Adaptive Traffic Segregation and Simulation in Software-Defined Healthcare Networks

Bhargavi Dalal[1,2*] and Pooja Sapra [2]

[1] Nutan Maharashtra Institute of Engineering and Technology, Talegaon, Pune, India
[2] Department of Information and Technology, PIET, Parul University, Vadodara, Gujarat, India
*bhargavi.b.dalal@gmail.com

Abstract. Increasing dependency and demand on digital healthcare systems, needs for secure, efficient, and intelligent management of patient data traffic have become critical. This research explores the study of patient related network traffic in a Software-Defined Networking (SDN) environment. Some promising paradigms of SDN encompass the ability to manage the network centrally with greater flexibility and decoupling of the control plane from the data plane accompanied by increased level of security. SDN's centralized architecture and programmability provide enhanced visibility and control over data flows. It makes highly suitable for handling sensitive healthcare data such as electronic health records (EHRs), real-time monitoring outputs, and medical imaging. The main objective of the research is to make sure that patient's confidential data is transmitted safely and reliably in the hospital, clinic, telemedicine platform, and health information system environment. The scope includes monitoring the flows of patient's data to get optimized network performance by ensuring regulatory compliance. This also includes to facilitates secure interoperability between healthcare systems along with enabled real-time clinical decision-making through traffic analytics. Key findings indicate that SDN-based traffic analysis significantly improves anomaly detection, reduces the risk of data breaches, and enhances the overall reliability, security, and efficiency of healthcare networks.

Keywords: Software Defined Network, SDN Network, Traffic Analysis.

1 Introduction

The healthcare networks deal with various kinds of data, including treatment follow- up notifications, medical images, and hospital records. Part of the information such as live heart rate display is incredibly essential and should be transmitted to doctors as fast as possible. Delay on other data, like administrative data, can be tolerated [1], [2]. When all the data are treated in a similar fashion, the network will be flooded and vital information will take more time. The adaptive traffic segregation may provide a solution to this issue as it separates valuable healthcare data and assigns it greater priority, thereby making it traverse the most effective and the shortest network paths [3].

The programmability of Software-Defined Networking (SDN) is especially useful in

J. Saxena et al. (eds.), *Proceedings of the International Conference on Sustainable Micro-Nano Materials & Innovative Technology (ICSUMMIT 2026)*, Atlantis Highlights in Engineering 44,
https://doi.org/10.2991/978-94-6239-727-9_9

healthcare, which provides the ability to have flexible and efficient control of the network. Hospitals and the clinics processes large amounts of patient data every day and SDN makes it possible to manage such information safely and in real-time [4]. Programmable networks can enable healthcare providers to dynamically assign resources, enhancing services, including telemedicine, electronic health records and wearable de- vice monitoring. Such functions minimize delay, enhance dependability, and promote even better patient care with minimal mistakes [5]

This paper dwells upon three main areas in healthcare networks. First, it provides a review of traffic separation methods meant to ensure the isolation of sensitive patient information and improved security. Second, it discusses simulation frameworks that can be applied in testing and deploying SDN-based healthcare systems. Lastly, it brings to the fore healthcare-related needs such as low latency of telemedicine and high protection of medical records, and SDN programmability is crucial to address these needs [6], [7].

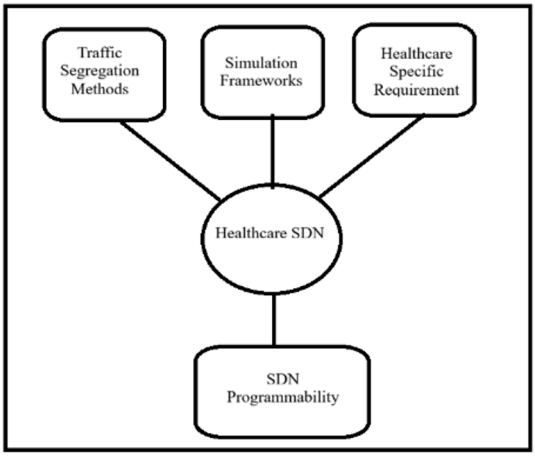

Fig. 1. Core Design Aspects of SDN in Healthcare Networks

Figure 1 represents the conceptual model of Healthcare Software-Defined Network- ing (SDN) and its ability to successfully use traffic segregation mechanisms, simulation models, healthcare-specific demands, and SDN programmability to provide efficient, safe, and adaptable management of healthcare networks. The main contributions of the paper are as follows: This paper presents the available literature on medical data priority in health networks, and how the issue of latency, reliability, and security impact the delivery of life-sensitive infor- mation. It highlights how secure and priority sensitive network architectures are crucial to ensure punctual and safe delivery of sensitive medical information.

The paper explores the old approach to traffic classification using packets in SDN and indicates the transition to flow-based and smart classification approaches of traffic. The application of SDN programmability and network slicing in facilitating the adapt- ability and efficiency of traffic management is also talked over.

The paper divides and discusses typical cyber-attacks in SDN settings such as TCP/UDP

flooding, ARP spoofing, DDoS, and brute force attacks. It analyzes the weaknesses of signature-based detection systems and highlights the benefits of machine learning-based systems to improve threat detection. This paper reviews on use of machine learning in SDN, which provides an overall view of current practices, trends, and research gaps to be filled, hence giving a clear picture of future trends in the area.

1.1 Paper Outline

The paper is structured in the following way. In section 2, the background on the centralized SDN architecture is provided, including the application and control planes, the important research issues, the discussion on centralized SDN controllers, and a literature review. Section 3 specifies the prerequisites of the secure healthcare networks, which include priorities of the essential data and the reliable functioning. Section 4 is devoted to the process of traffic classification and management in SDN and compares traditional and modern methods and approaches, comparing flow-based and packet-based methods and outlines the functions of traffic separation, priority, network slice and dynamical traffic control. Section 5 discusses detection of cyberattack in SDN with an overview of most common types of attacks and a comparison of signature-based detection and machine learning-based detection. Section 6 considers simulation tools of experimental SDN environments, and Section 7 contains the discussion of the application of machine learning techniques in SDN. Section 8 offers a comparative analysis of the surveys that have been carried out on the field. Section 9 discusses open challenges and research gaps. Section 10 presents future research directions for traffic segregation and intelligent network traffic detection in Enhanced-SDN environments. And lastly paper gets concluded with summary of study in terms of segregation of network traffic.

2 Background and Preliminaries

A traditional SDN (Software-Defined Networking) network is a modern approach to designing and managing computer networks. Unlike conventional networks, traditionally in which the control and data planes are tightly coupled within each networking device (such as routers and switches), SDN separates these planes, as shown in Figure 2. Although SDN is a modern networking paradigm, one of its key weaknesses is also its centralized control architecture, which makes it more vulnerable to Distributed Denial of Service (DDoS) attacks than conventional networks [8]. Figure 2 illustrates the fundamental architectural differences between traditional networking and Software-Defined Networking (SDN), highlighting the separation of control and data-plane functionality.

In conventional networks, the application plane, control plane, and data plane are closely related to each other in single network devices, making it tightly coupled and vendor-specific hardware, with each network device making its own decision-making (control plane) and packet forwarding (data plane) decisions. The plane communicates desired network behaviors to the controller through northbound APIs, enabling centralized, policy-driven network management and greater adaptability. This isolation increases

international observability, eases the structure, and facilitates quick adjustment to evolving traffic requirements, which makes SDN especially appropriate in environments that demand a high level of dependability, fine-grained Quality of Service (quality of service), and centralized control.

Software-Defined Networking (SDN) isolates the network into two components: the control plane and the data plane. The control plane determines how data is to travel, whilst the data plane transmits the actual packets. SDN uses the OpenFlow protocol to transmit instructions from the controller to the switches. The best-known protocol for implementing SDN is OpenFlow. It is a standard interface which enables the controller to interact with network switches. It operates on the basis of flow tables that are stored in the switches and are similar to a kind of a set of rules that the switch operates in order to process network traffic. The computer brain in the network is known as the controller which makes decisions and adjusts them real-time. Prediction and credible classification will allow the SDN controller to make the best routing decisions as they vary in time [9]. This type of arrangement simplifies the management of the network and gives a possibility to use it in other applications, such as health.

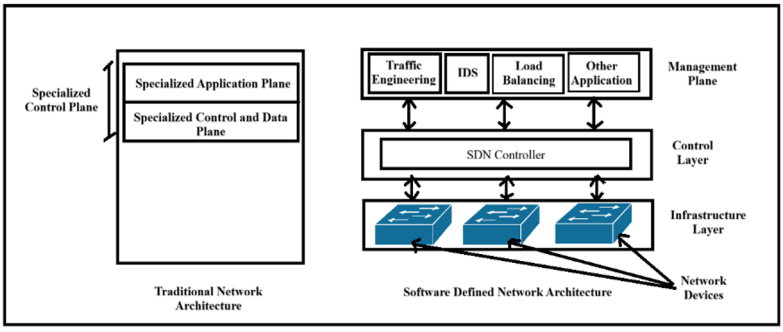

Fig. 2. Traditional Network vs. Software-Defined Network Architecture

2.1 Literature Survey

Table 1: Structured Literature Review

Category	Authors (Citation No.)	Key Idea / Contribution
Traffic Prediction & Management	Deva Priya [1], Hossein [2], Preveze [3]	Traffic forecasting, adaptive RTO, and protocol switching to improve SDN performance.
Resource Allocation & Optimization	Chen [4], Somayeh [12], Rasoul [14], Sagrika [15], Khawaja [42]	Dynamic priority, PSO, AI-based load balancing, and energy-efficient controller placement.
Healthcare & IoMT Applications	Ravi Ranjan Kumar [6], Yassine [7]	Medical image segmentation and SDN+NDN integration for healthcare networks.

DDoS Detection & Prevention	Anam [8], P. Karthika [17], Kumari [40], Sanjeetha [22]	Multi-layer defense and ML-based real-time DDoS detection in SDN.
Intrusion Detection Systems (IDS)	Halman [18], Waseem Asif [19], Sheik Abdullah [20], Mvah [21], Hazman [23], Rakine [24]	ML/DL-based IDS including SVM, KNN, ensemble, and deception-based techniques.
Privacy & Federated Learning	Rohit [10], Hossain [41]	Privacy-preserving learning using federated and self-supervised methods.
SDN Security & Challenges	Emmanuel [36], Jiang Ji [16], Tebbaa [44]	SDN vulnerabilities, cloud security, and implementation challenges.
Machine Learning Algorithms	Mienye [29], Salman [30], Anamika Jain [31], Raj Kiran [32]	Decision Tree, Random Forest, XGBoost, AdaBoost improvements.
Advanced Threat Detection	Ibrahim [38], Yan [39]	Hybrid ML (LSTM, KNN, LR) and RF-based encrypted traffic classification.
Emerging Networks (5G/6G, Cloud)	Pablo [13], Khalid [34]	Security and optimization in 5G/6G and network slicing using DL models.
Simulation & Tools	Alotaibi [25], Roseline [26], M. Said [37]	Mininet, ONOS setup and datasets like NSL-KDD, InSDN for experimentation.
Energy & Controller Optimization	Marjan Keramati [11]	Backup controller placement using NN and meta-heuristics.
Routing & Traffic Engineering	Amine Tcherak [27], Shahirar [45], Kumar [46]	RL-based routing and priority-based traffic engineering in SDN.
Applications of ML (Other Domains)	Farida Brahimi [28], Anish Kumar [33]	ML for disease prediction (E-KNN) and construction optimization.
Future Trends & Challenges	Boutaba [43], Salau [35]	Intelligent networking, traffic classification challenges, and future research directions.

2.2 Research Gap

Although the modern methods of the traffic classification are developed, the major research gap is the effective traffic segregation and prioritization in SDN-based healthcare networks. Current methods fail to sufficiently consider the dynamic categorization of traffic based on critical, normal and non-critical and the priority treatment that should be assigned to each category. This weakness influences the provision of sensitive healthcare information and real-time applications on time. Thus, it requires a dynamic framework, which will guarantee efficient traffic segregation and prioritization to enhance network reliability, network security, and network

performance.

3 Healthcare Network Requirements

Healthcare networks must bring high speed, strong reliability, and robust security to effectively support critical medical services. Healthcare networks require *firstly*, Quality of Service (QoS) requirements to handle large medical data, high bandwidth, low latency for real-time services like patient monitoring, telemedicine and medical imaging, here strong reliability and security must be ensured to get uninterrupted care and protect patient privacy [10]. The user demands for high availability of data, with guar- anteed support as a when patient demand for any medication or treatment. Here safety is most important concern which makes authentic clinical decisions in patient health. Secondly, the network must be reliable to prevent disruptions by making network traffic prioritization and enhance security while protecting patient privacy. Here healthcare system required safe and efficient care in hospitals and clinics [11]. Lastly, Security and Privacy constraints relevant to these Healthcare networks are required to adhere to regulatory frameworks such as the Health Insurance Portability and Accountability Act (HIPAA), the General Data Protection Regulation (GDPR), and analogous regional healthcare data protection legislations. Furthermore, there exist significant challenges associated with the management of heterogeneous medical traffic.

4 Traffic Classification and Management in SDN

Traffic segregation in healthcare networks is important because different types of data need different levels of service. During traffic classification identification of traffic type, priority level, flow level attribute like protocol, port number, packet size, and timing behavior and knowledge of policy rule with ML model classification techniques must require.

4.1 Traditional Vs Modern Traffic Classification Techniques

Traditional SDN based traffic classification mechanisms include, *port based, protocol based, deep packet inspection (DPI), flow based statistical classification as well as rule based, policy based and signature-based classification.* They may work well in a fixed environment, but tend to scale, be flexible and impractical to manage when their use is expanded to a highly dynamic and heterogeneous network environment

On the other hand, the sophisticated SDN traffic classification techniques address the vulnerability of the old rule-based and signature-based schemes using programmability, cunning and flexibility. They consist of machine learning and deep learning- based traffic classification, flow-level traffic intelligent classification through SDN controllers, programmable SDN data plane-based traffic classification, intent-based traffic classification, reinforcement learning-based traffic classification, and hybrid frameworks of traffic classification. Recent methods are stronger and more scalable in classification, learning the behavior of traffic and reacting to the variation of network conditions. Table

2 shows as traffic classification techniques.

Table 2. Traffic Classification Techniques - Traditional vs Modern with their features

Feature	Traditional SDN Traffic Classification	Modern SDN Traffic Classification
Method	Rule-based and Static	AI-based and Dynamic
Methods Used	Port-based, Protocol-based, DPI, Flow-based, Signature-based	Machine Learning (ML), Deep Learning (DL), Reinforcement Learning (RL)
Traffic Classification	Manual / Predefined Rules	Automated Intelligent Traffic Classification
Data Plane	Non-Programmable	Programmable Data Plane
Decisions	Policy-Based	Intent-Based and Adaptive
Flexibility	Inflexible	Dynamic and Flexible
Scalability	Low Scalability	High Scalability
Environmental Adaptability	Non-Adaptive Environment	Adaptive Environment
Performance	Lower performance for complex traffic	Better performance for large-scale complex traffic
Learning Ability	No learning capability	Learns from data and improves over time
Hybrid Support	Not supported	Supports hybrid approaches

The paper also elaborates different advantages of new traffic classification techniques over the old ones, including their *effectiveness in encrypted traffic, a reduced reliance on manual setup, a higher scale and adaptability as well as supporting real- time healthcare items.* Sophisticated techniques of packet-level and flow-level traffic analysis are also becoming influential in network design, security implementation, scalability and performance, especially in SDN-enabled healthcare settings that use Machine Learning. Here, *flow-based analysis* helps detect *behavioral and volumetric attacks,* and Packet-based analysis helps detect payload-level attacks. Packet Inspection does not rule due to per-packet processing overhead. The defined procedure tells us that every packet must be captured, Parsed, and checked for header and payload. In high-speed networks, this means millions of operations per second. Traffic analysis in SDN can be broadly categorized into *packet-based and flow- based approaches.* Table 3 show that each technique differs in terms of granularity, computational overhead, scalability, and suitability for modern encrypted networks.

Table 3. Flow-based vs packet-based traffic analysis

Feature	Flow-Based Analysis	Packet-Based Analysis
Data Granularity	Summarized data (flows)	Full detailed packets
Performance	Fast and lightweight	Slower and resource-heavy
Storage Requirement	Low storage needed	Very high storage needed
Visibility	Only metadata (no payload)	Complete data (header + payload)

Scalability	Works well for large networks	Hard to scale
Security Analysis	Detects patterns/anomalies	Deep inspection of attacks

4.2 Flow-based vs packet-based traffic analysis

Network design, security, scalability, performance, and modern technologies are influenced by both flow-based and packet-based traffic analysis. Packet-level and flow-level traffic analysis, when applied through intelligent techniques, significantly influence network design, security enforcement, scalability, and performance, particularly in SDN-enabled healthcare environments. Here, flow-based analysis helps detect behavioral and volumetric attacks, and Packet-based analysis helps detect pay-load-level attacks. Packet Inspection does not rule due to per-packet processing overhead. Every packet must be captured, parsed, and checked for header and payload. In high-speed networks, this means millions of operations per second.

4.3 Importance of traffic segregation and prioritization

Traffic segregation and prioritization are critical for ensuring secure, efficient, and reliable network performance—especially in SDN-enabled healthcare environments where sensitive patient data and real-time applications must be protected and delivered without delay.

Traffic segregation *Enhanced Security* which prevents the lateral entry of cyberattacks, *Regulatory Compliance* healthcare networks must comply with HIPPA and similar standards here segregation helps to control strict access control, *Easy to find fault* from any segment of the network, *performance get increases* by implementing different types of traffic-like voice, video, IOT sensor which can be used to be separated to avoid congestion and interference. Here *scalability of traffic gets increased* due to reconfiguring networks into the separate segments.

OpenFlow updates its routing tables entry so the controller can change routes, control traffic, and apply rules whenever needed. When processing an input packet, the switch evaluates it against the input conditions outlined in each routing table entry. If a match is found, the switch executes the instructions specified in the second part of that entry. Instructions may encompass actions such as forwarding the packet to a designated output port, discarding the packet, forwarding the packet to the controller, and more [12].

Figure 3 shows that instead of waiting for packets to arrive and then deciding routes (reactive mode), the controller installs flow rules in advance (proactively) into switches. This reduces delay and improves performance.

Components in the Diagram: Controller (center), Switches (s1, s2, s3, s4), Flow Tables (top boxes). Each switch has rules like: 128.119.0.0/16 Output Port 1, 128.120.0.0/16 Output Port 2 Each row has two fields: Match Defines the condition (usually destination IP prefix) Action What the switch should do when the condition matches. When a packet arrives at the switch: The switch checks the **destination IP address**. It compares

it with the **match fields**. If a match is found the corresponding **action is executed**.

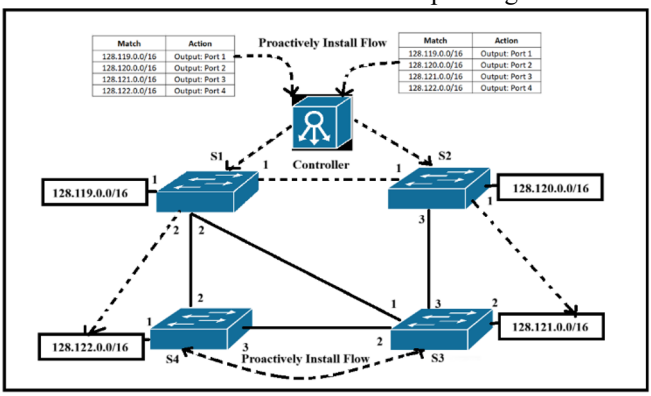

Fig 3. Proactive SDN Flow Installation

Here, p*roactive SDN flow installation* is an approach in which the SDN controller preinstalls flow rules (match–action entries) into the switch flow tables before any actual data traffic arrives. This study focuses on the basic characteristics of proactive flow installation: *first,* the controller installs rules in advance; *second*, there is no initial controller-switch interaction during data forwarding; *third*, there is lower control-plane overhead; and *lastly,* improved scalability and deterministic behavior.

In contrast, here there is no concern about *reactive flow installation,* which is another approach in which flow rules are installed *only after a switch receives the first packet of an unknown flow. However, this thought also moves a new research area where after packet receiving towards switch or controller can get examined based on reactive approach.* If the switch doesn't find a matching rule in its flow table, it sends a Packet-In message to the SDN controller. Here, the controller examines the packet, decides where it should go, and then adds the right flow entry back into the switch. This study also focuses on Rules installed on demand; Initial packet incurs controller delay, higher control plane interaction, and is suitable for dynamic and unpredictable traffic.

4.4 Network slicing

Network slicing is a fundamental technology in modern Network slicing that enables the creation of multiple virtualized and independent logical networks, referred to as slices, over a shared physical infrastructure [13]. These slices act as separate virtual channels, allowing sensitive patient data to be kept apart from less important traffic. This separation helps improve security, performance, and the overall quality of service in healthcare networks.

The importance of network slicing moves towards like in healthcare that is the data is not the same for all applications. Some data can be life-critical where urgency is required, where some areas can manipulate some amount of tolerance/delay. Also Telecommunications play a vital role in 5G and future 6G networks. These networks enable the creation of multiple virtualized and autonomous systems, supporting advanced services and applications.

Logical networks, which are sometimes referred to as slices, are traversing a com- mon physical infrastructure. Network Slicing allows the development of different virtual channels, which isolate sensitive patient data. Example is Critical Care Slice, Medical Imaging Slice, Administrative & EHR Slice and IoT & Wearable Devices Slice.
This is achieved through priority mechanisms to make sure that emergency notifications with exact resource allocation and telemedicine sessions are timely provided and important messages are not postponed. A combination of these methods helps to increase reliability, security, and health care communications systems. The most important part of the network is congestion is also reduced due to this network slicing. The network slicing-based mitigation strategy was implemented in the software data path using Software-Defined Networking (SDN) and Network Function Virtualization (NFV). A network-slicing-based mitigation strategy implemented in the software data path is presented, along with a use case that demonstrates its effectiveness against cyberattacks [13]. It is promising because it provides isolated virtual networks tailored to service needs. Nonetheless, inflexible slicing schemes are unable to keep up with high-frequency traffic variations and the mobility of UAVs, leading to poor resource utilization and Quality of Service (quality of service) deterioration.
Let us consider a hospital network where assemblies of telemedicine and patient-monitoring devices generate traffic constantly. An intelligent policy can dynamically scale flow collection at the acquisition point's entry points, so that, in the event of a slight surge in traffic from wearable devices, the policy immediately identifies it as an anomaly. Such quick detection is used to spot threats in the environment, such as denial-of-service attacks or malfunctioning machines, and prevent them from interfering with patients' care. The network is secure and reliable for critical healthcare services, as it responds in real time.

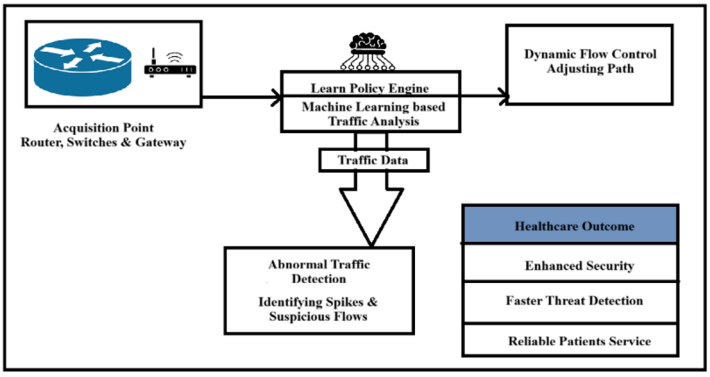

Fig. 4. Learned Policy Engine for dynamic Flow control in Healthcare Network

Figure 4 shows arrows flowing from acquisition points into the learned policy en- gine, branching into dynamic flow control and abnormal detection, and finally connect- ing to healthcare outcomes such as security and reliability. As networks grow and user numbers and traffic increase, traditional networks face significant management challenges due to the use of proprietary hardware devices such as switches, routers, and load balancers. This complexity hinders effective network management.

In Software-Defined Networking (SDN), Controllers implement load-balancing strategies to manage high volumes of network traffic and congestion [14]. The latency threshold constraint is a fundamental rule that must be satisfied when placing controllers in an SDN. It works on the principle of first calculating the Shortest Path Delay, and second calculating the Haversine Distances. Third, Threshold Enforcement, Pur- pose, and Equations [15].

4.5 Role of SDN in dynamic traffic control

Software Defined Networking (SDN) plays a crucial role in enabling dynamic, adaptive quality of service enforcement and intelligent traffic control by decoupling the control plane from the data plane and providing a global view of the network. In this paper, the focus is moved to Traffic Prioritization and quality of service Enforcement, and traffic can dynamically be classified as per the priority level. When network traffic is used in adaptive-based learning, here the role of reactive and proactive traffic control is greatly increased. Reducing latency by improving the load balancing compared to traditional methods, and introducing reinforcement learning for dynamic traffic allocation [16].

5 Cyberattack Detection in SDN

SDN has challenges with controller vulnerability, flexibility, and hardware security. But distributed denial-of-service (DDoS) attacks pose a serious threat to the SDN. Transmission Control Protocol (TCP)- synchronized (TCP-SYN) floods, a common cyberattack that can harm SDNs, can deplete network resources by opening an excessive number of illegitimate TCP connections. This OpenFlow port uses a statistical-based architecture for machine learning (ML)- enabled TCP-SYN flood detection [17]. **Detection in Healthcare SDN Environments:** Cyberattack detection is a critical requirement in Software-Defined Networking (SDN), particularly within healthcare infrastructures that manage sensitive and mission-critical patient data. Common network-based attacks, including Distributed Denial of Service (DDoS) where attacker disturb the server or network traffic in a real time environment, IoT & Wearable Devices Slice where unauthorized which is collected from sample may get temper, flooding, and spoofing, can severely disrupt healthcare services or lead to unauthorized data disclosure [18]. *Traditional signature-based Intrusion Detection Systems (IDSs)* are limited to identifying previously known attack patterns and are generally ineffective against novel or evolving threats [19]. In contrast, *machine learning–based IDSs* analyze traffic behavior and learn underlying patterns from collected data, which enables it to adapt according to change in network conditions and detect previously unseen attacks. These capabilities make ML-driven detection mechanisms clearer and more suitable for large-scale healthcare SDN deployments, while maintaining high levels of reliability and security which is essential in network traffic [20].

ARP Spoofing in Healthcare Oriented SDN Environments: A significant security threat in Software-Defined Networking (SDN) is ARP spoofing, particularly in

healthcare systems where private patient data is shared on a regular basis. It may be of different sources yet the intention is to be safe and arrive at and when it is mandatory to the doctor or patient. In this form of attack, a malicious host uses the ARP protocol by sending fabricated address mappings, thus fooling switches and the SDN controller. They may result in violation of data confidentiality, more communication delays, and critical medical services may be disrupted. To overcome these risks, SDN controllers are able to utilize the centralized control of their nature to identify anoma- lous traffic of the network and implement security policies on the fly. The detection accuracy and flexibility is also improved by the fact that the machine learning tech- niques are integrated into controller logic, which are highly adaptable to dynamic and real-time healthcare network settings [21].The basis for real-time DDoS Detection and Mitigation is the threshold for the type of application transmitting data to a specific switch, determined by a machine learning (ML) model, and whether the application traffic constitutes DDoS traffic. Once a de- tection occurs, only the application type that transmits the DDoS traffic can be blocked, and other authentic applications should be permitted to transmit network traffic without interruption. The use of a dynamic threshold based on current network traffic will help detect DDoS efficiently [22].

Brute Force Attack: A brute-force attack is a trial-and-error method in which an attacker repeatedly tries different passwords or keys until the correct one is found. It relies on computational power rather than intelligence and is commonly used against login systems. This attack is among the top 10 of the Open Web Application Security Project (OWASP).

5.1 Comparison of signature-based vs ML-based detection

Signature-based detection identifies attacks by matching network traffic or system behavior against a database of known attack signatures. It works well in antivirus pattern-matching, is very effective against known attacks, and requires frequent signature updates. An example is detecting a known SQL injection or malware hash using predefined rules [23].

ML-based detection uses algorithms that learn patterns from data to identify normal and abnormal behavior, including unknown attacks. It works well at detecting zero-day and evolving attacks. Uses features such as flow statistics. Requires training data and computational resources. Examples include detecting unusual traffic behavior indicative of a new type of DDoS or brute-force attack [24].

Table 4. Comparison of signature-based vs ML-based detection

Feature	Signature-Based Detection	ML-Based Detection
Detection Method	Matches predefined signatures	Learns patterns from data
Intelligence	It is Static (rule-based)	It is Adaptive (learning-based)
Zero-Day Attacks	It Cannot detect	It Can detect (anomalies)

Accuracy	High for known attacks	High for both known + unknown (if trained well)
Speed	It is very fast	It is Moderate (depends on model complexity)
Updates Required	Frequent manual updates	Model retraining needed
Complexity	Simple to implement	More complex (training + tuning)
Data Requirement	No training data needed	Requires large datasets
Use in SDN	Basic intrusion detection	Advanced traffic classification & anomaly detection

6 Simulation Tools and Experimental Environments

The open network operating system (ONOS) is a Java-based SDN operating system. A network topology consisting of various networking devices and links can be emulated in Mininet. The ONOS interacts with networking devices emulated in Mininet through the OpenFlow protocol [25]. *Simulation tools* play an important role in testing and analyzing SDN-based healthcare networks. As shown in Table 5, various simulation tools are listed, with their primary use in healthcare SDN. Mininet is popular for building virtual network topologies for experimentation, whereas NS-3 is a more detailed simulation package that operates at the packet level. OMNeT++ is a versatile tool for modeling complex communication systems, and CloudSim is well-suited for examining health care services in the cloud. All these tools enable the researcher to test performance, reliability, and scalability without installing expensive software.

Table 5. Simulation Tools and their primary use in healthcare SDN

Tools	Primary Use	Strength in Healthcare
Mininet	Emulates SDN topologies with real controllers	Quick prototyping and realis- tic testing
NS-3	Packet-level network Simulation	Detailed protocol analysis and performance evaluation
OMNeT++	Modular simulation of communication systems	Flexible modeling of complex healthcare scenarios
CloudSim	Simulation of cloud computing environments	Studies resource allocation for healthcare cloud services

This research will advance the experimental environment setup using Mininet to simulate the SDN topology, with the Ryu controller serving as the centralized control plane for traffic classification and segregation. It will use Python (3.x) to generate custom Ryu applications, iperf, or Wireshark to generate, monitor, and validate traffic. The topology is made up of numerous hosts that are symbolic of healthcare equipment, hospital information systems, and guest users, linked together through SDN-enabled switches (Open vSwitch). In this research, the study will begin by establishing a small network

and attempt to trace a small number of network traffic flows.
A single controller system will be transferred to multiple Ryu controllers using the iperf and Wireshark tools. All the generated real-time data, which was taken in the network traffic, was entered into a machine learning module to enhance the classification of traffic. This classification became possible through the identification of flows by IP addresses, port numbers or flow analysis by machine-learning. The classes were assigned to different queues or priorities in the switch and the Ryu controller was inserted dynamically with OpenFlow rules to implement Quality of Service (quality of service) policies.

6.1 Simulation Framework

This organized workflow allows to manage traffic and to retain stable network performance on SDN-based healthcare settings.

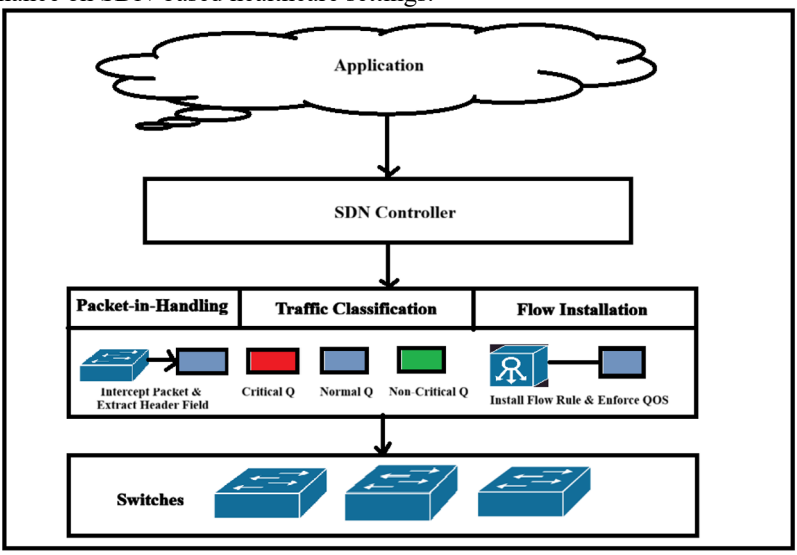

Fig 5. Efficient Traffic Management in SDN-Based Healthcare Network

Figure 5 shows SDN-Based Traffic Management for Healthcare Networks. The diagram shows a centralized SDN structure where the controller classifies traffic received into critical, normal and non-critical traffic. The controller dynamically configures flow rules in the network switch with the right QoS restrictions, to achieve low latency and high priority to critical medical flows. The processing is initiated by Packet- In processing in which header features are decoded and then the traffic is classified into priority-based queues. Lastly, in Flow Installation, the controller makes use of OFP Flow Mode messages to insert flow rules in switches and implement quality of service with queue identifiers or meter tables. Machine Learning Techniques in SDN

In order to provide a complete evaluation and a good performance, an ML-based

framework is created to train, test, and evaluate nine classification algorithms. The presence of machine learning classification models is essential in real-time atypical traffic and cyberattacks detection. Random Forest, Support Vector machine (SVM), Artificial Neural Networks (ANN) and Gradient Boosting are some of the models that are popular in the accurate classification of normal and malicious healthcare data. The framework offers a balanced set of interpretable models and advanced ensemble techniques by sustaining a variety of classifiers. This method will allow to understand better the accuracy, flexibility and computational efficiency of SDN-based medical traffic classification.

Table 6. Comparison between simple, interpretable models & advanced ensemble techniques.

Algorithm	**How it Works**	**Strength in SDN**	**Limitation in SDN**
Logistic Regression [26]	Uses linear equation to predict probability of a class (e.g., normal vs attack traffic).	Simple, fast, interpretable; good for binary classification tasks.	Limited for complex, non-linear traffic patterns.
Naive Bayes [27]	Probabilistic model assuming independence among features.	Very fast, low memory usage, good for real-time detection.	Assumption of independence is often unrealistic in SDN traffic.
KNN (K-Nearest Neighbors) [28]	Classifies flows based on similarity to nearest neighbors in feature space.	Easy to implement, effective for small datasets.	Computationally heavy for large SDN traffic; sensitive to noisy data.
Decision Tree [29]	Splits data into branches based on decision thresholds.	Clear decision rules, interpretable, fast training.	Can overfit with complex traffic unless pruned.
Random Forest [30]	Ensemble of multiple decision trees with majority voting.	High accuracy, robust to noise, handles large datasets well.	Slower than single trees; less interpretable.
XGBoost [31]	Gradient boosting framework optimized for speed and accuracy.	Very high accuracy, scalable, widely used in intrusion detection.	Requires careful tuning; higher resource consumption.
AdaBoost [32]	Boosting technique combining weak learners into a strong classifier.	Improves accuracy, adapts to difficult cases.	Sensitive to noisy data and outliers.
SVM (Linear) [33]	Finds a hyperplane to separate classes in linear feature space.	Effective for linearly separable traffic; strong theoretical foundation.	Struggles with non-linear traffic patterns.
SVM (RBF Kernel) [33]	Uses radial basis function to capture non-linear boundaries.	Handles complex traffic flows; high accuracy.	Computationally expensive for large-scale SDN datasets.

6.2 Probabilistic Models

One of the most common and easy-to-use machine learning tools in Software-Defined Networks (SDN) is Logistic Regression, which is useful for classification tasks. It approximates the likelihood that a specific network flow is of a given type, e.g., normal or malicious traffic. Due to its low computational cost and simplicity, Logistic Regression can be applied to real-time intrusion detection in healthcare SDN environments. Nonlinear traffic patterns, however, can be tricky to model with, and could prove harder to handle with it [26]. Logistic regression works on the core idea of computing a linear combination of input features, passing it through a Sigmoid (Logistic) function, and finally outputting a value between 0 and 1 (a probability). Here in this research, linear regression will give results based on traffic class segregation, like critical, normal, and non-critical, so that the result is based on class-wise predictions, confusion matri- ces, and classification reports for the concern model.

Naive Bayes

Naive Bayes is a lightweight machine learning algorithm often used for intrusion detection in Software-Defined Networks (SDN). It works by applying probability rules to classify traffic as normal or malicious, assuming independence among features. Naive Bayes is applicable to real-time monitoring in healthcare SDN because it is fast to train and requires low memory. Nevertheless, its independence assumption can reduce accuracy when features are highly correlated, making it ineffective in complex traffic patterns. The Naïve Bayes classifier, as applied in this study, segments traffic into the Critical, Normal, and Non-Critical classes. The evaluation outcomes are presented as class-wise predictions, confusion matrices, and classification reports for the dis- cussed model [27].

6.3 Distance-Based

KNN (K-Nearest Neighbors) An effective, but easy to use machine learning algorithm, K-Nearest Neighbors (KNN), is commonly used in traffic classification and intrusion detection in Software- Defined Networks (SDN). It works by finding the closest ones in the dataset and label- ing it after majority voting. KNN is simple to use when the data is well-organized and it can attain high accuracy. However, it is ineffective and slow in processing the large amounts of traffic data, which is characteristic of SDN healthcare systems [28].

6.4 Tree Based

Decision Tree is also an ancient machine learning and this algorithm has been used in Software-Defined Networks (SDN) in classification problems. It works by breaking the information into branches created according to feature values and making the decisions in a sequential manner until a final class is identified. The model is simple to comprehend and interpret, which helps detect the normal and malicious traffic within SDN healthcare setups**.**

The tendency of decision trees to overfitting is especially high with complex and varied traffic patterns, since decision trees can produce too specific decision rules. To counter this drawback, methods like tree-pruning and the ensemble-based methods are

usually used to enhance the generalization and overall prediction performance [29].

Random Forest Applicability of Random Forest in Healthcare SDN: The random Forest model is used to create more powerful decision trees to classify data and minimize variation in the model. It is particularly adapted to SDN intrusion detection and traffic classification because of its ability to handle high-dimensional and noisy network traffic data. Moreover, the ensemble aspect of the algorithm reduces overfitting and hence more reliable and consistent predictions. This has led to the high adoption of Random Forest in healthcare SDN settings, where reliable and robust security measures are required [30].

XGBoost

XGBoost is a superior ensemble learning model that builds upon gradient boosting and is common in classification in Software-Defined Networking (SDN) applications. The model is constructed sequentially, whereby each decision tree is trained to address the mistakes of the previous decision trees. The consequence of this learning process which is optimized in nature is a high classification accuracy and the ability to handle complex data patterns easily. XGBoost is especially useful in SDN-based healthcare networks to identify the abnormal traffic and ensure the safety of transmission of data. Even though the model is quite reliable in intrusion detection and traffic analysis, its computational needs are also quite high as they are associated with its scalability and robustness [31].

AdaBoost

AdaBoost for Intrusion Detection in SDN: AdaBoost is an ensemble learning algorithm which builds a powerful classifier in iterations. Gathering together several weak learners, which are usually shallow decision trees. This algorithm can be used with intrusion detection with networking (SDN) environments. Gradually providing a greater significance to traffic flows that are hard to classify. As a result, AdaBoost is more efficient in identifying complex or weak attack patterns that are otherwise. be disregarded by the individual classifiers. The main advantage of AdaBoost is that it is able to improve. Precision without the need to have too complicated models. However,it seems to be susceptible to noisy data. and outliers, may influence its performance in large healthcare SDN settings [32].

6.5 Margin Based

SVMs with a linear kernel are among the most common classification algorithms used in Software-Defined Networks (SDN). It operates by identifying the optimal hyperplane that separates normal and malicious traffic. The linear version is easy to compute and works well when the data can be linearly separated, making it appropriate for real-time intrusion detection in healthcare SDN settings. Nonetheless, it can perform poorly under complex, non-linear traffic patterns [33].

7 Comparative Analysis of Existing Survey works

Table 7. Comparative Analysis of Various ML Models

Ref	SDN in Healthcare	Traffic Management	Attack Detection	Simulation / Emulation	ML Model Focus
1	✓	✓	-	-	✓
2	-	✓	-	-	-
3	✓	-	-	✓	-
4	-	✓	-	-	-
5	-	✓	-	✓	-
6	-	-	-	-	✓
7	✓	✓	-	✓	-
8	-	-	✓	✓	✓
9	-	-	✓	✓	✓
10	✓	-	-	-	✓
11	-	✓	-	-	✓
12	-	✓	-	✓	✓
13	-	✓	✓	✓	✓
14	-	✓	-	-	✓
15	-	✓	-	✓	✓
16	-	-	✓	✓	✓
17	-	-	✓	✓	✓
18	✓	-	✓	✓	✓
19	✓	-	✓	✓	✓
20	✓	-	✓	✓	✓
21	-	-	✓	✓	-
22	-	-	✓	✓	✓
23	✓	-	✓	-	✓
24	-	-	✓	-	✓
25	-	-	-	✓	-
26	-	-	✓	-	✓
27	-	✓	-	✓	✓
28	✓	-	-	-	✓
29	-	-	-	-	✓
30	-	-	-	-	✓

31	✓	-	-	-	✓
32	-	-	-	-	✓
33	-	-	-	-	✓
34	-	✓	-	✓	✓
35	-	✓	✓	✓	✓
36	-	-	✓	-	✓
37	-	-	✓	✓	✓
38	-	-	✓	-	✓
39	-	✓	✓	-	✓
40	-	-	✓	✓	✓
41	-	-	✓	✓	✓
42	-	✓	-	✓	-
43	-	✓	✓	-	✓
44	-	-	✓	-	✓
45	-	✓	-	✓	-
46	-	✓	-	✓	✓

Table 7, Comparative Analysis of Various ML Models, indicates the use of SDN in healthcare system, which suggests scenarios to manage traffic, some researcher's undertakes network attack detection with different type of attacking network, some works describes about the simulation infrastructure that should have been used to execute attack on real time network and the last some paper contribution on the focus area of ML.

8 Open Challenges and Research Gaps

Scalability Challenges Issue: Most traffic segregation methods are tested only on small networks. They do not work well in large data centers, healthcare IoT network s, or high-traffic environments. The classification of network traffic has become increasingly critical due to the rapid growth in internet users and the dynamic, encrypted nature of modern network traffic. Traditional approaches, such as identifying traffic by port numbers and payload inspection, are no longer effective due to these characteristics [35].

Datasets Availability: High-quality datasets are a key success factor in the design and testing of ML algorithms. However, public and standard SDN datasets are limited.

The first way is to implement testbeds, i.e., establish real-world network environments with a specific purpose to collect data and gain experience with them. The second way is using standard network simulators or emulators, e.g., Mininet, NS3, or EstiNet [36]. **Testbeds are expensive and can be difficult to establish** due to the resources and facilities required. On the contrary, simulators and emulators may introduce

simplifications and assumptions that fail to accurately represent the realities of real network environments. Synthetic datasets: Data that is outdated (e.g., KDD 99 NSL-KDD). Re- search Gap: No healthcare-specific SDN traffic datasets provide a good representation of Emergency flows, Encrypted traffic, and Zero-day attacks [37].

The challenge lies in the generalization of machine learning models, where models trained in one network environment often fail to perform effectively in a different setting. This is mainly due to their strong dependence on network topology and traffic behavior patterns.

Limited work on cross-domain generalization is still limited and requires further exploration. Generalized traffic models can improve robust traffic segregation by leveraging advanced approaches such as transfer learning, federated learning, and domain adaptation. [38].

Encrypted Traffic Processing Issue This is due to the increasing use of TLS or HTTPS encryption. Payload Checking is no longer possible. Traffic encryption techniques facilitate cyberattacks to hide their presence and activities [39].

Dynamic and Adaptive Traffic Policies Issue Current SDN systems mostly use fixed (static) traffic rules, but real network conditions keep changing. This makes it difficult to manage traffic efficiently in a dynamic environment [40].

Multi-Class and Fine-Grained Traffic Segregation Problem: Most existing research focuses only on two traffic types—normal and attack. Limited Traffic and Attack Differentiation:

Most of the solutions that are in place do not offer much provision in terms of handling various types of traffic classes such as emergency, video, control and malicious traffic. These types of traffic are either not given due attention in a number of studies or they are given inadequate attention. Additionally, the literature tends to provide no clear separation of various types of cyberattacks, which reduces the efficiency of traffic-sensitive security and mitigation measures [41].

Lack of Standardized Evaluation Metrics: The available literature utilizes numerous evaluation criteria, and hence, there is no uniformity in performance evaluation. The majority of work is based on classic metrics of networks like end-to-end latency, packet loss rate, throughput, bandwidth available, and jitter. Although these parameters provide effective usage in measuring the overall network behavior, they fail to provide sufficient insight into the performance of underlying algorithms. Further, even the lack of standardized experimental designs complicates the possibility of meaningful comparison of various studies [42]. ML-Based Evaluation Metrics (Model-Centric Evaluation) like Accuracy, Precision, Recall, F1 Score, Confusion Matrix, Regression Metrics (MSE, RMSE, MAE). These metrics do not directly reflect network performance (e.g., a model with high accuracy may still degrade latency if computational overhead is high). Results are often hard to compare across different ML tasks unless standardized metrics and datasets are used [43].

Single-point failure of the controller: The SDN controller is the network's central brain, responsible for managing flow rules, policies, and traffic decisions. If the controller fails, the entire network can lose visibility and control, leading to service disruption, traffic mismanagement, or even a complete network outage [44].

9 Future Research Directions

9.1 Traffic Segregation

A majority of the traffic segregation schemes are tested on small-scale or laboratory network topologies. It is required to create traffic segregation techniques that are scalable to large data centers, healthcare IoT networks, and high-throughput systems. By creating algorithms capable of responding to different types of traffic loads and structures, it will be more desirable to stop relying on small test networks. Lack of standardized, current datasets of encrypted, heterogeneous, and healthcare-oriented SDN traffic. Stop binary classification (normal vs. attack) and go towards multi-class segregation (emergency, video, control, attack). Add a finer-grained distinction between attack types to provide more specific mitigation measures. Train ML/DL models to classify encrypted traffic without inspecting payloads based on flow-based and statistical features, along with side-channel information. Check out privacy-saving strategies that balance security and traffic exposure [45].

9.2 Intelligent-Based Information in SDN

Following traffic segregation, agentic AI can dynamically control each traffic class using multi-agent reinforcement learning. Agents dynamically change policies in situ, giving priority to emergency flows, mitigating attacks, and maximizing video/control traffic quality of service. Federated and transfer learning through these policies generalize domains, and explainable AI offers transparency and trust in choices. This makes the SDN network self-learning, adaptive, and resilient [45]. It is at the end of this study that one learns of the need for AI-driven SDN controllers that can self-learn and self- adapt. Use explainable AI (XAI) to enhance the trust and interpretability of traffic segregation decisions [46].

10 Conclusions

This paper discusses how SDN can be integrated with machine learning to enhance the identification of normal and malicious traffic by optimizing quality of service in modern network settings. The paper compares and contrasts four multi-class and binary classification scenarios before and after scaling and tests three types of linear, non-linear, and hybrid ML models using 5 major measures, including accuracy, F1 score, kappa score, ROC curve, and confusion matrix. The results emphasize that a more realistic analysis should be conducted when the number of SDN controllers is varied, and they indicate that future research should focus on multi-controller structures, controller replication and fault tolerance, the hierarchy of the control plane, and the agentic integration of AI.

References

1. Isravel, D.P., Silas, S., Kathrine, J.W., Rajsingh, E.B., Andrew, J.: Multivariate forecasting of network traffic in SDN-based ubiquitous healthcare systems. IEEE

Open Journal of the Communications Society 5, 1537–1550 (2024).
2. Zangoulechi, H., Babaie, S.: An adaptive traffic engineering approach based on retransmission timeout adjustment for software-defined networks. Journal of Ambient Intelligence and Humanized Computing 15(1), 739–750 (2024).
3. Preveze, B., et al.: SDN-driven Internet of Health Things: A novel adaptive switching technique for hospital healthcare monitoring system. Wireless Communications and Mobile Computing 2022(1), 3150756 (2022).
4. Chen, Y.-F., et al.: Adaptive traffic control: OpenFlow-based prioritization privacy strategies for achieving high quality of service in software-defined networking. IEEE Transactions on Network and Service Management (2025).
5. Barkire, D.I., et al.: QoSFlow: Prioritizing flows in software-defined networking for enhanced quality of service. In: 2025 27th International Conference on Advanced Communications Technology (ICACT), pp. 1–6. IEEE, Piscataway (2025).
6. Kumar, R.R., Priyadarshi, R.: Denoising and segmentation in medical image analysis: A comprehensive review on machine learning and deep learning approaches. Multimedia Tools and Applications 84(12), 10817–10875 (2025).
7. Sembati, Y., Naja, N., Jamali, A.: Software-defined named data networking for efficient healthcare content distribution. In: International Conference on Advances in Communication Technology and Computer Engineering. Springer, Cham (2024).
8. Rajper, A., Paraman, N.B., Marsono, M.N., Rajper, N.J., Hameed, H., Usman, M.: An efficient three-tier defense mechanism for mitigation of DDoS attack with port connection analysis in SDN. Scientific Reports 16(1), 3510 (2026).
9. Khanal, B., Kumar, C., Ansari, M.S.A.: Real-time anomaly detection framework to mitigate emerging threats in software-defined networks. Journal of Network and Systems Management 33(2), 26 (2025).
10. Kanauzia, R., et al.: A comprehensive survey on federated learning for privacy preservation in digital healthcare applications. Knowledge and Information Systems 68(1), 55 (2026).
11. Keramati, M., Mozayani, N.: Neural network and meta-heuristic-based method for backup controller assignment in SDN-smart grid. Telecommunication Systems 89(1), 5 (2026).
12. Azizi, S., Soltanaghaei, M., Ghaffarian, H.: Hierarchical traffic engineering with PSO: A path to efficient congestion management in SDN. Computing 107(2), 55 (2025).
13. Benlloch Caballero, P., et al.: E2E network slicing for enhanced cybersecurity, orchestration, automation and response in 5G/6G: The RIGOUROUS approach. Journal of Network and Systems Management 34(1), 22 (2026).
14. Farahi, R.: A comprehensive overview of load balancing methods in software-defined networks. Discover Internet of Things 5(1), 6 (2025).
15. Mohanty, S., Sahoo, B.: Energy-efficient controller placement in software-defined networks: A heuristic approach. Arabian Journal for Science and Engineering 50(23), 20285–20316 (2025).
16. Jiang, J., Wang, T., Zhang, S.: Research on traffic monitoring and security

protection of cloud computing network based on SDN. In: Proceedings of the 2025 International Conference on Big Data, Communication Technology, and Computer Applications, pp. 1–6 (2025).
17. Karthika, P., Arockiasamy, K.: Simulation of SDN in Mininet and detection of DDoS attack using machine learning. Bulletin of Electrical Engineering and Informatics 12(3), 1797–1805 (2023).
18. Halman, L.M., Alenazi, M.J.F.: MCAD: A machine learning based cyberattacks detector in software-defined networking (SDN) for healthcare systems. IEEE Access 11, 37052–37067 (2023).
19. Asif, M.W., Aqdus, A., Amin, R., Chaudhry, S.A., Alsubaei, F.S., Iqbal, S.: An efficient intrusion detection system using advanced machine learning techniques in software-defined networks (SDN) for healthcare system. IEEE Journal of Biomedical and Health Informatics, 3651–3664 (2026). doi:10.1109/JBHI.2025.3530563
20. Abdullah, A.S., Sunil, H.J., Nazmudeen, M.S.H.: A new model to evaluate signature and anomaly-based intrusion detection in medical IoT system using ensemble approach. SN Computer Science 6(4), 347 (2025).
21. Mvah, F., Tchendji, V.K., Djamegni, C.T., Kamhoua, C., Anwar, A.H., Tosh, D.K.: Deception-based IDS against ARP spoofing attacks in software-defined networks. In: Proc. IEEE Workshop on Computing, Networking and Communications (CNC), pp. 188–192. IEEE, Piscataway (2024).
22. Sanjeetha, R., Kanavalli, A., Gupta, A., Pattanaik, A., Agarwal, S.: Real-time DDoS detection and mitigation in software-defined networks using machine learning techniques. International Journal of Computing 21(3), 353–359 (2022). doi:10.47839/ijc.21.3.2691
23. Hazman, C., et al.: Intrusion detection approaches in healthcare systems: An overview. In: Reliability in Cyber-Physical Systems: The Human Factor Perspective, pp. 131–145. Springer (2026).
24. Rakine, I., Oukaira, A., El Guemmat, K., Atouf, I., Ouahabi, S., Talea, M., Bouragba, T.: Comprehensive review of intrusion detection techniques: ML and DL in different networks. IEEE Access 13, 104345–104367 (2025). doi:10.1109/ACCESS.2025.3579990
25. Alotaibi, E.: A tutorial on software-defined network emulation. Journal of Engineering Research 13(2), 666–673 (2025).
26. Ogundokun, R.O., Odusami, M., Sisodia, D.S., Awotunde, J.B., Tiwari, D.P.: A novel PCA-logistic regression for intrusion detection system. In: Garg, L., et al. (eds.) Key Digital Trends Shaping the Future of Information and Management Science. ISMS 2022. Lecture Notes in Networks and Systems, vol. 671, pp. 555–567. Springer, Cham (2023). doi:10.1007/978-3-031-31153-6_46
27. Tcherak, A., Loucif, S., Ould Khaoua, M.: Efficient routing for software-defined wireless sensor networks: A Naïve Bayes approach. IEEE Access 13, 207277–207302 (2025). doi:10.1109/ACCESS.2025.3638863
28. Brahimi, F., et al.: Enhanced K-nearest neighbors for smart cardiovascular disease

prediction in IoT system. Revue d'Intelligence Artificielle 38(4), 1047–1054 (2024).
29. Mienye, I.D., Jere, N.: A survey of decision trees: Concepts, algorithms, and applications. IEEE Access 12, 86716–86727 (2024).
30. Salman, H.A., Kalakech, A., Steiti, A.: Random forest algorithm overview. Babylonian Journal of Machine Learning 2024, 69–79 (2024).
31. Jain, A., Singh, A., Doherey, A.: Prediction of cardiovascular disease using XGBoost with OPTUNA. SN Computer Science 6(5), 421 (2025). doi:10.1007/s42979-025-03954-x
32. Kiran, R.J., Sanil, J., Asharaf, S.: A novel approach for model interpretability and domain-aware fine-tuning in AdaBoost. Human-Centric Intelligent Systems 4(4), 610–632 (2024). doi:10.1007/s44230-024-00082-2
33. Kumar, A., Sen, S., Sinha, S.: Support vector machine-based prediction model for the compressive strength for concrete reinforced with waste plastic and fly ash. Asian Journal of Civil Engineering 26(4), 1429–1447 (2025).
34. Aljonubi, K.: AI-driven network slicing for 5G UAV connectivity using SDN and graph attention networks in NS-3. Research Square Preprint (2026). doi:10.21203/rs.3.rs-8585548/v1
35. Salau, A.O., Beyene, M.M.: Software-defined networking-based network traffic classification using machine learning techniques. Scientific Reports 14(1), 20060 (2024).
36. Emmanuel, I.: Security threat mitigation in SDN. British Journal of Computer, Networking and Information Technology 9(1), 15–31 (2026). doi:10.52589/BJCNIT-XZPZ0SJZ
37. Elsayed, M.S., Le-Khac, N.-A., Jurcut, A.D.: InSDN: A novel SDN intrusion dataset. IEEE Access 8, 165263–165284 (2020). doi:10.1109/ACCESS.2020.3022633
38. Ibrahim, N., et al.: An optimized hybrid ensemble machine learning model combining multiple classifiers for detecting advanced persistent threats in networks. Journal of Big Data 12(1), 212 (2025).
39. Yan, X., et al.: High-speed encrypted traffic classification by using payload features. Digital Communications and Networks 11(2), 412–423 (2025).
40. Kumari, P., Jain, A.K., Sharma, A.: An adaptive framework for real-time detection and mitigation of DDoS attacks in software-defined networks. Peer-to-Peer Networking and Applications 19(1), 31 (2026).
41. Hossain, S., et al.: A privacy-preserving self-supervised learning-based intrusion detection system for 5G-V2X networks. Ad Hoc Networks 166, 103674 (2025). doi:10.1016/j.adhoc.2024.103674
42. Mehmood, K.T., Atiq, S., Sajjad, I.A., Hussain, M.M., Basit, M.M.A.: Examining the quality metrics of a communication network with distributed software-defined networking architecture. Computer Modeling in Engineering and Sciences 141(2), 1673–1708 (2024). doi:10.32604/cmes.2024.053903
43. Boutaba, R., Salahuddin, M.A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., Caicedo, O.M.: A comprehensive survey on machine learning for

networking: Evolution, applications and research opportunities. Journal of Internet Services and Applications 9, 16 (2018). doi:10.1186/s13174-018-0087-2
44. Tebbaa, K., et al.: Mitigating DDoS attacks in software-defined networks: A systematic literature review of machine learning and deep learning approaches. Iran Journal of Computer Science 9(1), 5 (2026).
45. Shahriar, M.S., et al.: Prioritized multi-tenant traffic engineering for dynamic quality of service provisioning in autonomous SDN-OpenFlow edge networks. arXiv preprint arXiv:2403.15975 (2024).
46. Kumar, H., et al.: Reinforcement learning-driven adaptive traffic routing role delegation for enhanced SDN network performance. In: 2024 IEEE World AI IoT Congress (AI-IoT), pp. 1–6. IEEE, Piscataway (2024).