Pak. j. life soc. Sci. (2025), 23(1): 1738-1758 E-ISSN: 2221-7630;P-ISSN: 1727-4915

Pakistan Journal of Life and Social Sciences

www.pjlss.edu.pk

https://doi.org/10.57239/PJLSS-2025-23.1.00135

RESEARCH ARTICLE

Brief Review of Using Machine Learning for Traffic Engineering in Software-Defined Networks

Entisar H. Khalifa[1*], Shimaa A. Ahmed[2], Nawaz Majid[3], Faroug A. Abdalla[4], Asma Mohamed[5], Safwa E. I.Yagoub[6], Lamiaa Galal Amin[7]

[1,3,4] Department of Computer Science, College of Science, Northern Boarders University, Arar, Saudi Arabia

[2] Department of Electrical Engineering, College Of Engineering, Northern Boarder University

[5] Department of Mathematics, College of Science, Northern Boarders University, Arar, Saudi Arabia

[6,7] Physics Department, Faculty of Science, Northern Boarder University, Arar, Saudi Arabia.

ARTICLE INFO

Received: Nov 21, 2024

Accepted: Jan 7, 2025

Keywords

Traffic Engineering

Software Defined Networking

Machine Learning

Quality of Service

Traffic Forecasting and Management

***Corresponding Author:**

Entisar.Osmab@nbu.edu.sa

ABSTRACT

This review is a brief exploration of deploying Machine Learning (ML) in Traffic Engineering (TE) for Software Defined Networks (SDN). SDN changes traditional network management by separating the control plane from the data plane, opening up new possibilities for flexible and adaptive traffic control. As we show, TE in SDNs can optimize network performance by using resources more efficiently, cutting down on latency, and reducing congestion—all while responding to real-time conditions to maintain high Quality of Service (QoS). However, taking full advantage of these benefits requires advanced algorithms and real-time data analysis, which can be computationally demanding. TE also relies on having accurate, up-to-date information about the network. Meanwhile, ML is making SDNs more effective by integrating with technologies like Edge Computing, Network Function Virtualization (NFV), and the Internet of Things (IoT). This combination enables real-time analytics, quick decision-making, intelligent routing, load balancing, and stronger security. Still, these integrations bring fresh challenges in scalability and interoperability, meaning we need major investments in both infrastructure and expertise. Even with all the progress made so far, several hurdles remain. These include issues with scaling up, maintaining robust security, and making split-second decisions in real-time. Looking ahead, future research should concentrate on autonomous networking, energy-efficient ML techniques, and hybrid ML solutions, aiming to reach new heights in network security and performance.

INTRODUCTION

For a long time, the advancement of network structures has been significantly affected by the approach of SDN. SDN contribute to classic network by decoupling the network's control plane from its data plane, centralizing organized administration, and giving exceptional adaptability [1]. This design encourages energetic, organized arrangement and coordination but presents modern challenges in successfully overseeing and optimizing organized activity [2]. Conventional TE strategies, which depend on inactive steering and manual setup, frequently drop brief in adjusting to advanced systems' dynamic changes and complexities [3]. Recent studies show that traditional Traffic Engineering (TE) methods can be inefficient and lead to performance problems, especially in large-scale networks with diverse and unpredictable traffic patterns [4]. As demands for high performance and reliable network management continue to grow, the importance of network engineering within Software-Defined Networks (SDNs) becomes even more evident. SDN's design centralizes control of network resources, enabling more sophisticated and adaptable management strategy [5]. SDN's effectiveness in Traffic Engineering depends on harnessing cutting-edge

technologies like Machine Learning, which has shown significant promise for enhancing network operations. [6].

Machine Learning (ML) has the potential to revolutionize Traffic Engineering (TE) by offering predictive analytics, real-time optimization, and automated decision-making. [7]. or example, ML algorithms can process massive volumes of traffic data to predict congestion and adjust routing strategies on the fly, boosting network efficiency and reducing latency. Research has also shown that ML-driven traffic management can enhance network performance, leading to higher throughput and lower packet loss. [8-10]. Despite these advancements, combining ML with SDN for network engineering remains in its early stages, as ongoing studies continue to tackle challenges like data quality, model accuracy, and scalability. [11-12]. Exploring the combined power of ML and SDN is crucial for enhancing network performance and efficiency. By understanding how ML can expand SDN's ability to manage and optimize networks, we can achieve more intelligent, versatile, and adaptable network solutions. This paper aims to provide a comprehensive review of current research and practices in this field, evaluate the effectiveness of various ML strategies, and identify both the opportunities and challenges that lie ahead. Through this study, we offer valuable insights into ML-enhanced Traffic Engineering (TE) in SDNs, highlight potential advancements, and propose directions for future research [13].

The following sections show the structure of this review paper. Section 2 provides foundations of SDN and TE, building up an establishment for understanding their parts and the integration challenges. Section 3 investigates the transformative potential of ML in traffic networks inside SDNs, highlighting key progressions and applications. This will be discussed after examining the current state of investigation, centering on different ML strategies and their effect on network execution. Section 4 highlights the applications of ML in TE through cases.

Moreover, section 5 addresses the challenges confronted when combining ML with SDN, including issues related to information quality, demonstrated precision, and adaptability. Section 6 bolsters later propels and patterns within ML within SDNs. In conclusion, the paper will conclude with a discourse of future inquiry relating to ML for TE in SDN, giving experiences and direction for progressing ML-enhanced network management in SDN environments.

2. FOUNDATIONS OF SOFTWARE DEFINED NETWORKING (SDN)

2.1 SDN architecture

SDN is an inventive network design that decouples the control plane from the data plane, permitting centralized network activity management [14]. In traditional networks, both the control plane and the data plane are integrated into the same devices, like switches and routers. [15], SDN isolates these planes to upgrade adaptability, versatility, and control. In SDN architecture, the SDN Controller has a global view and manages overall network behavior by issuing instructions and gathering data. (See Figure 1) [16].

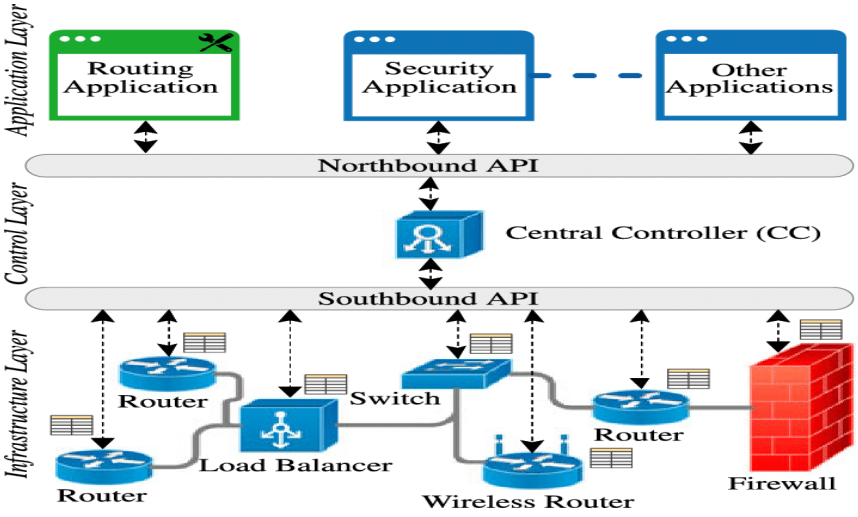

Figure 1: Software-defined network architecture [16]

SDN applications run on the best of the controller and characterize the network's behavior. These applications can incorporate organized administration, observation, security administration, and activity designing arrangements [17]. These devices connect to a central SDN controller, which enforces network policies and ensures optimal performance. Traditionally, each device's control plane decides how traffic is routed. However, in an SDN framework, all those decisions are made by the centralized controller—often called the "brain" of the network—where it sets policies, directs traffic, and makes top-level decisions. [18]. In addition, the data plane—often called the forwarding plane—is in charge of actually moving data packets around, following the instructions set by the control plane. You'll typically find it implemented in devices like switches, which abide by the controller's rules. [19]. The sepration of the control and data planes in SDN offers critical focal points, such as expanded network agility and programmability. However, it also presents challenges, especially in terms of security and quality of service [20]. Centralizing the control plane simplifies network operations but also increases reliance on a single controller. Meanwhile, the shift to software-driven management calls for careful planning to ensure smooth performance. Striking the right balance among these factors is key to a successful SDN deployment. [21].

2.2 Traffic engineering in SDNs

2.2.1 Definitions and goals of traffic engineering

Applying ultimate Traffic Engineering (TE) scheme in SDNs, make networks run more smoothly and efficiently. [22]. The main goal is to carefully manage the flow of data throughout the network, minimizing congestion, reducing latency, and making the best use of available resources. [22, 23]. Traffic Engineering (TE) works to boost overall Quality of Service (QoS) by dynamically adjusting how data moves through the network. It plays a vital role in SDNs, offering more fine-grained control over resources. Leveraging SDN's centralized management, traffic can be rerouted on the fly to relieve congestion and make the most of available resources.

[24]. However, this energetic process requires advanced algorithms and real-time information examination, which can be done computationally completely. Moreover, the success of TE in SDNs depends intensely on the precision of the network state information accessible to the controller [25]. Wrong or nonreal time information and data can lead to imperfect decisions, possibly compounding congestion or making bottlenecks.

2.2.2 Techniques and strategies

Compared to traditional routing protocols like IS-IS, which rely on fixed link costs, SDN offers a major step forward through dynamic network state management. Here, controllers can update link values in real time, making way for new routing strategies that optimize resource use and enhance QoS. Various TE methods have emerged, highlighting SDN's ability to swiftly adapt to changing network conditions. For example, Google's B4 architecture dynamically adjusts bandwidth and reroutes traffic to handle failures, ensuring reliable performance across its wide area network. [26, 27].

Techniques such as Hedera and DevoFlow have introduced effective methods for managing large data flows in data centers. Hedera identifies "elephant flows" that consume significant bandwidth and reroutes them to less congested paths [28]. Meanwhile, DevoFlow minimizes interactions between the control and data planes, allowing switches to handle routing decisions locally for smaller flows, thus reducing overhead [29]. However, both methods have their downsides: Hedera's periodic polling can lead to high resource utilization, while DevoFlow relies heavily on detecting elephant flows, which may not always be efficient [29, 30]. Mahout offers a promising alternative by integrating end-host modifications for flow detection, achieving quicker responses with less overhead, yet it necessitates changes to the end-host architecture [31].

Emerging frameworks like Atlas leverage machine learning for application-aware traffic classification, demonstrating a shift toward intelligent network management [32]. By accurately classifying traffic based on application types, networks can enforce policies that enhance performance and user experience. Tag-based classification further reduces processing overhead by marking packets at the edge. Additionally, multi-agent systems are being explored to improve task scheduling and resource management in distributed networks. The MSDN-TE mechanism exemplifies a proactive approach to traffic engineering by dynamically selecting optimal paths to

avoid congestion [33]. Collectively, these advancements reflect a trend toward more adaptive, efficient, and intelligent network management in SDN, highlighting both the potential benefits and the ongoing challenges in achieving seamless integration and performance optimization. Table 1 give examples of traffic engineering techniques in the SDN network.

Table 1: Examples of traffic engineering techniques in SDN

Technique	Description	Routing	Comments
B4 [26]	This approach employs centralized traffic engineering that operates above existing routing protocols and uses the Min-Max fairness method to allocate resources.	It utilizes a hashed Equal-Cost Multi-Path (ECMP) to distribute the load across multiple links.	TE service can be halted, allowing packets to be routed using a short path forwarding method.
Hedera [28]	This technique identifies elephant flows at edge switches, marking a flow as an elephant if it reaches 10% of the network interface cards (NIC) bandwidth threshold, with updates every 5 seconds.	It leverages a comprehensive view of the network to identify optimal, conflict-free paths for these large flows	This method achieves a throughput of 15.4 b/s and enhances the network's bandwidth bisection compared to ECMP; however, the periodic polling may lead to high resource consumption on switches
DevoFlow [29]	This method marks flows as elephant flows when they exceed a size threshold of 1–10 MB, detected at edge switches.	It employs wildcard OpenFlow rules and a static multi-path routing technique for traffic forwarding.	A CLOS network can enhance throughput by up to 32%.
Mahout [31]	Mahout identifies elephant flows at end-hosts via a shim layer, with a threshold of 100 KB, using in-band signaling to notify the controller.	It finds the optimal path for elephant flows while routing smaller flows with ECMP and assesses congestion by gathering statistics from switches.	This technique can identify elephant flows in approximately 1.53 ms and offers 16% better bisection bandwidth than ECMP.
MicroTE [34]	This approach detects elephant flows at end-hosts and calculates the average traffic matrix between top-of-rack (ToR) switches. If the traffic deviates from this average by 20%, it is considered predictable.	It utilizes short-term predictability for multi-path routing, while other flows are managed by an ECMP strategy with a heuristic threshold.	When traffic patterns are predictable, performance is near-optimal; otherwise, it behaves similarly to ECMP.
Atlas [32]	Atlas uniquely classifies applications using the C5.0 machine learning model and requires users to install agents on their devices to gather data for training.	It routes flows based on specific application requirements and network conditions.	It achieves approximately 94% accuracy but needs extensions to the OpenFlow protocol.
MSDN-TE [33]	This mechanism collects information about the network state and evaluates the load on paths to distribute traffic across multiple routes.	It dynamically chooses the most efficient shortest paths among available options.	MSDN-TE outperforms other forwarding methods like Shortest Path First, reducing download times by 48%

From Tab1e 1, the techniques highlighted represent various approaches to traffic engineering in SDN. **B4** utilizes centralized traffic engineering with Min-Max fairness and hashed ECMP to efficiently manage resources, allowing fallback to short path forwarding if needed. **Hedera** identifies large elephant flows at edge switches, optimizing their routing through a global network view while facing potential resource overhead due to periodic polling. **DevoFlow** employs wildcard rules to enhance throughput for elephant flows in CLOS networks, while **Mahout** focuses on end-host detection of these flows, achieving rapid identification and improved bisection bandwidth. **MicroTE** and **Atlas** utilize traffic predictability and machine learning for application-specific

routing, respectively, with **MSDN-TE** dynamically selecting optimal paths, demonstrating significant improvements in performance and download times compared to traditional methods.

Load balancing

Load balancing is crucial for traffic engineering in SDN environments, ensuring that network traffic is distributed evenly across multiple paths or resources so no single path or device gets overloaded. [35]. This method is basic for optimizing network performance, moving forward asset utilization, and guaranteeing a high QoS. SDN permits real-time checking and alteration of network traffic. Load balancing can powerfully disperse activity based on current network conditions, making the network more responsive to changes in requests or startling traffic spikes [36]. This flexibility is essential for preserving service quality in ever-changing network conditions. By distributing traffic evenly, load balancing ensures that all available paths and resources are used efficiently. [37]. Beyond simply preventing overload on individual devices or links, this approach also helps maximize the entire network's capacity. As shown in Figure 2, an OpenFlow switch receives traffic flows from the SDN environment and matches them against its internal flow data. [38]. When existing flow entries match, data is forwarded directly. If there's no match, the packet header goes to the load balancer and SDN controller. From there, the SDN controller creates new flow tables, collaborates with the load balancer to find the best path, and exchanges periodic updates [38].

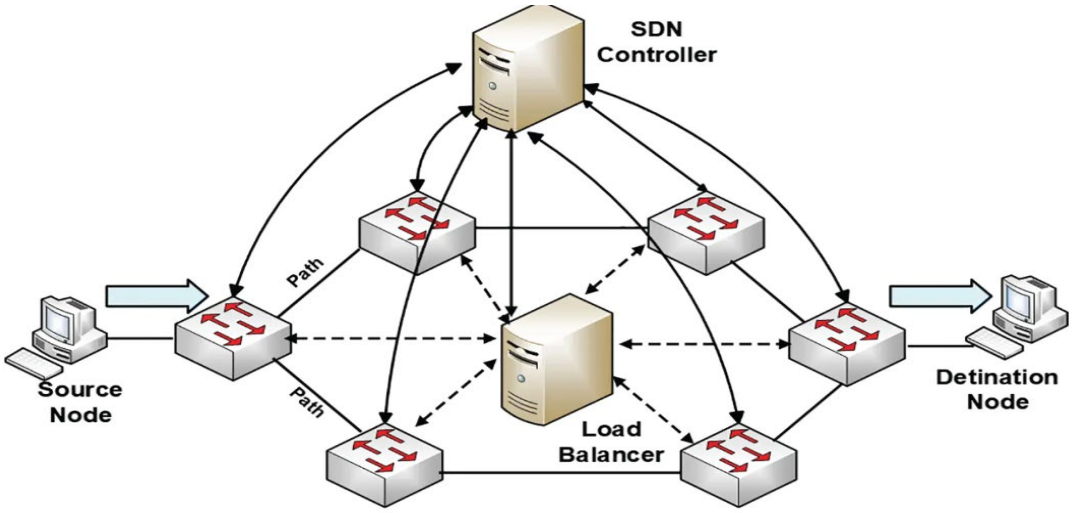

Figure 2: Load balancing in SDN [38]

Load balancing improves network quality by distributing traffic across multiple paths. If one path fails, the controller can quickly redirect traffic to other available routes, reducing the risk of service interruptions. [39]. In conventional networks, personal devices often manage load balancing, which can lead to complicated setups. In an SDN environment, the centralized controller takes charge of load balancing, simplifying both configuration and management. This centralization also enables more advanced algorithms, which further enhance load balancing effectiveness. [40]. In addition, SDN's programmability permits advanced load-balancing strategies, but it can pose challenges to versatility. The controller processes large volumes of data to monitor traffic and make decisions, which can sometimes cause delays in managing network traffic.

[41]. Successful load balancing requires advanced algorithms that consider components like traffic loads and interface capacities. However, relying on a central controller can be risky—if it fails or becomes overloaded, the entire load-balancing system may collapse. Malicious actors could also target the load-balancing process in SDN, which highlights the need for robust authentication and encryption for control messages. [42].

Path optimization

In SDN, path optimization is a vital issue of traffic designing that emphasizing the selection of the most efficient routes for data to travel across the network. [42]. It relies on advanced algorithms that evaluate factors like latency, available bandwidth, and overall network conditions. [43]. The

main goal of path optimization is to cut down on delays, reduce congestion, and use network resources efficiently. By ensuring data packets travel along the shortest and least congested paths [44], it improves network performance through lower latency and higher throughput—both of which are crucial for real-time services like video conferencing and online gaming [20]. Path optimization algorithms continuously monitor traffic conditions and can adjust routes on the fly to avoid congested or faulty links. This kind of real-time adaptability is especially important in large, complex networks where traffic patterns can shift rapidly [25].

As shown in Figure 3, the network includes two main types of devices: Provider Edge (PE) switches and Core Routers (CR). In a typical MPLS setup [45], PE switches can initiate or terminate Label Switched Paths (LSPs), while both PE and CR devices can exchange labels in the middle of a path. PE switches connect to Customer Edge (CE) switches, which act as external traffic sources and destinations.Following the OSHI architecture proposed in [46], both PE and CR devices operate as hybrid IP/SDN nodes. There's no need for a separate MPLS control plane in these devices; instead, an SDN-based approach uses the OpenFlow protocol to program the flow tables of what's called an OpenFlow Capable Switch (OFCS). This allows dynamic adjustments to the network's configuration without requiring traditional MPLS control plane mechanisms [46].

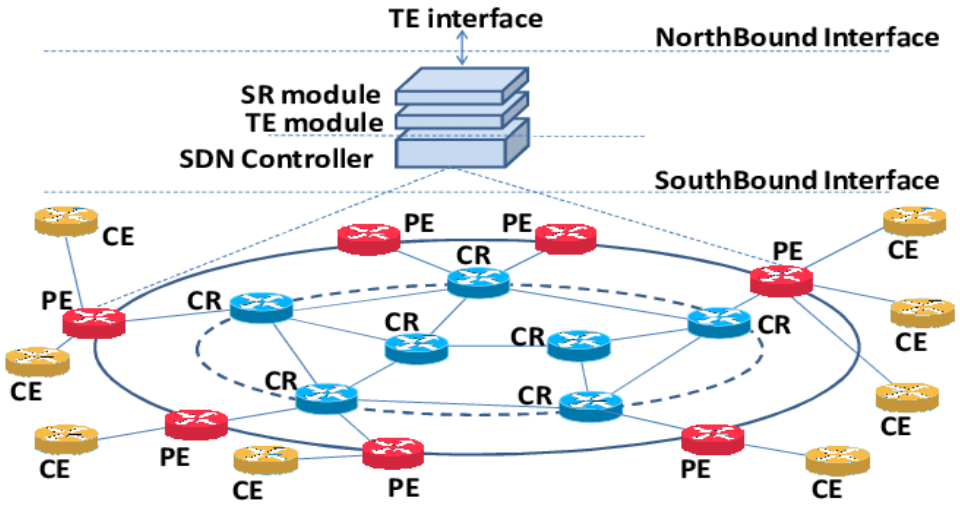

Figure 3: Traffic engineering (TE) path optimization [45]

SDN ensures that all available network resources are used efficiently by optimizing paths based on the current network conditions. This approach prevents some links from becoming overused while others remain underutilized, resulting in a more balanced and effective use of the network's capacity [47]. Additionally, path optimization can be customized to meet specific Quality of Service (QoS) requirements by prioritizing certain types of traffic over others. For example, latency-sensitive applications can be routed through the fastest available paths, while less critical traffic can take longer routes. This flexibility is essential for maintaining high service quality in environments with diverse traffic types [47].

However, calculating the best paths is complex and must consider factors like latency, bandwidth availability, link failures, and routing needs [48]. As networks grow, the computational demands on the SDN controller increase, which can lead to scalability challenges and less accurate routing decisions. If the controller doesn't have an accurate view of the network, it can cause increased congestion or higher latency. Additionally, the training process for these systems can introduce delays and temporary performance drops [49]. Moreover, path optimization can become a target for malicious attacks, potentially leading to data breaches or disruptions in service.

3. MACHINE LEARNING FUNDAMENTALS

3.1 Introduction to machine learning

Machine Learning (ML) is a subset of AI that focuses on creating systems that can learn from data and make decisions based on that information [50]. Instead of being explicitly programmed to

perform a task, ML models improve their performance through exposure to data. Figure 4 illustrates how ML-related models have evolved over time [51]. ML is particularly effective for drawing conclusions from large, representative datasets. These techniques are designed to identify patterns and extract hidden information from data, making them well-suited for problem-solving in SDNs. For instance, in SDN, a classification problem could be set up to detect anomalous activity. The most common ML algorithms used for traffic classification in SDNs are shown in Figure 4 [51].

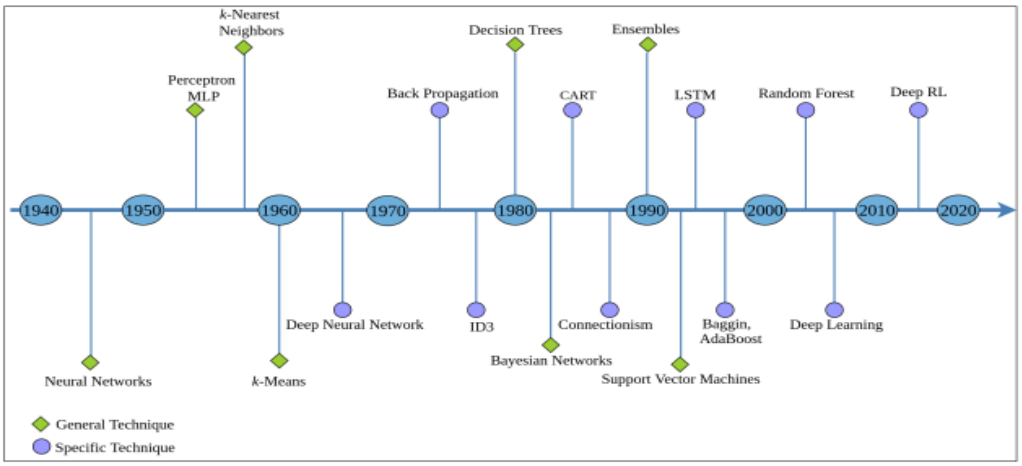

Figure 4: ML models development over the Years [51]

3.1.1 Basic ML algorithms and models

Linear regression

Is a simple algorithm commonly used for regression tasks, where the goal is to predict a continuous output based on input features [52]. The model assumes a straight-line relationship between the inputs and the output. Its simplicity and ease of interpretation make it a popular choice for basic problems. However, it assumes a linear relationship between variables, which may not always reflect real-world data [53]. As a result, its performance can suffer in more complex situations where the relationships between variables are non-linear. Figure 8 illustrates how the linear regression algorithm works and its flowchart [54].

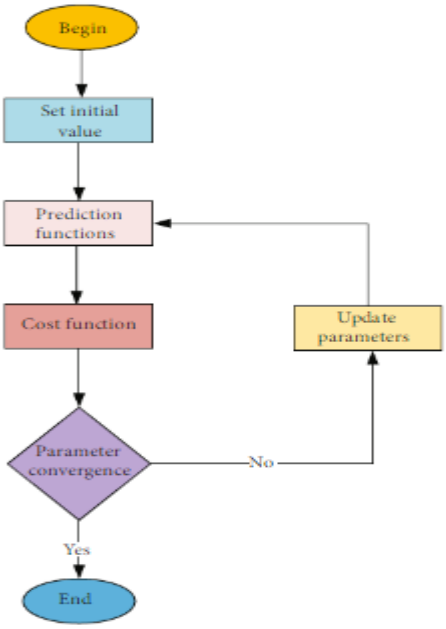

Figure 8: Linear regression implementation flow chart [54]

The linear regression model is designed as

$$h(x)=w_1 x_1+w_2 x_2+w_3 x_3+\cdots+ w_n x_n + b_x \qquad \text{Eq.1}$$

Equation 1, where w is the weight function, b is the deviation function, and x is the input variable, outlines the components of the direct relapse show. When there is one fair variable, the straight regression model is ordinarily a straight line in a plane or maybe a straight line. A model is a plane in space when there are two factors; in addition, the model will be more highly dimensional when there are more factors.

Decision Trees are a tree-based model used for both regression and classification tasks. They work by splitting data into subsets based on the values of input attributes, creating a decision tree structure [55]. Decision trees are intuitive and easy to visualize, making them useful for understanding how decisions are made. However, they can easily overfit the training data, which leads to poor generalization when applied to new, unseen data. To address this, ensemble methods like random forests are often used, though they add complexity to the model [56]. Additionally, decision trees create rules from training datasets using repeated splitting to categorize traffic [57]. They are commonly applied in cybersecurity to detect denial-of-service (DoS) attacks by analyzing factors like traffic rate, size, and duration. Autonomous vehicles, for example, can detect command injection attacks by analyzing data volume, network flow, and CPU usage [58], as shown in Figure 9. This method is popular because of its straightforward nature and ability to analyze traffic in real-time. Designers can easily identify unusual traffic patterns and classify them as normal or potential attacks. Once a set of rules is established, the AI system can provide immediate alerts if suspicious activity is detected.

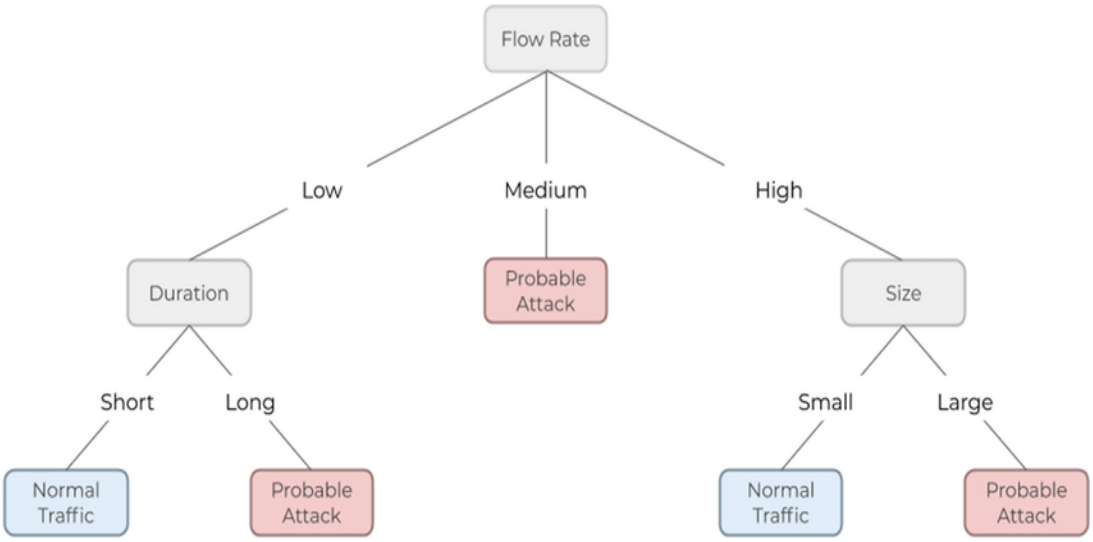

Figure 9: Decision trees model [58]

Support vector machines (SVM) are classification algorithms that aim to find the optimal hyperplane to separate different classes in the feature space [59]. SVMs are particularly effective in high-dimensional spaces and are widely used in areas like bioinformatics and text classification. They excel in binary classification tasks and can manage non-linear data by using kernel functions (as shown in Figure 10) [60]. However, SVMs can be computationally expensive, especially with large datasets, and their performance is heavily influenced by the choice of kernel and regularization parameters. SVMs also face challenges in multiclass classification, often requiring approaches like one-vs-one or one-vs-all, which can add complexity to the model [61].

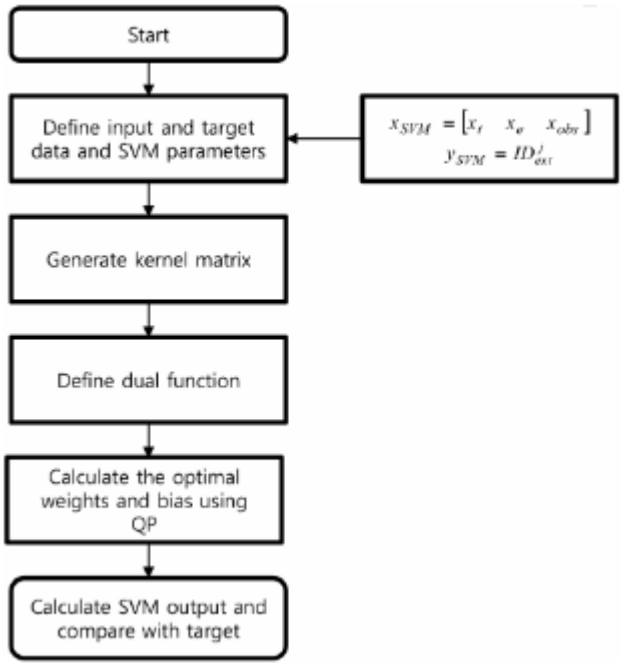

Figure 10: Flow chart of SVM training process [60]

In 1995, Vapnik introduced the Support Vector Machine (SVM) method, an ML theory that uses an algorithm to find the hyperplane that maximizes the margin [62]. SVMs are used in both regression and classification tasks, and when applied to regression problems, they are known as Support Vector Regression (SVR). An SVM defines the regression function, f(Xsvm), in such a way that the target, Vsvm, falls within a specified range.

$$f(xsvm) = \hat{}ysvm = w^T xsvm + b \quad \text{Eq. 3}$$

$$f(xsvm) - \varepsilon \leq ysvm \leq f(xsvm) + \varepsilon, \varepsilon > 0$$

Where xsvm is the input that contains [x_1 x_e x_{obs}], and w^T is the transposed weighting matrix; ysvm is the target that signifies the true ionosphere delay within the region of extrapolation, and xε is the acceptable error level for ysvm. In many practical cases, ysvm is not in the range of (f (xsvm)−ε, f (xsvm)+ε), and ysvm is frequently adjusted to the range of (f (xsvm)− ε, f (xsvm) + ε), where ε is a slack variable. The optimal regression function is determined when the slack variable total magnitude, $\sum_i \xi_i$ is reduced. Besides, the distance between the support vector and f (xsvm) should be maximized; this distance is called the margin, and the margin may also be minimized [60]. Consequently, the optimal regression function minimizes || w || and ξ to achieve the maximum margin.

3.2 ML techniques for traffic engineering in SDN

ML techniques are progressively becoming functional for traffic engineering in SDN to progress management, security, and performance.

3.2.1 Classification

Classification algorithms are used to categorize network traffic, identify intrusions, and detect types of network attacks [5]. In networking, classification enables real-time, automated decision-making, such as detecting malicious activity or classifying data packets based on priority [57]. For example, in smart city applications, to ensure Quality of Service (QoS), network traffic, control estimation, security, and resource management need to be organized into specific categories, a process known as traffic classification [63]. The most basic method for traffic classification involves mapping application traffic to specific port numbers [64]. However, since many applications use dynamic

port numbers, this approach can lead to incorrect classifications. Alternatives to port-based methods, such as payload-based algorithms, classify traffic by analyzing the packet payloads [64].

Due to privacy and security concerns, the packet payload is not accessible when the traffic is encrypted. As a result, ML and DL-based methods can be used to address the limitations of traditional approaches (Figure 11) [65]. Ren, Gu, and Wei [66] proposed a Tree-RNN model to classify network traffic into 12 distinct categories. This deep learning model features a tree structure, where each node corresponds to a set of classes, allowing the large classification task to be broken down. Additionally, Lopez-Martin et al. [67] proposed an integrated CNN and RNN-based network to classify traffic from IoT services and devices. Unlike traditional ML methods, this approach eliminates the need for feature selection, automatically extracting complex patterns from the input data.

Figure 11, which demonstrates a deep learning classifier for network traffic classification, can be directly related to ML applications in SDN for traffic engineering [65]. In this scenario, network traffic from IoT devices is first analyzed by extracting important features like packet payload, inter-arrival time, and source/destination ports. The deep learning classifier is capable of distinguishing between various types of traffic, such as web browsing, video streaming, or social networking, and can categorize them into priority levels (low, medium, high).

For instance, in SDN, when the network detects high-priority traffic, such as video conferencing, the SDN controller can automatically allocate more bandwidth or choose a lower-latency path for that traffic, improving the user experience. On the other hand, low-priority traffic, such as email or background updates, can be routed through less congested or lower-priority paths. The continuous feedback loop of real-time traffic classification by ML models, combined with dynamic reconfiguration by SDN, enables more intelligent and efficient traffic management in complex networks. This is especially beneficial in environments with diverse and high-volume traffic, such as IoT ecosystems [65].

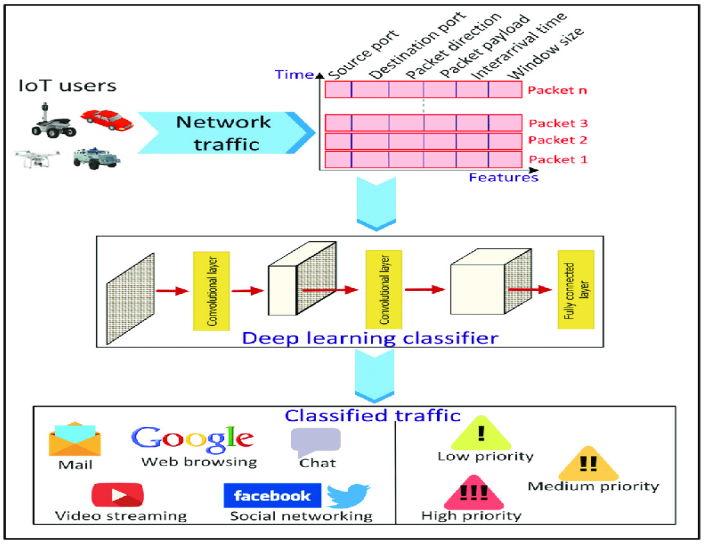

Figure 11: Deep Learning (DL)-based network traffic classification for internet of things (IoT) applications [65]

Conversely, classification models' accuracy depends on the labeled training data quality and capability to generalize to new attacks and traffic types. There is also a false negative and false positive risk, which can have substantial consequences in the context of network security.

3.2.2 Anomaly detection

Clustering groups similar network patterns or entities, which helps in user segmentation and anomaly detection [68]. However, its effectiveness relies on the choice of distance metrics and the algorithm used, and variations in initial parameter settings can lead to different outcomes. Anomaly detection methods identify unusual patterns in network traffic, which could indicate security breaches, system failures, or performance issues [69]. This is crucial for network security, as it can uncover threats that don't match known attack signatures. Defining what constitutes an "anomaly" can be

difficult, leading to missed detections or false alarms [70]. Anomaly detection models also need to adapt to evolving network conditions, which can complicate their deployment and ongoing maintenance. Additionally, the noise and high dimensionality of network data can make it challenging to differentiate between genuine inconsistencies and acceptable variations. Michau and Fink's study [71] supported the concept of domain adaptation for anomaly detection. They collected Condition Monitoring (CM) data from various units in a fleet. Because each unit and its environmental conditions had unique characteristics, the data exhibited different distributions, a phenomenon referred to as "Domain Shift." An analysis was performed using their proposed ADAU approach, which relied solely on noise-free CM data (represented by blue triangles and green stars), as shown in Figure 12 [71].

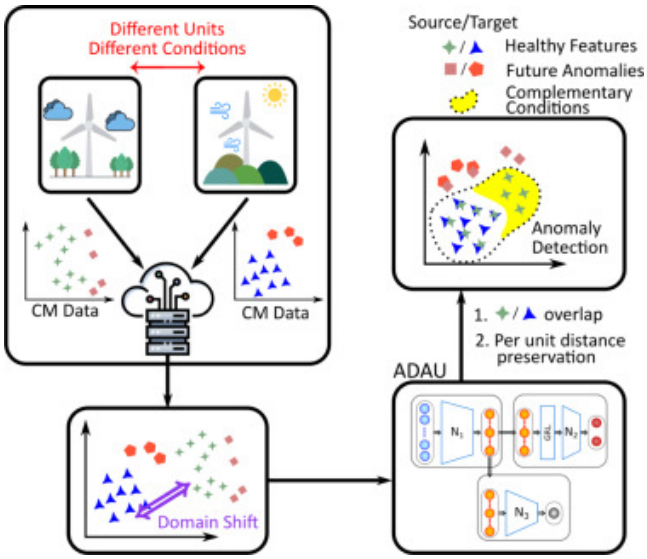

Figure 12: Anomaly detection [71]

3.2.3 Predictive analytics

Predictive analytics uses historical data to forecast future network conditions, including resource needs, potential failures, and traffic surges [72]. This allows for proactive network management, enabling administrators to allocate resources or take preventative actions before issues arise. As networks continue to grow in complexity and scale, relying solely on human-driven, centralized control will no longer be sufficient to handle the increasing complexity and unpredictable crises that may occur [73]. By integrating AI technologies and leveraging the vast data collected by the SDN controller for deep learning, AI-powered systems could address up to 90% of network security attacks or issues, offering valuable solutions for reference [74]. In terms of network planning and path optimization, future networks must meet requirements for low latency and high throughput [75]. Traditional path planning algorithms struggle to provide optimal traffic management solutions in real-time, especially as network traffic fluctuates. With AI, large-scale traffic data can be used to predict and efficiently plan traffic flows across network interfaces [76]. Artificial intelligence can play a crucial role in various aspects of network management, path planning, security, anomaly detection, and more. Figure 13 illustrates intelligent network traffic optimization and management [77]. It shows a layered approach to network management that combines SDN and AI. The intelligent control layer is made up of three main components: the AI analysis module, the SDN controller, and the network status collection module. The AI system processes real-time data to assess network conditions, optimize traffic flow, and predict potential problems, while the SDN controller uses these insights to adjust network configurations [77]. The network forwarding layer represents the physical infrastructure, including the Metropolitan Area Network and Backbone Network, along with distributed data centers (Edge, Regional, and Core) that work together to manage data traffic [77]. The data from this layer is continually sent to the AI system and SDN controller, enabling adaptive decision-making that improves efficiency, reduces latency, and enhances fault management [77]. This integrated AI and SDN architecture provides scalable, dynamic control, making it well-suited for complex environments like 5G networks and large-scale data infrastructures.

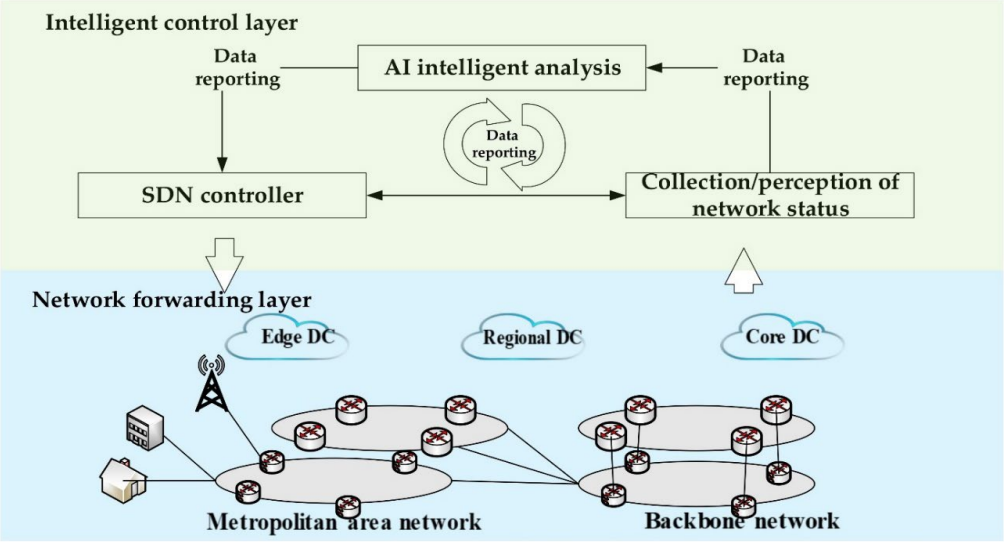

Figure 13: The architecture of intelligent network control [77]

However, the accuracy of prediction models relies heavily on the quality and availability of historical data, as well as the models' ability to adapt to unexpected changes or events in network behavior. In dynamic environments, predictive models can quickly become outdated, requiring regular updates and retraining to stay effective.

4. APPLICATIONS OF ML FOR TRAFFIC ENGINEERING IN SDN

4.1 Traffic prediction and forecasting

Predicting traffic patterns is crucial for effective network management as it allows for improved user experience, optimized routing, and proactive congestion control [78]. Traditional methods relied on static models and historical data, which often struggled in dynamic environments [79]. Machine learning (ML) has transformed traffic prediction by enabling models to learn from vast amounts of data and adapt to changing conditions. Google Maps, for instance, uses ML and AI to enhance traffic forecasting, leveraging models that analyze both historical and real-time traffic data [80]. The key advantage of ML in this context is its ability to process large datasets and provide accurate, real-time predictions that improve route planning and reduce travel times for users. Google continuously refines its traffic predictions by using various ML models, including time-series forecasting and regression, which take into account factors like traffic conditions, road closures, and weather, adjusting predictions in real-time. Google Maps' ML-powered traffic forecasts have become an essential tool for millions of users, helping to optimize routes and reduce congestion on a global scale [80]. While ML significantly improves traffic forecasting, concerns remain about its dependence on data quality and the challenges of processing data in real-time in highly variable environments. Additionally, privacy and security issues related to the vast amounts of user data collected require ongoing attention.

4.2 Dynamic resource allocation

In networking, dynamic resource allocation is essential for maximizing the use of limited resources like computing power and bandwidth. Machine learning (ML) adds flexibility by enabling real-time adjustments based on current and predicted network demands, improving efficiency [81]. For example, Netflix uses ML to measure performance [82]. The value of ML in this context lies in its ability to accurately predict content demand and allocate resources accordingly, ensuring smooth and high-quality streaming experiences. Similarly, companies like Amazon, YouTube, and Netflix leverage ML to explore ways to better serve their customers and generate personalized recommendations for movies and TV shows [82]. By using ML for dynamic resource allocation, Netflix can consistently provide a high-quality experience to its global user base while reducing infrastructure costs.

While ML offers substantial benefits in resource allocation, implementing these systems and the need for ongoing model retraining can be challenging. Additionally, the heavy reliance on accurate predictions highlights the importance of having robust data handling and processing systems in place.

4.3 Anomaly detection and fault management

In network management, quickly identifying and responding to anomalies is crucial for maintaining security and performance. Traditional methods often relied on static thresholds and manual analysis, which were prone to errors and slow responses. Machine learning (ML) enhances anomaly detection by identifying deviations and patterns that signal potential issues [83]. This approach has been widely adopted by organizations, leading to improvements in network reliability, reduced downtime, and faster responses to emerging problems [84]. While ML-driven anomaly detection is highly effective, its success depends on the quality and variety of the available data. False positives remain a challenge, as they can result in unnecessary caution and resource allocation. Additionally, the complexity of ML models requires advanced infrastructure and skilled personnel for implementation and ongoing maintenance.

4.4 Load balancing and path optimization

Effective path optimization and load balancing are crucial for maintaining network performance, especially in large, distributed systems. Machine learning (ML) provides a solution by dynamically optimizing path selection and load distribution based on current and predicted traffic conditions. For example, Facebook uses ML to optimize load balancing across its global data centers, ensuring efficient traffic delivery and minimizing latency [85]. ML enables continuous learning from network data, allowing the system to adapt to changes and improve load-balancing strategies over time. Facebook's infrastructure leverages predictive analytics and reinforcement learning within SDNs to efficiently distribute traffic across its data centers [85]. The ML models take into account factors like network latency, server load, and historical traffic patterns to make real-time routing decisions.

This approach has allowed Facebook to maintain high performance and reliability on its platform during peak traffic periods, enhancing the user experience [86]. While ML-based load balancing offers significant advantages, it also introduces complexity and requires ongoing monitoring and adjustment. The accuracy of ML models depends heavily on the quality of input data, and any errors can lead to suboptimal routing decisions [87]. Additionally, the carbon footprint and energy consumption of large-scale data centers are critical factors that must be balanced with performance goals. ML is proving to be a transformative force in traffic engineering, offering advanced solutions that traditional methods may not be able to achieve. Examples from leading companies like Google, Netflix, IBM, and Facebook highlight ML's practical benefits, from improving traffic forecasting to optimizing resource allocation and enhancing network reliability. However, implementing ML comes with challenges, including data reliance, complexity, and the need for continuous monitoring and adjustment. Overcoming these challenges is essential for fully realizing the potential of ML in traffic engineering.

5. CHALLENGES AND CONSIDERATIONS

As machine learning (ML) becomes increasingly integrated into SDN and other network management systems, there are several challenges and considerations that need to be addressed for successful implementation and operation. This section covers key issues, including the need for high-quality data, ensuring model accuracy and performance, scalability and real-time processing, as well as security and privacy concerns.

5.1 Data requirements and quality

Data quality is crucial for the success of any ML application, especially in networking and SDN. High-quality data ensures that ML models can learn effectively and make accurate predictions or decisions [88]. Poor data quality, such as missing values, noise, or inconsistencies, can lead to flawed models, resulting in degraded network performance or system failures. The importance of data quality cannot be overstated—incorrect or biased data can cause ML models to learn inaccurate patterns, leading to faulty predictions and decisions [89]. For example, in a network environment, if an ML model is trained on incomplete traffic data, it may fail to predict congestion

accurately, leading to poor load balancing or inefficient resource allocation [90]. Ensuring data quality requires rigorous data collection, preprocessing, and ongoing validation processes. However, achieving high data quality is challenging, particularly in large, dynamic systems where data is continuously generated from various sources.

Data collection and preprocessing are essential steps in preparing data for ML models, especially in SDNs where data comes from various sources, including network traffic, logs, and sensors [91]. Preprocessing involves cleaning, normalizing, and transforming this data into a format suitable for ML models. Challenges include handling large volumes of data, dealing with missing or incomplete information, and ensuring data consistency across different sources [92]. The process of data collection and preprocessing is often time-consuming and resource intensive. In SDNs, where real-time data processing is crucial, delays in preprocessing can hinder the timely application of ML models. Additionally, the diversity of data sources and formats creates significant challenges in ensuring the data is processed accurately and consistently [93]. Overcoming these challenges requires continuous data engineering practices, which can be complex and costly for the overall system.

5.2 Model accuracy and performance

There is often a trade-off between the accuracy of an ML model and its computational efficiency [94]. Highly accurate models tend to be more complex, requiring greater computational resources, which results in slower training times. On the other hand, simpler models may be faster but less accurate, making them better suited for real-time applications. In SDNs, where real-time decision-making is crucial, finding the right balance between model accuracy and computational efficiency is a significant challenge [95]. While a highly accurate model may provide better network performance predictions, it could introduce latency due to its heavy computational demands [96]. This trade-off needs to be carefully managed to ensure the model can deliver timely and reliable results without overburdening the network's computational resources. Techniques like model pruning, feature selection, and using approximate algorithms can help, but they must be applied carefully to avoid significant accuracy loss [36].

Additionally, evaluating the performance of ML models is crucial to ensure they meet the required standards for accuracy, precision, recall, and other relevant metrics. Common evaluation methods include feature selection, cross-validation, confusion matrices, appropriate algorithm selection, and ROC-AUC plots [61]. In the context of SDNs, it's also important to consider metrics related to overall network performance, such as latency and throughput. Choosing the right evaluation metrics is essential for assessing the effectiveness of an ML model. In networking applications, traditional metrics like accuracy may not fully reflect the model's impact on network operations [97]. For example, a model with high accuracy might still cause unacceptable delays if it is slow in processing data. Therefore, it's critical to combine standard ML metrics with network-specific ones to gain a more complete understanding of the model's performance. Additionally, models should be continually updated as network conditions and data evolve, since changes over time can affect their effectiveness.

5.3 Scalability and real-time processing

Scaling machine learning models to handle large networks presents a major challenge due to the rapid growth in data volume and computational demands [98]. This can create bottlenecks in decision-making processes. In SDNs, models that work well in smaller environments may struggle to perform effectively at scale, leading to wasted resources and potential system failures. As communication networks become more complex and dynamic, developing efficient traffic engineering (TE) policies become even more challenging, especially when optimizing traffic scheduling. Traditional methods often rely on fixed traffic models and pre-defined objectives, which might not always provide efficient solutions. However, combining Deep Reinforcement Learning (DRL) with SDN presents a promising way to create a model-free TE strategy using machine learning. Many existing DRL-based TE solutions face scalability issues, limiting their usefulness in larger networks. To overcome this challenge, ScaleDRL, a new network control framework, was introduced to integrate control theory with DRL techniques [92].This approach applies principles from pinning control theory to identify and assign critical links within the network. By collecting traffic distribution data from the SDN controller, ScaleDRL dynamically adjusts the weights of these

key links using a DRL algorithm. This allows for real-time changes to flow paths through a weighted shortest path algorithm, which is updated based on the adjusted link weights. Simulation results showed that ScaleDRL could significantly reduce the average end-to-end transmission delay by up to 39%, outperforming other leading DRL-based TE methods across various network topologies [92]. While real-time processing is vital for optimal network performance, it introduces challenges like computational costs, delays, and issues with data consistency and synchronization. Effective planning and resource allocation are crucial to achieving this goal.

5.4 Security and privacy concerns

Integrating machine learning (ML) into SDNs introduces privacy risks and new security challenges. For example, ML models can be vulnerable to adversarial attacks, where an attacker manipulates input data to make the model produce incorrect decisions [96]. Additionally, the collection and processing of large amounts of data for ML purposes can raise privacy concerns, especially if sensitive or identifiable information is involved. Privacy and security are critical considerations when implementing ML in SDNs. Adversarial attacks can have serious consequences, such as network disruptions or data breaches, if the ML model is tricked into making harmful decisions [97]. Protecting against these attacks requires robust security measures, including the use of secure ML techniques, model validation, and monitoring for unusual patterns that could indicate an attack. Privacy concerns must also be addressed, particularly when handling sensitive data. Techniques such as data anonymization, secure multi-party computation, and differential privacy can help mitigate these risks, but they may also add complexity and reduce the efficiency of the ML models.

6. RECENT ADVANCES AND TRENDS

The intersection of machine learning (ML) and Software-Defined Networking (SDN) is a rapidly evolving field, with continuous advancements and emerging trends shaping the future of network management. This section explores recent innovations in ML techniques for SDN, the integration of ML with other technologies, and future research directions that could address current gaps and lead to further breakthroughs.

6.1 Emerging ML techniques

In recent years, there has been a surge of innovative ML techniques applied to SDN, driven by the demand for more flexible and efficient network management. Notable advancements include Deep Reinforcement Learning (DRL), which combines deep learning with reinforcement learning to create models that can learn through trial and error in their environment [99]. In SDNs, DRL has been applied to dynamic resource allocation, traffic management, and security enforcement, offering more adaptable and robust solutions compared to traditional methods [100]. Additionally, the Unified Learning approach allows ML models to be trained across decentralized data sources, preserving data privacy while benefiting from large-scale datasets. In SDNs, unified learning can optimize network performance across different areas without centralizing sensitive information. On the other hand, as ML models become increasingly complex, there is a growing need for transparency in how these models make decisions. Explainable AI (XAI) methods are being developed to provide insights into the decision-making process of ML models in SDNs, which is essential for troubleshooting, compliance, and building trust [101].

While these emerging methods hold great potential, they also present several challenges. For example, DRL demands substantial computational resources and time for training, which can be a significant obstacle in real-time SDN environments [11, 102]. Federated learning helps address privacy concerns but introduces new difficulties in ensuring model consistency and managing diverse data types. Additionally, while XAI improves transparency, it could also make models vulnerable to incompatible attacks if not carefully designed. These advancements represent a significant step forward, but their practical implementation requires careful consideration of the trade-offs and potential risks.

6.2 Integration with other technologies

The integration of machine learning with other network management technologies is driving significant improvements in the efficiency and capabilities of SDNs. For example, in Edge Computing, ML models are increasingly being deployed at the network edge, where data is

generated. This integration enables real-time analytics and decision-making closer to the data source, reducing latency and enhancing responsiveness in SDN environments.

By integrating machine learning with Network Function Virtualization (NFV), network functions become more flexible and efficient [103]. For example, ML can predict demand and optimize the allocation of virtual resources in real-time, leading to better resource utilization and reduced operational costs. The growth of Internet of Things (IoT) devices has added complexity to network management [61]. ML techniques are incorporated with SDN to handle the massive amounts of data generated by IoT devices, ensuring efficient routing, load balancing, and security [96].

Integrating machine learning with these advancements enhances the capabilities of SDNs, but it also introduces complexity. As edge computing reduces latency, it requires a distributed system that can be difficult to manage and secure. The combination of ML with NFV and IoT adds new dimensions of scalability and interoperability, which must be carefully managed to prevent bottlenecks and security risks [97]. Additionally, these integrations demand significant investments in infrastructure and expertise, which could be a challenge for smaller organizations.

7. FUTURE RESEARCH DIRECTIONS

Despite significant progress in ML for SDN, there are still research gaps to be addressed. Security vulnerabilities, scalability issues, and real-time decision-making remain challenges that need attention. Future research should focus on ML-driven approaches, energy-efficient ML, and autonomous networking, with the integration of ML and AI technologies offering more flexible solutions for SDN. Energy-efficient ML models, which balance performance with energy savings, are also crucial. While autonomous networking holds promise, it raises ethical and security concerns. Addressing these challenges is essential for the continued advancement of ML in SDN, as it could unlock new levels of performance and security. Further research in these areas could lead to important breakthroughs in the field.

REFERENCES

Z. Liu, Y. Wang, F. Feng, Y. Liu, Z. Li, and Y. Shan, "A DDoS detection method based on feature engineering and machine learning in software-defined networks," *Sensors,* vol. 23, no. 13, p. 6176, 2023.

Y. Wang, X. Wang, M. M. Ariffin, M. Abolfathi, A. Alqhatani, and L. Almutairi, "Attack detection analysis in software-defined networks using various machine learning method," *Computers and Electrical Engineering,* vol. 108, p. 108655, 2023.

M. R. Ahmed, S. Shatabda, A. M. Islam, and M. T. I. Robin, "Intrusion Detection System in Software-Defined Networks Using Machine Learning and Deep Learning Techniques--A Comprehensive Survey," *Authorea Preprints,* 2023.

A. A. Bahashwan, M. Anbar, S. Manickam, T. A. Al-Amiedy, M. A. Aladaileh, and I. H. Hasbullah, "A systematic literature review on machine learning and deep learning approaches for detecting DDoS attacks in software-defined networking," *Sensors,* vol. 23, no. 9, p. 4441, 2023.

J. Cunha *et al.*, "Enhancing Network Slicing Security: Machine Learning, Software-Defined Networking, and Network Functions Virtualization-Driven Strategies," *Future Internet,* vol. 16, no. 7, p. 226, 2024.

A. Shirmarz and A. Ghaffari, "Network traffic discrimination improvement in software defined network (SDN) with deep autoencoder and ensemble method," *Journal of Ambient Intelligence and Humanized Computing,* vol. 14, no. 5, pp. 6321-6337, 2023.

M. Sreelekha and Midhunchakkaravarthy, "Intelligent Transportation System for Sustainable and Efficient Urban Mobility: Machine Learning Approach for Traffic Flow Prediction," in *International Conference on Multi-Strategy Learning Environment*, 2024: Springer, pp. 399-412.

H. Elubeyd and D. Yiltas-Kaplan, "Hybrid deep learning approach for automatic DoS/DDoS attacks detection in software-defined networks," *Applied Sciences,* vol. 13, no. 6, p. 3828, 2023.

P. A. D. S. N. Wijesekara and S. Gunawardena, "A Machine Learning-Aided Network Contention-Aware Link Lifetime-and Delay-Based Hybrid Routing Framework for Software-Defined Vehicular Networks," in *Telecom*, 2023, vol. 4, no. 3: MDPI, pp. 393-458.

M. Elnawawy, A. Sagahyroon, and T. Shanableh, "FPGA-based network traffic classification using machine learning," *IEEE Access,* vol. 8, pp. 175637-175650, 2020.

W. C. Chanhemo, M. H. Mohsini, M. M. Mjahidi, and F. U. Rashidi, "Deep learning for SDN-enabled campus networks: proposed solutions, challenges and future directions," *International Journal of Intelligent Computing and Cybernetics,* vol. 16, no. 4, pp. 697-726, 2023.

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

A. A. Alashhab *et al.*, "Enhancing DDoS attack detection and mitigation in SDN using an ensemble online machine learning model," *IEEE Access,* 2024.

N. Kumari and K. Kathuria, "Overview of SDN Building Foundations and Applications," *Journal of Research in Science and Engineering,* vol. 6, no. 7, pp. 43-53, 2024.

R. Network and P. M. Gupta, "6 Software-Defined," *Software-Defined Network Frameworks: Security Issues and Use Cases,* p. 89, 2024.

M. K. Awad, M. El-Shafei, T. Dimitriou, Y. Rafique, M. Baidas, and A. Alhusaini, "Power-efficient routing for SDN with discrete link rates and size-limited flow tables: A tree-based particle swarm optimization approach," *International Journal of Network Management,* vol. 27, no. 5, p. e1972, 2017.

D. Perepelkin, M. Ivanchikova, and T. Nguyen, "Research of Multipath Routing and Load Balancing Processes in Software Defined Networks Based on Bird Migration Algorithm," in *2023 International Russian Smart Industry Conference (SmartIndustryCon)*, 2023: IEEE, pp. 247-252.

S. Keerthiga and R. Murugeswari, "Survey on software defined networking in IoT," in *AIP Conference Proceedings*, 2023, vol. 2548, no. 1: AIP Publishing.

C. N. Tadros, B. Mokhtar, and M. R. Rizk, "Software defined network-based management architecture for 5g network," in *Paradigms of Smart and Intelligent Communication, 5G and Beyond*: Springer, 2023, pp. 171-195.

M. K. Vadlamudi and K. Rayudu, "A Review on Traffic Engineering Systems in Software-defined Networks using Routing Mechanisms," in *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, 2024: IEEE, pp. 1549-1554.

B. J. Ospina Cifuentes, Á. Suárez, V. García Pineda, R. Alvarado Jaimes, A. O. Montoya Benitez, and J. D. Grajales Bustamante, "Analysis of the Use of Artificial Intelligence in Software-Defined Intelligent Networks: A Survey," *Technologies,* vol. 12, no. 7, p. 99, 2024.

X. Pei, P. Sun, Y. Hu, D. Li, B. Chen, and L. Tian, "Enabling efficient routing for traffic engineering in SDN with Deep Reinforcement Learning," *Computer Networks,* vol. 241, p. 110220, 2024.

B. Lin, Y. Guo, H. Luo, and M. Ding, "TITE: A transformer-based deep reinforcement learning approach for traffic engineering in hybrid SDN with dynamic traffic," *Future Generation Computer Systems,* vol. 161, pp. 95-105, 2024.

S. Mehraban and R. K. Yadav, "Traffic engineering and quality of service in hybrid software defined networks," *China Communications,* vol. 21, no. 2, pp. 96-121, 2024.

U. Prabu and V. Geetha, "Towards the implementation of traffic engineering in SDN: a practical approach," in *Inventive Systems and Control: Proceedings of ICISC 2023*: Springer, 2023, pp. 155-161.

S. Dou, L. Qi, J. Wang, and Z. Guo, "EPIC: Traffic Engineering-Centric Path Programmability Recovery Under Controller Failures in SD-WANs," *IEEE/ACM Transactions on Networking,* no. 01, pp. 1-14, 2024.

Z. Guo, C. Li, Y. Li, S. Dou, B. Zhang, and W. Wu, "Maintaining the Network Performance of Software-Defined WANs With Efficient Critical Routing," *IEEE Transactions on Network and Service Management,* 2023.

F. Hao, S. Jing, and C. Zhao, "Link load balancing scheme for elephant flow in SDN data center," in *2023 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, 2023: IEEE, pp. 1026-1033.

[29] M. Hamdan *et al.*, "Elephant flow detection intelligence for software-defined networks: a survey on current techniques and future direction," *Evolutionary Intelligence,* pp. 1-19, 2024.

M. Hassan, M. A. Gregory, and S. Li, "Multi-Domain Federation Utilizing Software Defined Networking—A Review," *IEEE Access,* vol. 11, pp. 19202-19227, 2023.

M. AbdulRaheem *et al.*, "Machine learning assisted snort and zeek in detecting DDoS attacks in software-defined networking," *International Journal of Information Technology,* vol. 16, no. 3, pp. 1627-1643, 2024.

A. M. Eldhai *et al.*, "Improved Feature Selection and Stream Traffic Classification Based on Machine Learning in Software-Defined Networks," *IEEE Access,* 2024.

K. T. Dinh, S. Kukliński, W. Kujawa, and M. Ulaski, "MSDN-TE: Multipath based traffic engineering for SDN," in *Intelligent Information and Database Systems: 8th Asian Conference, ACIIDS 2016, Da Nang, Vietnam, March 14–16, 2016, Proceedings, Part II 8*, 2016: Springer, pp. 630-639.

K. T. Mehmood, S. Atiq, and M. M. Hussain, "Enhancing QoS of Telecom Networks through Server Load Management in Software-Defined Networking (SDN)," *Sensors,* vol. 23, no. 23, p. 9324, 2023.

A. H. Alhilali and A. Montazerolghaem, "Artificial intelligence based load balancing in SDN: A comprehensive survey," *Internet of Things,* vol. 22, p. 100814, 2023.

D. Shah *et al.*, "FAST: AI-based Network Traffic Analysis and Load Balancing Framework Underlying SDN Clusters," in *2024 8th International Conference on Smart Cities, Internet of Things and Applications (SCIoT)*, 2024: IEEE, pp. 82-87.

K. Truong Dinh, S. Kukliński, T. Osiński, and J. Wytrębowicz, "Heuristic traffic engineering for SDN," *Journal of Information and Telecommunication,* vol. 4, no. 3, pp. 251-266, 2020.

S. Ahmad, F. Jamil, A. Ali, E. Khan, M. Ibrahim, and T. K. Whangbo, "Effectively Handling Network Congestion and Load Balancing in Software-Defined Networking," *Computers, Materials & Continua,* vol. 70, no. 1, 2022.

J. Gómez-delaHiz and J. Galán-Jiménez, "Improving the Traffic Engineering of SDN networks by using Local Multi-Agent Deep Reinforcement Learning," in *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, 2024: IEEE, pp. 1-5.

J. Xiao, X. Pan, J. Liu, J. Wang, P. Zhang, and L. Abualigah, "Load balancing strategy for SDN multi-controller clusters based on load prediction," *The Journal of Supercomputing,* vol. 80, no. 4, pp. 5136-5162, 2024.

Z. Ye, G. Sun, and M. Guizani, "ILBPS: An Integrated Optimization Approach Based on Adaptive Load-Balancing and Heuristic Path Selection in SDN," *IEEE Internet of Things Journal,* 2023.

H. Iesar *et al.*, "Revolutionizing Data Center Networks: Dynamic Load Balancing via Floodlight in SDN Environment," in *2024 5th International Conference on Advancements in Computational Sciences (ICACS)*, 2024: IEEE, pp. 1-8.

K. Vani and K. RamaMohanBabu, "An Intelligent Server load balancing based on Multi-criteria decision-making in SDN," *International journal of electrical and computer engineering systems,* vol. 14, no. 4, pp. 433-442, 2023.

M. Karakus, "GATE-BC: Genetic Algorithm-Powered QoS-Aware Cross-Network Traffic Engineering in Blockchain-Enabled SDN," *IEEE Access,* 2024.

L. Davoli, L. Veltri, P. L. Ventre, G. Siracusano, and S. Salsano, "Traffic engineering with segment routing: SDN-based architectural design and open source implementation," in *2015 Fourth European Workshop on Software Defined Networks*, 2015: IEEE, pp. 111-112.

S. Salsano *et al.*, "Hybrid IP/SDN networking: open implementation and experiment management tools," *IEEE Transactions on Network and Service Management,* vol. 13, no. 1, pp. 138-153, 2015.

Y. Feng, "Application of hybrid genetic algorithm in large traffic scheduling in SDN architecture," *International Journal of Wireless and Mobile Computing,* vol. 24, no. 3-4, pp. 341-351, 2023.

P. Boryło *et al.*, "SDNRoute: Proactive routing optimization in Software Defined Networks," *Computer Communications,* 2024.

M. Beshley, N. Kryvinska, H. Beshley, O. Panchenko, and M. Medvetskyi, "Traffic engineering and QoS/QoE supporting techniques for emerging service-oriented software-defined network," *Journal of Communications and Networks,* vol. 26, no. 1, pp. 99-114, 2024.

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

D. Nuñez-Agurto, W. Fuertes, L. Marrone, E. Benavides-Astudillo, and M. Vásquez-Bermúdez, "Traffic classification in software-defined networking by employing deep learning techniques: a systematic literature review," in *International Conference on Technologies and Innovation*, 2023: Springer, pp. 67-80.

V. Tong, S. Souihi, H. A. Tran, and A. Mellouk, "Machine learning based root cause analysis for SDN network," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021: IEEE, pp. 1-6.

R. B. Shohani and S. A. Mostafavi, "Introducing a new linear regression based method for early DDoS attack detection in SDN," in *2020 6th International Conference on Web Research (ICWR)*, 2020: IEEE, pp. 126-132.

L. Zeng, "[Retracted] Analysis of the Stage Performance Effect of Environmental Protection Music and Dance Drama Based on Artificial Intelligence Technology," *Journal of Environmental and Public Health,* vol. 2022, no. 1, p. 2891993, 2022.

A. I. Owusu and A. Nayak, "An intelligent traffic classification in sdn-iot: A machine learning approach," in *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2020: IEEE, pp. 1-6.

M. M. Raikar, S. Meena, M. M. Mulla, N. S. Shetti, and M. Karanandi, "Data traffic classification in software defined networks (SDN) using supervised-learning," *Procedia Computer Science,* vol. 171, pp. 2750-2759, 2020.

R. H. Serag, M. S. Abdalzaher, H. A. E. A. Elsayed, M. Sobh, M. Krichen, and M. M. Salim, "Machine-Learning-Based Traffic Classification in Software-Defined Networks," *Electronics,* vol. 13, no. 6, p. 1108, 2024.

M. Kuzlu, C. Fair, and O. Guler, "Role of artificial intelligence in the Internet of Things (IoT) cybersecurity," *Discover Internet of things,* vol. 1, no. 1, p. 7, 2021.

T. Jafarian, M. Masdari, A. Ghaffari, and K. Majidzadeh, "SADM-SDNC: security anomaly detection and mitigation in software-defined networking using C-support vector classification," *Computing,* vol. 103, no. 4, pp. 641-673, 2021.

M. Kim and J. Kim, "Extending the coverage area of regional ionosphere maps using a support vector machine algorithm," in *Annales Geophysicae*, 2019, vol. 37, no. 1: Copernicus Publications Göttingen, Germany, pp. 77-87.

O. Belkadi, A. Vulpe, Y. Laaziz, and S. Halunga, "ML-Based Traffic Classification in an SDN-Enabled Cloud Environment," *Electronics,* vol. 12, no. 2, p. 269, 2023.

T. Yang, S. Vural, P. Qian, Y. Rahulan, N. Wang, and R. Tafazolli, "Achieving robust performance for traffic classification using ensemble learning in SDN networks," in *ICC 2021-IEEE International Conference on Communications*, 2021: IEEE, pp. 1-6.

T. V. Phan, T. G. Nguyen, N.-N. Dao, T. T. Huong, N. H. Thanh, and T. Bauschert, "DeepGuard: Efficient anomaly detection in SDN with fine-grained traffic flow monitoring," *IEEE Transactions on Network and Service Management,* vol. 17, no. 3, pp. 1349-1362, 2020.

A. Malik, R. de Fréin, M. Al-Zeyadi, and J. Andreu-Perez, "Intelligent SDN traffic classification using deep learning: Deep-SDN," in *2020 2nd International Conference on Computer Communication and the Internet (ICCCI)*, 2020: IEEE, pp. 184-189.

L. Ismail and R. Buyya, "Artificial intelligence applications and self-learning 6G networks for smart cities digital ecosystems: Taxonomy, challenges, and future directions," *Sensors,* vol. 22, no. 15, p. 5750, 2022.

X. Ren, H. Gu, and W. Wei, "Tree-RNN: Tree structural recurrent neural network for network traffic classification," *Expert Systems with Applications,* vol. 167, p. 114363, 2021.

M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for Internet of Things," *IEEE access,* vol. 5, pp. 18042-18050, 2017.

M. J. Nazar, A. Alhudhaif, K. N. Qureshi, S. Iqbal, and G. Jeon, "Signature and flow statistics based anomaly detection system in software-defined networking for 6G internet of things network," *International Journal of System Assurance Engineering and Management,* pp. 1-11, 2023.

M. F. Akbaş, C. Güngör, and E. Karaarslan, "Usage of machine learning algorithms for flow based anomaly detection system in software defined networks," in *Intelligent and Fuzzy Techniques: Smart and Innovative Solutions: Proceedings of the INFUS 2020 Conference, Istanbul, Turkey, July 21-23, 2020*, 2021: Springer, pp. 1156-1163.

G. Khekare *et al.*, "Optimizing Network Security and Performance Through the Integration of Hybrid GAN-RNN Models in SDN-based Access Control and Traffic Engineering," *International Journal of Advanced Computer Science & Applications,* vol. 14, no. 12, 2023.

G. Michau and O. Fink, "Unsupervised transfer learning for anomaly detection: Application to complementary operating condition transfer," *Knowledge-Based Systems,* vol. 216, p. 106816, 2021.

V. A. Reddy, K. Venkatesh, and L. Srinivas, "Software defined networking based delay sensitive traffic engineering of critical data in internet of things," *Journal of Computational and Theoretical Nanoscience,* vol. 17, no. 1, pp. 48-53, 2020.

M. Tian, C. Sun, and S. Wu, "An EMD and ARMA-based network traffic prediction approach in SDN-based internet of vehicles," *Wireless Networks,* pp. 1-13, 2021.

A. H. Abdi *et al.*, "Security Control and Data Planes of SDN: A Comprehensive Review of Traditional, AI and MTD Approaches to Security Solutions," *IEEE Access,* 2024.

T. E. Ali, A. H. Morad, and M. A. Abdala, "Traffic management inside software-defined data centre networking," *Bulletin of Electrical Engineering and Informatics,* vol. 9, no. 5, pp. 2045-2054, 2020.

R. Setiawan *et al.*, "Encrypted network traffic classification and resource allocation with deep learning in software defined network," *Wireless Personal Communications,* pp. 1-17, 2022.

A. Guo and C. Yuan, "Network intelligent control and traffic optimization based on SDN and artificial intelligence," *Electronics,* vol. 10, no. 6, p. 700, 2021.

K. T. Selvi and R. Thamilselvan, "An intelligent traffic prediction framework for 5G network using SDN and fusion learning," *Peer-to-Peer Networking and Applications,* vol. 15, no. 1, pp. 751-767, 2022.

D. Adanza *et al.*, "Enabling traffic forecasting with cloud-native SDN controller in transport networks," *Computer Networks,* vol. 250, p. 110565, 2024.

M. A. Pramanik, M. M. Rahman, A. I. Anam, A. A. Ali, M. A. Amin, and A. M. Rahman, "Modeling traffic congestion in developing countries using google maps data," in *Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 1*, 2021: Springer, pp. 513-531.

M. Anoushee, M. Fartash, and J. Akbari Torkestani, "An intelligent resource management method in SDN based fog computing using reinforcement learning," *Computing,* vol. 106, no. 4, pp. 1051-1080, 2024.

G. G. Hallur, A. Aslekar, and S. G. Prabhu, "Digital solution for entertainment: An overview of over the top (ott) and digital media," *Digital Entertainment as Next Evolution in Service Sector: Emerging Digital Solutions in Reshaping Different Industries,* pp. 35-53, 2023.

A. Hirsi, L. Audah, A. Salh, N. M. Sahar, S. Ahmed, and M. A. Alhartomi, "DDoS Anomaly Detection in Software-Defined Networks: An Evaluation of Machine Learning Techniques for Traffic Classification and Prediction," in *2024 International Conference on Future Technologies for Smart Society (ICFTSS)*, 2024: IEEE, pp. 100-105.

L. Desgeorges, J.-P. Georges, and T. Divoux, "Detection of anomalies of a non-deterministic software-defined networking control," *Computers & Security,* vol. 129, p. 103228, 2023.

M. D. Tache, O. Păscuțoiu, and E. Borcoci, "Optimization Algorithms in SDN: Routing, Load Balancing, and Delay Optimization," *Applied Sciences,* vol. 14, no. 14, p. 5967, 2024.

K. Hwang, *Cloud computing for machine learning and cognitive applications*. Mit Press, 2017.

C. Kumar, S. Marston, R. Sen, and A. Narisetty, "Greening the cloud: a load balancing mechanism to optimize cloud computing networks," *Journal of Management Information Systems,* vol. 39, no. 2, pp. 513-541, 2022.

M. F. Audah, T. S. Chin, Y. Zulfadzli, C. K. Lee, and K. Rizaluddin, "Towards efficient and scalable machine learning-based QoS traffic classification in software-defined network," in *Mobile Web and Intelligent Information Systems: 16th International Conference, MobiWIS 2019, Istanbul, Turkey, August 26–28, 2019, Proceedings 16*, 2019: Springer, pp. 217-229.

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

J. Kwon, D. Jung, and H. Park, "Traffic data classification using machine learning algorithms in SDN networks," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020: IEEE, pp. 1031-1033.

L. Yanjun, L. Xiaobo, and Y. Osamu, "Traffic engineering framework with machine learning based meta-layer in software-defined networks," in *2014 4th IEEE International Conference on Network Infrastructure and Digital Content*, 2014: IEEE, pp. 121-125.

P. Sun, Z. Guo, J. Lan, J. Li, Y. Hu, and T. Baker, "ScaleDRL: A scalable deep reinforcement learning approach for traffic engineering in SDN with pinning control," *Computer Networks,* vol. 190, p. 107891, 2021.

Z. Fan and R. Liu, "Investigation of machine learning based network traffic classification," in *2017 International Symposium on Wireless Communication Systems (ISWCS)*, 2017: IEEE, pp. 1-6.

R. Thupae, B. Isong, N. Gasela, and A. M. Abu-Mahfouz, "Machine learning techniques for traffic identification and classifiacation in SDWSN: A survey," in *IECON 2018-44th annual conference of the IEEE Industrial Electronics Society*, 2018: IEEE, pp. 4645-4650.

G. Wassie, J. Ding, and Y. Wondie, "Traffic prediction in SDN for explainable QoS using deep learning approach," *Scientific Reports,* vol. 13, no. 1, p. 20607, 2023.

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

Y. Yoo, G. Yang, C. Shin, J. Lee, and C. Yoo, "Machine Learning-Based Prediction Models for Control Traffic in SDN Systems," *IEEE Transactions on Services Computing,* 2023.

H. Padmanaban, "Machine Learning Algorithms Scaling on Large-Scale Data Infrastructure," *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023,* vol. 3, no. 1, pp. 171-196, 2024.

M. Ye, J. Zhang, Z. Guo, and H. J. Chao, "Date: Disturbance-aware traffic engineering with reinforcement learning in software-defined networks," in *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, 2021: IEEE, pp. 1-10.

S. Troia, F. Sapienza, L. Varé, and G. Maier, "On deep reinforcement learning for traffic engineering in SD-WAN," *IEEE Journal on Selected Areas in Communications,* vol. 39, no. 7, pp. 2198-2212, 2020.

D.-H. Le, H.-A. Tran, S. Souihi, and A. Mellouk, "An ai-based traffic matrix prediction solution for software-defined network," in *ICC 2021-IEEE International Conference on Communications*, 2021: IEEE, pp. 1-6.

G. Kim, Y. Kim, and H. Lim, "Deep reinforcement learning-based routing on software-defined networks," *IEEE Access,* vol. 10, pp. 18121-18133, 2022.

E. Vaezpour, "Deep learning-driven multi-objective dynamic switch migration in software defined networking (SDN)/network function virtualization (NFV)-based 5G networks," *Engineering Applications of Artificial Intelligence,* vol. 125, p. 106714, 2023.