ML models, particularly deep learning, poses a significant challenge for their adoption in critical networking applications like security and QoS management. Network operators need to understand *why* an ML model made a particular decision (e.g., why a specific flow was flagged as malicious, or why a particular routing path was chosen). Explainable AI aims to make ML models more transparent, interpretable, and understandable.

- ***Importance*:** XAI is necessary for building trust in ML-driven SDN systems, debugging models when they make incorrect predictions, complying with regulatory requirements, and enabling network operators to gain insights into complex network behaviors detected by AI.
- ***Research Focus*:** Future research will concentrate on developing XAI techniques tailored for networking domain, such as generating human-understandable explanations for anomaly detection alerts, providing justifications for routing decisions, and visualizing the influence of different network features on ML predictions. Integrating XAI into existing SDN monitoring and management dashboards will also be a key area.

5.5 ML-Integrated Intent-Based Networking

Intent-Based Networking represents the next evolutionary step beyond SDN, allowing network administrators to define high-level business intents (e.g., "ensure low latency for all financial transactions") rather than specifying low-level configurations. ML is a natural fit for bridging the gap between high-level intent and low-level network actions.

- ***Synergy*:** ML can be used to translate human-readable intents into network policies, validate if the current network configuration meets the intent, and continuously monitor the network to ensure intent assurance. If deviations are detected, ML can trigger autonomous corrective actions.
- ***Research Focus*:** Key research areas include developing natural language processing techniques for intent translation, using RL for policy generation and optimization based on intents, applying anomaly detection to identify intent violations, and creating feedback loops where ML models learn from intent failures to refine future policy deployments. The integration of XAI will be critical for providing transparency on how intents are translated and assured.

6. Conclusion

Machine Learning has emerged as a profoundly impactful and indispensable force for significantly enhancing the capabilities and intelligence of Software-Defined Networking across a multitude of domains. Its integration provides sophisticated solutions in critical areas such as robust security mechanisms, intelligent traffic management, adaptive routing optimization, proactive resource allocation, and highly efficient fault tolerance. By leveraging ML's power to analyze vast datasets, learn complex patterns, and make autonomous decisions, SDN environments can evolve into more adaptive, resilient, and intelligent infrastructures capable of meeting the rigorous demands of modern digital ecosystems.

While the synergistic integration of ML into SDN presents immense promise, it is also accompanied by a distinct set of challenges. These include the considerable computational overhead associated with running complex ML models, particularly in large-scale deployments, the persistent scarcity of realistic and properly labeled SDN-native datasets for training and validation, and the emerging vulnerabilities related to adversarial attacks on ML models themselves. Furthermore, issues such as managing controller latency under heavy ML workloads and achieving greater interoperability and standardization across diverse ML frameworks and SDN platforms remain significant hurdles.

Despite these challenges, the field continues to evolve rapidly, driven by innovative research and technological advancements. Emerging techniques such as Graph Neural Networks offer new ways to model and understand complex network topologies, while federated learning promises to overcome data privacy and scalability limitations by enabling distributed model training. The development of lightweight ML models and the emphasis on Explainable AI are crucial for ensuring real-time performance, building trust, and providing transparency in ML-driven network decisions. Moreover, the integration of ML within Intent-Based

Networking heralds a future where networks can autonomously interpret high-level business objectives and self-configure to achieve them.

In conclusion, the journey towards fully autonomous and intelligent networks powered by the convergence of SDN and ML is an exciting and ongoing endeavour. By systematically addressing the identified challenges and vigorously pursuing these promising future research directions, we can unlock the full potential of this powerful synergy, paving the way for next-generation network architectures that are inherently smarter, more secure, more efficient, and supremely resilient in the face of ever-increasing complexity and dynamic operational demands. The continued advancement in this interdisciplinary field will undoubtedly shape the future of network management and operations.

7. References

[1] Zhang, J., Ye, M., Guo, Z., Yen, C.-Y., and Chao, H.-J, "CFR-RL Critical flow rerouting using reinforcement learning for network-wide traffic engineering". arXiv. arXiv, 2020.

[2] Amin, R., Rojas, E., Aqdus, A., Ramzan, S., Casillas-Pérez, D., & Arco, J. M, "A survey on machine learning techniques for routing optimization in SDN". IEEE Access, 9, 104582–104611. NASA ADS, 2021.

[3] Bouzidi, E. L. H., Outtagarts, A., Langar, R., & Boutaba, R, "Deep Q-Network and traffic prediction-based routing optimization in software-defined networks. Journal of Network and Computer Applications", 193, 103181. ACM Digital Library, 2021.

[4] Quach, H. N., & (co-authors), "Survey on reinforcement-learning-based efficient routing in SDN", (Conference/Report). kyungbaekkim.jnu.ac.kr, 2020.

[5] Ferriol-Galmés, M., Badia-Campos, I., Suárez-Varela, J., Barlet-Ros, P., & Cabellos-Aparicio, A, "RouteNet-Erlang: A graph neural network for performance evaluation of computer networks". arXiv / conference paper. arXiv, 2022.

[6] Rusek, K., Suárez-Varela, J., Almasán, P., Barlet-Ros, P., & Cabellos-Aparicio, A, "RouteNet: Leveraging graph neural networks for network modeling and optimization in SDN". (Foundational GNN model used widely in 2020–2024 works). ResearchGate+1, 2019.

[7] Faezi, S., & Shirmarz, A, "A comprehensive survey on machine learning using in software-defined networks (SDN)". Human-Centric Intelligent Systems, 3(10). https://doi.org/10.1007/s44230-023-00025-3. SpringerLink, 2023.

[8] Serag, R. H, "Machine-learning-based traffic classification in software-defined networks. Electronics (MDPI)", 13(6), 1108. MDPI, 2024.

[9] Dhamala, B. K., & (co-authors) "Performance evaluation of graph neural network-based models for network QoS prediction (MDPI / Sensors)", MDPI, 2024.

[10] Shin, D. J., & (co-authors), "Deep reinforcement learning-based network routing and related applications in SDN" Applied Sciences. MDPI, 2021.

[11] Ye, M., & (co-authors), "Federated traffic engineering with supervised learning in multi-region SDN (ICNP/related work)", ICNP 2021, 2021

[12] Badia-Sampera, P., Suárez-Varela, J., Barlet-Ros, P., & Cabellos-Aparicio, A, "RouteNet family (RouteNet / RouteNet-E) and subsequent improvements — GNNs for routing & performance modeling", UPCommons+1, 2019-2022.

[13] Rusek, K., et al., "Graph Neural Networks for routing optimization: surveys and follow-ups (comprehensive overviews of GNN → networking)". BIMSA, 2020-2022.

[14] Boryło, P., & (co-authors), "SDNRoute: Proactive routing optimization in software defined networks" (ScienceDirect / Computers & Electrical Engineering or similar). ScienceDirect, 2024.

[15] Khalid, H. Y. I., & (co-authors), "A survey on the latest intrusion detection datasets for SDN and ML applications — overview of SDN-specific IDS datasets and challenges", 2024.

[16] Wang, T., Li, Y., Xu, K., & Xu, Y, "A deep learning-based intelligent routing strategy for software-defined networks", IEEE Access, 9, 108276–108287, 2021.

[17] Mousavi, S. M., & St-Hilaire, M, "Anomaly detection in software-defined networks using machine learning techniques: Survey and challenge", Computer Networks, 204, 108693, 2022.

[18] Liu, Y., Tang, F., Kawamoto, Y., Kato, N., & Jiang, T, "Deep reinforcement learning for dynamic routing in software-defined networking: A comprehensive review. IEEE Communications Surveys & Tutorials", 24(1), 283–317, 2022.

[19] Khan, M. A., Karim, A., & Akhunzada, A, "Machine learning-based DDoS attack detection for software-defined networks: A systematic review. Journal of Network and Computer Applications", 216, 103615, 2023.

[20] Alsaeedi, A., Alrasheed, H., & Alahmadi, A, "Traffic prediction and QoS optimization in SDN using LSTM-based deep learning models. IEEE Transactions on Network and Service Management", 21(2), 155–168, 2024.