International Journal of Research and Applied Technology

5(1)(2025) 1-24

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

Control Traffic in SDN Systems by using Machine Learning techniques: Review

Shavan Askar[1*], Diana Hussein[2], Media Ibrahim[1], Marwan Aziz Mohammed[2]

[1]Information System Engineering Department, Erbil Technical Engineering College, Erbil Polytechnic University, Erbil, Iraq

[2]Department of Computer science, college of engineering, knowledge university, Erbil 44001, Iraq

Email: *shavan.askar@epu.edu.iq

Abstract. Due to the rapid development of Internet and mobile communication technologies, which have spearheaded a fast growth of networking systems to become increasingly complex and diverse regarding infrastructure, devices, and resources. This requires further intelligence deployment to improve the organization, management, maintenance, and optimization of these networks. However, it is difficult to apply machine learning techniques in controlling and operating networks because of the inherent distributed structure of traditional networks. The centralized control of all network operations, holistic knowledge of the network, software-based monitoring of traffic, and updating of forwarding rules to enable the functions of (SDN) are factors that (SDN) has that facilitate the application of machine learning techniques. This study will make an extensive review of existing literature to be able to answer the research question of how machine learning techniques can be used in the context of the SDN. First, it gives a review of the foundational literature information. After this, a brief review of machine learning techniques is presented. We shall also delve into the application of machine learning techniques in the area of (SDN), with a sharp edge on traffic classification, prediction of Quality-of-Service (QoS), and optimization of routing and Quality-of-Experience (QoE) security management of the resource separately. Finally, we engage in discussions surrounding challenges and broader perspectives.

Keywords: Machine Learning (ML), Software-Defined Networking (SDN), Classifications of traffic, Management of resources.

1. Introduction

Recently, propelled by the rapid advancements in smart devices such as smartphones, smart cars, and smart home gadgets, coupled with the progress in network technologies like cloud computing and virtualization of the network, the volume of data traffic worldwide is experiencing exponential growth. To manage traffic allocation effectively and accommodate a significant the number of devices and networks are becoming increasingly diverse and complex. A typical network infrastructure comprises numerous devices, operates multiple protocols, and supports various applications. Wireless networks, for instance, employ a variety of the cells with distinct coverage of the transmission, level of the powers, and operational mechanisms. Furthermore, these networks utilize different technologies for the communication such as WiMAX, IEEE802.11 ac/ad, LTE and Bluetooth. The presence of diverse of the network infrastructure adds complexity to the networks, posing challenges in the efficiently organizing, managing, and maximizing of the network resources. One potential solution to address these challenges is to enhance the intelligence level in networks. In recent years, a Knowledge Plane (KP) approach has emerged as a promising avenue for tackling these issues[1] was introduced to enhance the Internet with automation, recommendation, and intelligence. This was accomplished through the implementation of the Machine Learning (ML) and cognitive methodologies. The central obstacle is in the intrinsic dispersion of the traditional network systems, for each node has localized insight and control over a rather minuscule part of the system. What we are looking at is executing control beyond the local domain while learning from nodes with only a partial view of the entire system[2]. Fortunately, recent advancements in (SDN) are poised to alleviate the challenges associated with knowledge acquisition. SDN design abstracts and separates the control plane from the data plane. Therefore, network resources can be abstracted and controlled centrally using the controller acting like the Network Operating System (NOS). This controller, therefore, dynamically programs the network and gets a holistic view of its operation by constant tracking and collection of real-time data about the state of the network, its configurations, packets, and flows.

Integrating machine learning techniques into the SDN is both pertinent and effective for several reasons. Firstly, computer technology advances with the Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs) likely give a chance to use state-of-the-art machine learning methodologies, including deep learning neural networks, in networking applications[3]. Secondly, data play a vital role in executing machine learning algorithms, given that they depend on data analysis. The centralized and hierarchical SDN controllers, with a complete network view, allow them to gather varied data in the network, enabling machine learning techniques. Thirdly, the SDN controller can derive intelligence by analyzing real-time and historical network data using machine learning techniques. This allows the controller to optimize the network, do data analysis, and automatically provision network services. The programmability of SDN comes with a realization of optimal implementation solutions—from network setup to resource allocation—using real-time, machine-learning-based network solutions[4]. The organization of the article is summarized in the following sections, as stated: Section I: Introduction of the study, Section II: Summary of the related literature, and after that, Section III expounds on the background knowledge on SDN. Section IV offers a succinct overview of prevalent machine learning techniques applied in SDN. Part V focuses on ML algorithm applications in SDN areas, including traffic classification, routing optimization, prediction of QoS/QoE, resource management, and security, while elaborating integrations in each. Part VI makes a comparison and analysis of the most related works. Finally, the study concludes in Section VII, with references provided in Section VIII (see Figure 1).

International Journal of Research and Applied Technology

5(1)(2025) 1-24

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

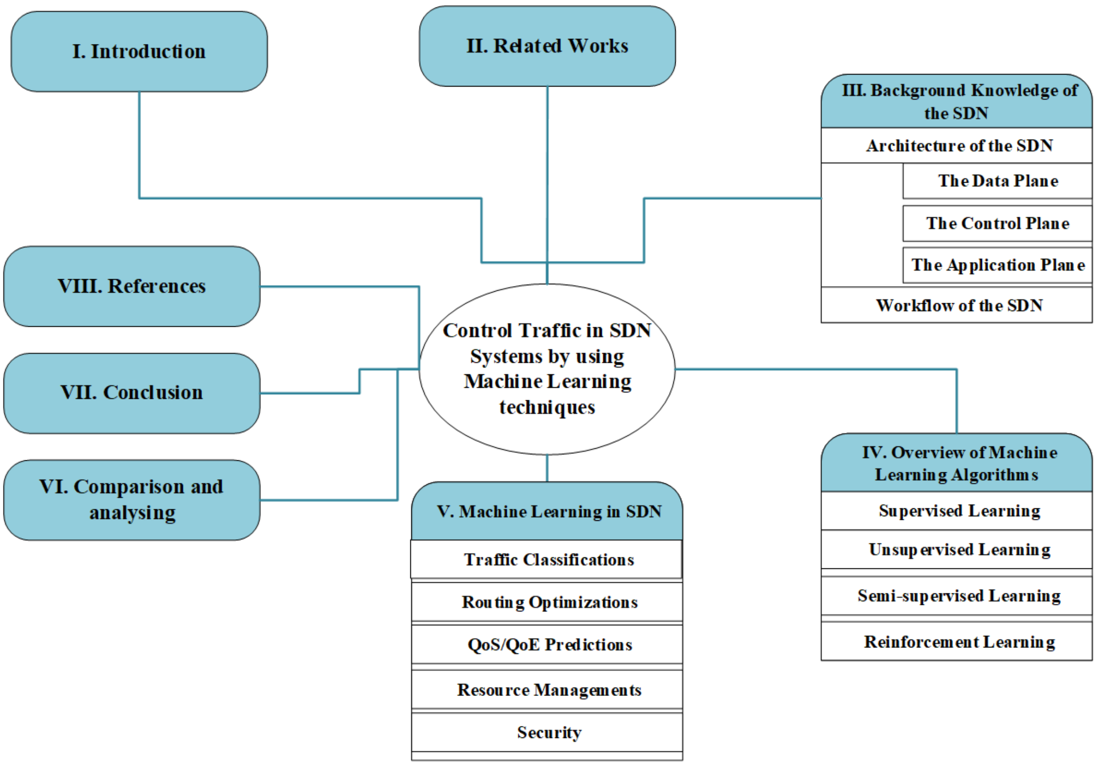

Figure 1. Diagrammatic of the subjects discussed in this paper.

2. Related Works

Machine learning has attracted considerable attention owing to its diverse array of applications. In the [5], a thorough examination of the application of machine learning methods in intrusion detection has been presented. The focus of [6] focuses on utilizing machine learning techniques for the classification of IP traffic. In [7] have explored numerous intricate learning challenges within Cognitive Radio Networks (CRNs) and reviewed the machine learning (ML) methods employed to address these issues. A survey has been conducted in [8] to investigate the application of machine learning approaches in addressing common challenges encountered in wireless sensor networks. In the study [9], the authors have emphasized the most advanced strategies leveraging Artificial Intelligence (AI) to enhance heterogeneous networks. Additionally, they have delved into the future research challenges within this domain. In [10] explored the use of machine learning (ML) and data mining (DM) techniques for the detection of cyber security intrusions. In [11] the authors illustrate deep research on machine learning techniques and their applications in self-organizing for cellular networks. They provided valuable classification and comparison of methods. A survey conducted in [12] has delved into the utilization of machine learning (ML) techniques to improve network traffic control. In [13] similarly focuses on a Machine Learning-based Intrusion Detection System (IDS). The main objective of [14] is initiative focuses on the use of machine-learning techniques and cognitive radio technology to achieve improved efficiency of wireless networks, with special emphasis on spectrum usage and energy consumption involved in modern technology-enabled operations. In [15], the study cantered

International Journal of Research and Applied Technology

5(1)(2025) 1-24

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

on employing neural networks to tackle challenges in wireless networks, spanning communication, virtual reality, and edge caching.

While machine learning techniques have found application across various domains, there is currently a gap in research specifically exploring their utilization in the realm of Software-Defined Networking (SDN). Traffic classification can indeed benefit from supervised learning methods like Support Vector Machines (SVM), neural networks, and decision trees [16]. In computer networks, obtaining labelled training data for supervised learning can be challenging due to the scarcity of adequately annotated network flow samples across various applications, compounded by the rapid emergence of new applications. As an alternative, unsupervised machine learning can be employed, where the learner is given unlabelled data. Unsupervised learning is commonly used for clustering tasks, where computers categorize data into distinct clusters based on similarities in feature values [17]. Indeed, semi-supervised learning, a fusion of unsupervised and supervised methods, is applicable to datasets containing both labelled and unlabelled data. This approach can leverage the benefits of both types of data to improve learning performance [18]. The strategy aims to overcome the challenges related to obtaining labelled data. Even if there are limited labelled instances available, leveraging a dataset primarily composed of unlabelled instances can still be viable. The labelled examples enable the mapping of identified clusters in the entire dataset to specific classes, thereby facilitating classification.

3. Background Knowledge of SDN

3.1. Architecture of the SDN

In recent years, SDN has been accorded great prominence, and the Open-Networking Foundation (ONF) is at the forefront of promoting the technology [19]. ONF is a non-profit organization that targets the promotion, standardization, and commercialization of Software-Defined Networking (SDN) (see Figure 2). ONF defines SDN as the separation of the control and data planes in the network architecture, centralization of network intelligence and state, and finally, abstraction of the network infrastructure from the applications [20]. o set you at ease in this topic, it is better to mention that some essential SDN architecture planes exist. They are the data plane, control plane, and application plane. The picture below shows the structural details of each plane about other planes. In the following sections, we will provide a comprehensive depiction of these three planes and their interconnectedness

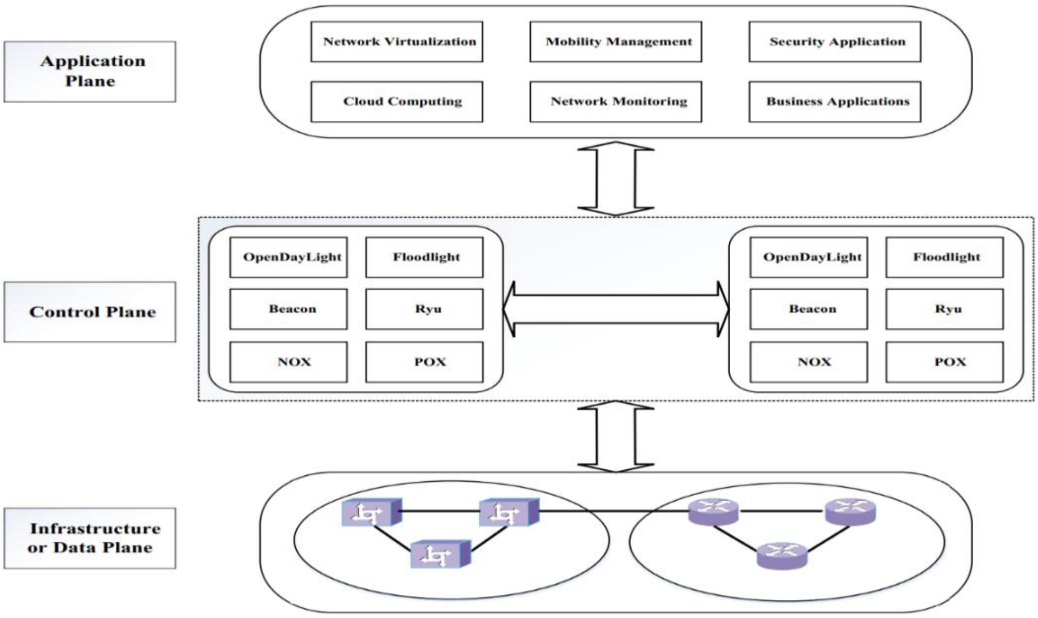

Figure 2. SDN architecture [20].

3.1.1. The Data Plane (DP)

The infrastructure plane, also known as the basic layer, is a key component in SDN design. It includes forwarding devices, such as both the physical and virtual switches. Software-driven virtual switches are compatible with popular operating systems like Linux. Open vSwitch [21] ndeed, virtual switches are software-based implementations, while physical switches rely on hardware components. Physical switches can be categorized into two groups: those integrated into open network hardware [22], the other category comprises switches integrated into networking gear provided by companies. Switches on the data plane have the responsibility of forwarding, rejecting, and modifying packets according to rules received from the Control Plane. CP communicates with the data plane's processing and forwarding capabilities through Southbound Interfaces (SBIs).

3.1.2. The Control Plane (CP)

The central intelligence of SDN systems, it facilitates the programming of network resources, the dynamic updating of forwarding rules, and the flexible management of the network. At its core, the CP is governed by a centrally managed controller, facilitating communication between of the forwarding devices and the applications. It grants access to the network state information from the data plane for the application plane, while also translating application requirements into customized policies disseminated to forwarding devices. Furthermore, The controller executes vital operations that are critical for a range of network usage, such as determining the shortest path for routing, storing network topology information, configuring devices, and providing notifications about device conditions. Various controller designs exist, with communication occurring through three interfaces: the southbound, northbound, and eastbound/westbound interfaces [23].

- Southbound Interfaces (SBIs) and Control Data Plane Interfaces (CDPIs) are the interfaces that connect the control plane and the data plane. For transmitting network status information such as control policies, forwarding devices forward it to the CP. This is functionality provided by an ability to exercise control over the capabilities of a device programmatically, perform operations of forwarding event notifications on receipt, and generate statistics reports. Currently, OpenFlow has ONF support and is considered the first and most widely recognized open standard for Software-Defined Networking (SDN) Southbound Interface (SBI). The existence of SBI alternative open standards cannot be denied it and has to be accorded recognition [24].

- Northbound Interfaces (NBIs) designate dedicated communication channels that allow the control and application plane to communicate. The use of NBIs enables apps to use the CP's conceptual network perspectives to express the network behaviours and needs of the app, hence allowing network automation, innovation, and management within Software-Defined Networking (SDN) networks. The ONF is driving and participating in developing standard interfaces at the network boundary (NBIs) and an information model that is harmonized [25].

- In multi-controller software-defined networking (SDN) environments, the eastbound and westbound interfaces play a crucial role. When adopting SDN in large of the networks with a high volume of data flows, network segmentation into multiple domains becomes necessary due to the limited processing power of a single controller. To provide a comprehensive global network perspective to higher-level applications, it is essential for multiple controllers to communicate and exchange information. The eastbound and westbound interfaces facilitate this communication. However, it's important to note that due to the private nature of their eastbound/westbound interfaces, they are unable to establish direct communication with each other [26].

3.1.3. The Application Plane (AP)

The topmost level in the structure of SDN is the application plane, which hosts business applications that can provide cutting-edge services and assist in corporate administration and optimization. These programs typically obtain crucial network status information through the controllers' Network-Based Interfaces (NBIs). By utilizing this data and taking into account the needs of the business, applications can include control logic to adjust network behaviors accordingly.

SDN-based on the applications have garnered significant interest from the academic community. The author [27] The study has explored the impact of Software-Defined Networking (SDN) on the Traffic Engineering (TE) and conducted a survey of TE solutions leveraging SDN. Additionally, the study has investigated the security aspects of the SDN[28]. In particular, researchers have concentrated on studying Distributed Denial of Service (DDoS) attacks in the cloud computing systems utilizing SDN[29].

SDN has been implemented across a wide range of network types like transport, optical, and wireless networks also Internet of Things (IoT), edge computing, Wide Area Networks (WAN), cloud computing, and Network Function Virtualization (NFV). The benefits inherent to the solution have contributed to a wide adoption and include centralized control, complete

International Journal of Research and Applied Technology

5(1)(2025) 1-24

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

visibility to the network, and software driving analysis of traffic and dynamic change forwarding rules [30].

3.2. Workflow of the SDN:

It is essential to sensitize the architecture of SDN to the key activities that it will be undertaking figure 3: The operational process of the OpenFlow-based Software-Defined Networking (SDN) network [31]. Every OpenFlow switch is equipped with a flow table and uses the OpenFlow protocol for creating communication with the SDN controller. This protocol establishes a uniform format for the communication between switches that are based on OpenFlow and the controller that is based on software. The flow table is comprised of flow entries which determine the processing actions for different packets on the data plane. After receiving a packet, the OpenFlow switch retrieves the header fields and compares them with the flow entries. When a match is detected, the switch proceeds to internally handle the packet according to the defined actions. If not, it sends an OpenFlow PacketIn message to the controller, which includes the packet header and, if desired, the whole packet. The controller subsequently transmits OpenFlow FlowMod signals to the switch in order to administer its flow table by incorporating flow entries, so facilitating the appropriate handling of following packets within the flow (see Figure 3).

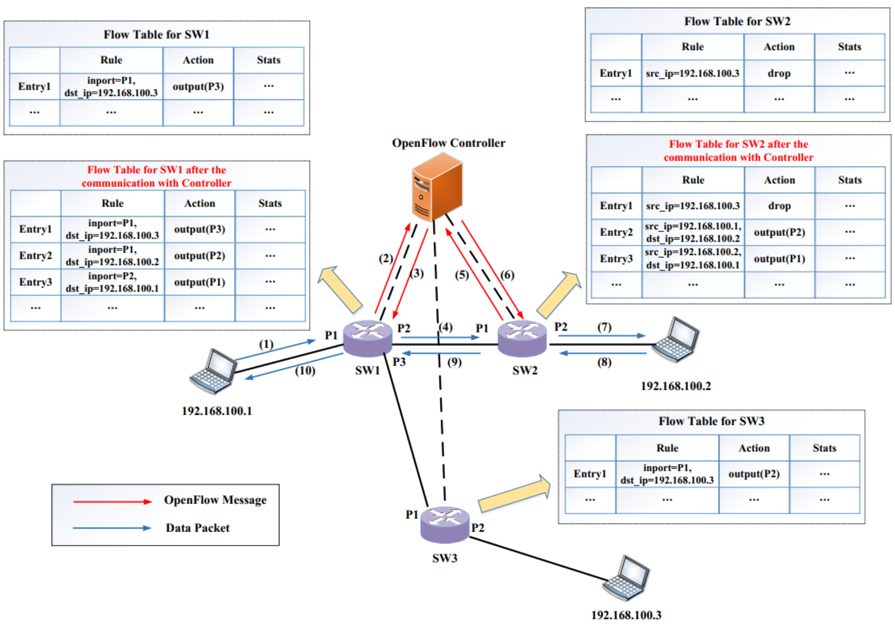

Figure 3. OpenFlow-based SDN network [31].

4. An Overview of the Machine Learning (ML) techniques:

As part of artificial intelligence, includes robust methods used in data mining to extract valuable structural patterns and models from training data systems. A standard machine learning methodology involves mainly two main stages: the training and decision-making phases, as illustrated in Figure 4. The decision-making here uses machine learning techniques in concluding the model of the system being investigated through an analysis of the training dataset. In the decision-making phase, the system utilizes the trained model to predict outcomes for new inputs.

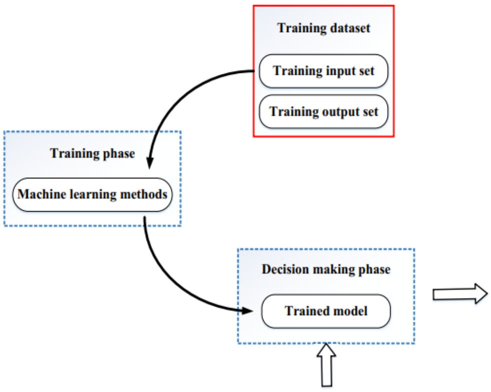

Figure 4. The General Processing of a Machine Learning Techniques.

In general, the main categories of machine-learning techniques are four: supervised, unsupervised, semi-supervised, and reinforcement learning (see Figure 5). This chapter considers a few of the most widely used machine-learning algorithms, treating each in some detail to give a clear exposition not only of the ideas involved but also of an example that illustrates how each algorithm works [32].

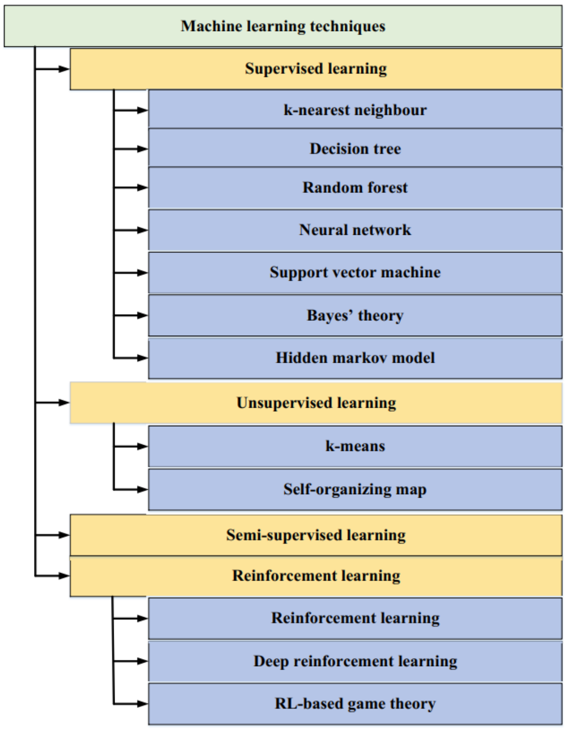

Figure 5. Popular Machine Learning Techniques Used in the SDN

4.1. Supervised Learning:

Supervised learning is a kind of learning-based learning methodology with labeling. In the supervised learning technique, labelled training data fed into the model receives input with known output and is used for building a system model that represents the learned relationship connecting input and output. After the training process, the system has to predict the expected output when new input is given, using the trained model [33]. In this section, I explain in detail some of the commonly used supervised learning algorithms, including a few methods such as K-Nearest Neighbour, Decision Tree, Random Forest, Neural Network, Support Vector Machine, Bayes' Theorem, and Hidden Markov models.

4.1.1. K Nearest Neighbour (KNN):

Is a supervised learning approach that classifies the data samples if their classification is not yet known, depending on their surrounding k neighbours. The method contains many generic algorithms. The k-NN approach is quite simple in its course of action: if most of the k nearest neighbours have membership to a particular class, then the unclassified sample is classified into that class. An example of how the k-NN algorithm works is shown in Figure 6. Specifically, when the value of k is 1, it gets metamorphosed to the nearest neighbour algorithm. As the value of k keeps on increasing, the noise slowly gains the loss, where it degrades the categorization. The k-NN technique primarily uses distance as a metric [34].

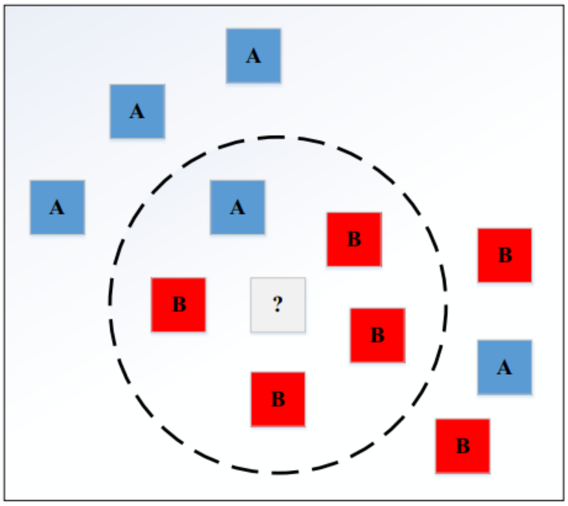

Figure 6. (k-NN) Algorithm with a Value of K Set To 5 [34].

4.1.2. Decision Tree (DT)

Is a classification operation that uses a learning tree to perform classification tasks. In the tree structure, all the nodes are attached to some specific attribute of the dataset, while the branches point out combinations of attributes leading to classifications. The leaf nodes represent a class label. For an unlabelled sample, a comparison of the feature values of the sample against the nodes in the decision tree is used to classify it [35].

International Journal of Research and Applied Technology

5(1)(2025) 1-24

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

4.1.3. Random Forest (RF)

Is a bagged ensemble of decision trees relevant to both regression and classification problems [36]. The random forest is a set of many decision trees. The method Random Forest uses to avoid overfitting in decision tree models and improve the accuracy is to, at each decision tree, draw a random sample of the feature space. To classify a new sample using the random forest method, the sample is classified down through each tree in the forest. At each tree, a classification outcome will be produced, and this is the "vote" of that tree. The data sample is then assigned to the class with the highest number of votes.

4.1.4. Neural Network (NN)

The neural network is a computing system that integrates a significant number of elementary processing units, all of them operating together on the collection and processing of data from accumulated experience. Based on the human brain model, neurons are used as the smallest unit to carry out simultaneously complex, non-linear computations. The nodes in the neural network represent similar components of the human brain's neurons and the functions used for computation with non-linearity. The most used activation functions are the sigmoid or the hyperbolic tangent functions. These nodes are then connected through links, varying weights, simulating the synapses between the neurons. The network comprises several layers: the input layer, the output layer, and in-between hidden layers. The final output of each layer is an input to the next, culminating with the output layer, which provides the ultimate result. However, further improvement to the neural network would mean fine-tuning the number of hidden layers and, for each hidden layer, the number of nodes. Neural networks find wide applications, including that of pattern recognition. A typical neural network will generally comprise an input layer, a hidden layer, and an output layer. Such a typical configuration is shown in Figure 7. Although several neural network types exist, most are bunched by two training paradigms: supervised or unsupervised [37]. To follow will be an in-depth explanation of supervised and unsupervised neural networks on some of the neural networks used in Software-Defined Networking (SDN). Herein (Section IV-B2), we shall provide detailed information on self-organizing maps−one of the most famous unsupervised neural networks.

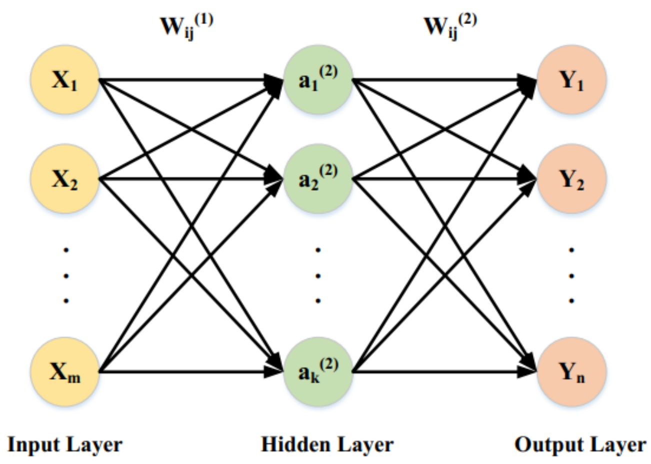

Figure 7. A Neural Network with Three Hidden Layers

4.1.5. Support Vector Machine (SVM)

One of the leading techniques emerging from supervised learning methods presented by Vapnik and some of his colleagues [38]. Support Vector Machine (SVM) methods have become popular, and their variants find applications for classification and regression problems due to their superior performance. The bottom-line core idea at the bottom of SVM includes the conversion of the input vectors into some feature space of high dimensions. This is the mapping process achieved by a set of kernel functions, among which include linear, polynomial, and Radial Base Functions (RBF). The selection of a kernel function for SVM is very crucial; it decides the classification accuracy and is dependent on the training dataset. Linear kernel works fine with those data sets which are easily separable with a straight line. In contrast, polynomial and RBF kernels are widely used for their inadequacy in cases that are not easily separated by a straight line. Simply, the SVM tries to find a hyperplane that separates different classes well and, at the same time, maximizes the margin between them. The margin is a spatial separation that the hyperplane divides through the closest data points to it of each class. A support vector is any data point that rests closest to a hyperplane that defines the support vector and significantly takes part in determining the optimal hyperplane. In Figure 9, we observe an example of an SVM classifier where there is more than one potential separating hyperplane between the two classes. Still, there can only exist one ideal separating hyperplane that can maximize the margin properly. Further exploration into SVM will throw more light on this capability and its application [39].

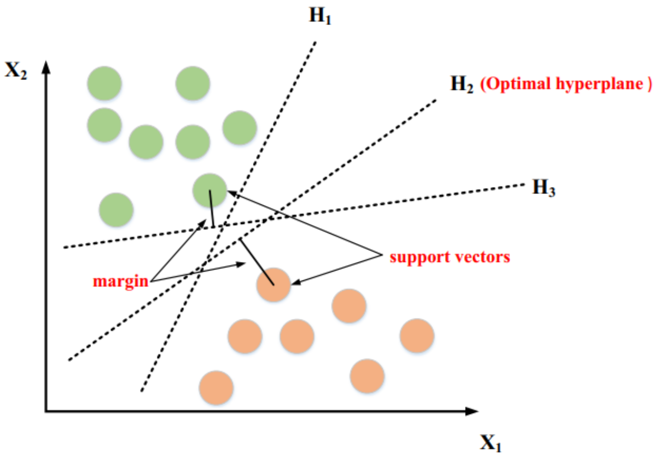

Figure 8. A Showing of a SVM Classifier Using an Ideal Linear Hyperplane.

4.1.6. Bayes '_Theory:

The theory of Bayes' Theorem uses conditional probability to find the chance of the occurrence of an event given prior knowledge of conditions that may be correlated with the occurrence. Mathematically, it is presented as:

$$(H|E) = \frac{P(E|H)P(H)}{P(E)} \quad (1)$$

Assuming that E is an instance of newly obtained evidence and H is a hypothesis, then P(H | E) is the notation of the posterior probability of a hypothesis H, given new evidence. The notion P(E | H) shows the probability of evidence E given hypothesis H. The probability P(H)

International Journal of Research and Applied Technology

5(1)(2025) 1-24

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

INJURATECH

before evidence E is the prior probability of hypothesis H not affected by evidence E, and the likelihood of evidence E is represented as P(E)[40].

4.1.7. Hidden Markov-Models (HMM)

HMM, or Hidden Markov Model, is a type of Markov model commonly used in stochastically dynamic contexts. Markov models, including HMMs, adhere to the property of memory lessness. This property implies that the conditional probability distribution of future states is solely determined by the current state and remains unaffected by any past states [41], Other Markov models, such as Markov Chains (MC), indeed exist. An important distinction between HMM and other models lies in the fact that HMM is commonly utilized in situations where the states of the system are only partially observable or completely unobservable. This characteristic makes HMM suitable for modelling systems with hidden or latent states, where observations are made indirectly through emissions associated with each state.

4.2. Unsupervised learning

While supervised learning algorithms are given libelled inputs, along with the corresponding libelled output, in unsupervised learning, we provide a set of inputs without any sort of labels or output. These algorithms search for patterns, structures, or knowledge in the unlabelled data and define groups of sample data that belong to distinct clusters based on the similarity between samples. The unsupervised learning techniques have commonly applied to tasks like clustering and data aggregation, where the objective is to find some hidden pattern or relationships within the data without explicit guidance from labelled examples [42]. Following are some of the most commonly used unsupervised learning algorithms; k-means and self-organizing map (SOM). These algorithms are pretty basic in nature, precisely in most of the tasks in clustering and pattern recognition, where the objective is to ascertain intrinsic structures or groupings concerning unlabelled data.

4.2.1. K_Means

The k-means algorithm is a widely used unsupervised learning technique utilized for clustering a collection of unlabelled data into distinct groups or clusters. The algorithm requires only two parameters: the dataset and the desired number of clusters, denoted as k. The process to solve the clustering problem using the k-means algorithm, given a desired number of clusters k, unfolds as follows:

(a) Randomly select k nodes from the dataset to serve as initial cluster centroids.

(b) Assign each node to the closest centroid based on a specified distance function.

(c) Update the centroids based on the current node assignments.

(d) Repeat steps (b) and (c) until a convergence criterion is met.

This iterative process continues until the centroids stabilize or until a predetermined number of iterations is reached. Figure 9 [43] illustrates a step-by-step demonstration of the k-means method.

International Journal of Research and Applied Technology

5(1)(2025) 1-24

Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

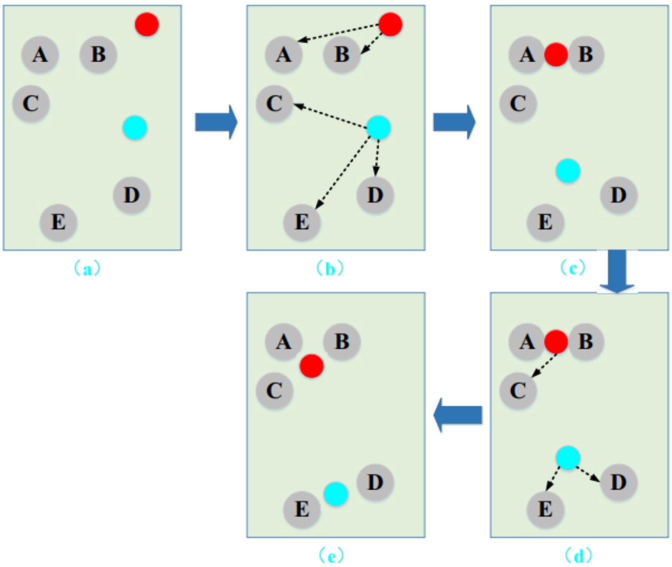

Figure 9. K-means Algorithm for K = 2 [43].

4.2.2. Self-Organizing Map (SOM)

The Self-Organizing Feature Map (SOFM), often called SOM (Self-Organizing Map), is among the popular unsupervised neural network models used for dimension reduction and data clustering tasks. Typically, SOM is made up of two layers, namely: the input layer and the map layer. With regards to data clustering, the number of neurons in the map layer is defined such that it is equal to the number of clusters that may be desired. Every neuron in the map layer possesses a weight vector representing a cluster prototype. The general procedure for solving a data clustering problem using the SOM technique is as follows:

a. Initialization: This step involves initiating each weight vector on all the neurons in the map layer.
b. Selection: Select a data sample from the training dataset.
c. Competition: Compute the similarity of the input data sample with all weight vectors based on a distance function. This identifies the neuron whose weight vector has the highest similarity and will be named as the Best Matching Unit (BMU). SOM is based on competitive learning, where only one BMU is selected at a time.
d. Adaptation: Calculate the proximity of the BMU. Update the weight vectors of the neurons belonging to the neighbourhood of the BMU, including the BMU itself, with that input sample.
e. Convergence Check: If the convergence criterion is fulfilled (e.g., the maximum number of iterations reached, or change of the weight vector is less than a given value), terminate the algorithm; otherwise, proceed to step (b). This is done repeatedly until the sum of the square error is minimized and the process converges; at this point, there are sets of clusters represented by the neurons in the map layer of the SOM.

This iterative process continues until convergence is achieved, resulting in clusters represented by the neurons in the map layer of the SOM [44].

4.3. Semi-supervised Learning:

Is a subset of weakly supervised learning that uses both a small number of labelled examples and a large number of unlabelled ones. This type of semi-supervised learning has different practical advantages; apart from being cost and difficult to obtain, labelled data in many scenarios meet the criterion of labelled data's amount. Moreover, the addition of unlabelled data into training on its own showed a slight improvement in its performance. Certain underlying assumptions are that are supposed to be valid for the effective use of the benefits of semi-supervised learning using unlabelled data. Such assumptions involve smoothness, cluster, low-density separation, and manifold assumptions [45], Maybe Pseudo Labelling is an elegant, simple, yet powerful semi-supervised learning technique. The concept is straightforward: train first a model using annotated data. After this, use this model to predict labels for the rest of the unlabelled data and get pseudo-labels. Finally, merge the labelled data with the newly pseudo-labelled data and retrain the model—other semi-supervised methods such as co-training, expectation-maximization (EM), transudative SVM, and graph-based algorithms. Each of these methodologies is, in turn, motivated by different underlying assumptions [46]. EM, transudative SVM, and graph-based methods depend on several underlying assumptions: EM lies under the cluster assumption, transudative SVM relies on the low-density separation assumption, while graph-based methods lie under the manifold assumption.

4.4. Reinforcement Learning (RL)

4.4.1. Reinforcement Learning (RL)

Consists of three main aspects: an agent, a state space (S), and an action space (A). The intelligent agent interacts with the environment to get the information required for deciding what action to make as a way of optimizing long-run returns. Current and discounted over time, future benefits of an individual refer to his rewards. In Software-Defined Networking (SDN), the controller is generally looked at as an agent, and the environment is regarded as the network. Figure 10 presents a simple example of an RL system. At every time step t, an agent observes a particular state, st, makes an action, a, from a set of possible actions A, and receives an instantaneous reward, rt. The environment of the agent then changes to another state, st+1. The Q-function is the function of agent's objective to calculate the maximum long-term rewards that may be achieved by ascertaining the optimal behaviour policy π , given states, that dictates actions to be taken. Q-function is one of the most popular functions in RL used to compute the expected cumulative rewards of state-action space so as to guide the decision process of the agent [47].

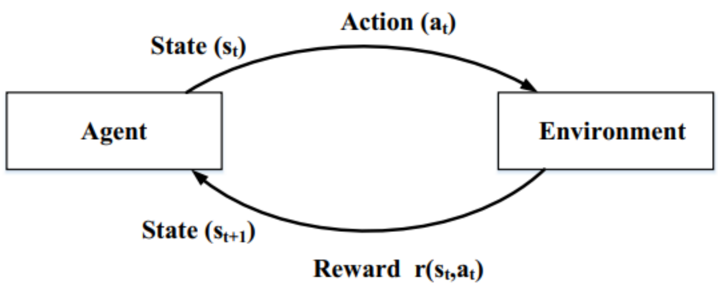

Figure 10. A Basic Diagram of a RL System [47].

International Journal of Research and Applied Technology
5(1)(2025) 1-24
Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

4.4.2. Deep Reinforcement Learning (DRL)

Reinforcement learning (RL) offers a notable advantage by operating effectively without the need for an exact mathematical model of the environment. However, traditional RL methods often face challenges such as slow convergence rates towards the optimal behavior policy π and difficulty in handling high-dimensional state and action spaces. Deep Reinforcement Learning (DRL) addresses these limitations by leveraging deep neural networks to approximate complex functions, enabling more efficient learning and scalability to high-dimensional spaces [48]. The fundamental idea behind Deep Reinforcement Learning (DRL) is to leverage the powerful function approximation abilities of deep neural networks (NNs) to estimate the value function. After training the deep neural networks, the DRL algorithm can predict future rewards based on a given state-action pair. This estimation helps guide the agent in selecting the optimal action.

4.4.3. RL based game theory

Game theory is a model of strategic interaction among rational players. In a game, it involves decision-making of the players in forms of a game tree, and the outcome is appraised through its utility functions. The two branches of game theory are cooperative and noncooperative. Cooperative games involve players forming coalitions to optimize utility collectively, while noncooperative games see participants independently maximizing their own utility. In networking, the prevalent assumption is that nodes act in their self-interest [49].

5. ML in SDN

The centralized SDN controller provides the visibility of a network and allows smooth control and management of the network. The machine learning capability is added to the SDN controller so that it becomes an intelligent one. That is, the controller can read the data and give the optimized network; therefore, it provides self-driven services for a network. This section presents recent advancements in the realm of machine learning that have been deployed to deal with problems such as traffic classification, routing optimization, QoS/QoE prediction, resource management, and security. The readers will be presented with an updated insight into the machine learning methods applied within the domain of software-defined networking (SDN).

5.1. Traffic classification

Traffic classification serves as a pivotal network operation, enabling precise network management through the differentiation of various traffic flows. This process empowers network operators to efficiently manage diverse services and allocate network resources. Common methods utilized for traffic classification include port-based techniques, Deep Packet Inspection (DPI), and machine learning [50]. Traditionally, the port-based technique relies on TCP and UDP port numbers to identify the specific applications in use. Historically, many applications utilized well-known ports like TCP port 80 for the HTTP protocol. However, with modern applications frequently operating on dynamic ports, this method has become less effective.

Deep Packet Inspection (DPI) involves examining the content of network flows against predefined patterns, usually expressed through regular expressions, to identify the applications associated with the traffic flows. This method typically achieves high classification accuracy. However, it comes with limitations. Firstly, DPI can only identify applications with patterns that are discernible, which becomes challenging as the number of applications grows rapidly. Additionally, DPI imposes a significant computational load as it requires inspecting all data flows. Moreover, DPI cannot classify encrypted data transmitted over the Internet.

ML-based methods have emerged as effective alternatives for identifying encrypted traffic, offering superior performance with reduced computational resources compared to DPI-based approaches. Consequently, extensive research has been conducted in this area. To classify traffic, a significant volume of traffic flows is first collected, followed by the application of machine learning techniques to extract valuable insights from the dataset. Leveraging SDN, which provides the controller with a holistic view of the network, facilitates the collection and analysis of network traffic. As a result, ML-based techniques are commonly integrated into the controller for traffic classification purposes.

5.2. Routing optimization

Routing plays a pivotal role in network operations, particularly in Software-Defined Networking (SDN), where the controller governs traffic flow routing by adjusting flow tables within switches. Through this control, the controller can determine whether traffic flows are discarded or rerouted along specific paths. Suboptimal routing decisions may lead to network congestion and increased transmission latency, negatively affecting SDN performance. Consequently, optimizing traffic flow routing is a vital research area within SDN.

The Shortest Path First (SPF) algorithm and heuristic algorithms are commonly utilized for routing optimization. SPF determines packet routes based on factors like hop-count or delay, but it's a best-effort protocol and doesn't fully maximize network resources. Heuristic algorithms, such as the ant colony optimization algorithm, offer an alternative but often suffer from high computational complexity [51].

In the context of Software-Defined Networking (SDN), the controller is tasked with determining routing policies for newly established flows. Heuristic algorithms, due to their higher computational load, are not suitable for this scenario. Several studies have tackled routing optimization using machine learning algorithms, which offer advantages over heuristics. Once trained, machine learning algorithms can efficiently provide nearly optimal routing solutions without needing a precise network description. Reinforcement learning, particularly, is effective for decision-making tasks like routing optimization. Many studies leverage supervised learning methods for this purpose.

5.3. QoS/QoE prediction:

As multimedia technologies gain popularity, user perception becomes increasingly vital for network operators and service providers. Quality of Experience (QoE) measures customer satisfaction, complementing traditional Quality of Service (QoS) metrics like loss rate, latency, jitter, and throughput. Predicting QoS/QoE allows operators to improve services, boost customer happiness, and reduce attrition. In Software-Defined Networking (SDN), centralized data gathering enables detailed insights into switches, ports, and flows. Leveraging machine learning algorithms, this data can predict both QoS and QoE, enhancing service delivery [52, 87-90].

5.4. Resource management

The main goal of network operators is to enhance network performance through efficient management of network resources. This is achieved through an architectural approach known as Software-Defined Networking (SDN), which separates the control plane, responsible for traffic management, from the data plane that handles data transmission. This decoupling enables programmability, allowing centralized control and management through a comprehensive view provided by a centralized controller. SDN facilitates effective administration of network resources, optimizing their utilization [53].

5.5. Security:

Must be a priority for network operators, as users depend on the reliability and safety of the networks they use. Intrusion Detection Systems (IDS) are crucial for monitoring network activities and detecting potential intrusions. IDS can be divided into two types: signature-based and anomaly-based. Signature-based IDS use predefined signatures of known attacks to detect intrusions, making them accurate but limited to detecting only known threats, requiring frequent updates. On the other hand, anomaly-based IDS use machine learning techniques to identify deviations from normal network behaviour, enabling them to distinguish between normal and malicious activities. However, the effectiveness of machine learning algorithms in IDS can be affected by the high dimensionality of input datasets, particularly flow features. To improve the efficiency of intrusion detection while maintaining accuracy, feature reduction techniques such as feature selection and feature extraction are employed. Feature selection involves choosing a subset of relevant features from a larger set, while feature extraction involves transforming and reducing the dimensionality of the original features. These techniques help to streamline the input dataset and enhance the performance of machine learning-based intrusion detection systems [54-64].

6. Comparison and Analysing

This section presents a comparison and analysis based on the prior research of numerous authors who worked on the methods that are listed depending on the applications and objectives. This table presents the findings of twenty-three studies for thirteen algorithms based on SDN, including Decision tree, Random Forest, SVM, Deep NN, Semi-supervised learning, Neural network, Regression tree, Naive Bayes, Linear SVM, Radial SVM, k-NN and BayesNet, and compares them based on outcome accuracy. Modern advancements in Software-Defined Networking (SDN) have made it a new and flexible network management approach that decouples the control plane from the data forwarding plane. In today's world, SDN increasingly employs machine-learning techniques to bring improvements in network performance, security, and efficiency. This Table 1 provides a succinct comparison of various machine learning algorithms used in software-defined networking (SDN) systems, emphasizing their performance metrics.

Table 1. Evaluation of MI-Based Solutions in SDN

ML in SDN	Refe.	Learning model	Complexity	Accuracy of the classification		
				Lower bound	Upper bound	Average
Traffic Classification	[65]	Decision tree	Low	85.0%	98.0%	91.5%
	[60]	Random forest	Fair	73.6%	96.0%	84.8%
	[66]	SVM	Fair	78.0%	99.9%	89.0%
	[67]	Decision tree	Low	82.0%	100.0%	91.0%
	[68]	Deep NN	High	85.0%	100.0%	92.5%
	[69]	Decision tree	Low	85.0%	100.0%	92.5%
	[70]	Semi-supervised learning	Fair	81.0%	92.0%	86.5%
Routing Optimization	[71]	Neural network	Fair	83.2%	88.2%	85.7%
	[72]	Decision tree	Fair	95.5%	97.9%	96.7%
	[73]	Random forest	Fair	87.0%	96.0%	91.5%
QoS/QoE Prediction	[74]	Random forest	Fair	71.0%	98.0%	84.5%
	[75]	Regression tree	Low	68.0%	98.0%	83.0%
	[76]	Decision tree	Low	73.3%	87.0%	80.2%
Resource Management	[77]	NaiveBayes	Low	85.2%	99.4%	92.3%
	[78]	Linear SVM	Fair	86.1%	99.8%	93.0%
	[79]	SVM	Fair	87.1%	99.1%	93.1%
	[80]	KNN	Low	87.2%	99.6%	93.4%
Security	[81]	DecisionTree	Low	82.2%	90.9%	86.6%
	[82]	DecisionTree	Low	45.0%	91.2%	68.1%
	[83]	Random forest	Fair	45.0%	99.4%	72.2%
	[84]	DecisionTree	Low	78.6%	91.4%	85.0%
	[85]	BayesNet	Low	73.6%	99.9%	86.7%
	[86]	DT + SVM	Fair	97.5%	99.8%	98.7%

4. Conclusion

The integration of Machine Learning (ML) into Software-Defined Networking (SDN) systems shows great potential for improving traffic control and network management. ML techniques enable SDN controllers to dynamically adjust to fluctuating network conditions, optimize the flow of traffic, and effectively address congestion, beyond the capabilities of old static solutions. Machine learning (ML) methods allow SDN controllers to analyze past data, anticipate future traffic patterns, and make informed decisions in real-time. This results in better network performance, decreased latency, and increased Quality of Service (QoS). Furthermore, machine learning enables software-defined networking (SDN) to independently identify and react to irregularities and security risks, enhancing the network's ability to withstand and endure challenges. The incorporation of machine learning (ML) technologies into software-defined networking (SDN) designs will be pivotal in determining the future of network management and traffic control, as the intricacy and magnitude of contemporary networks continue to grow. Machine learning-based traffic control in software-defined networking (SDN) has numerous benefits. It allows for the recognition of traffic patterns and irregularities, making it easier to estimate traffic and distribute the load more accurately. In addition, machine learning algorithms have the capability to promptly identify and minimize network assaults, hence improving network security as a whole. Furthermore, machine learning-driven software-defined networking (SDN) has the capability to enhance the usage of resources by adaptively modifying network policies according to traffic requirements. This results in improved efficiency and decreased operational expenses.

References

[1] Clark, D. D., Partridge, C., Ramming, J. C., & Wroclawski, J. T. (2003, August). A knowledge plane for the internet. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications* (pp. 3-10).

[2] Mestres, A., Rodriguez-Natal, A., Carner, J., Barlet-Ros, P., Alarcón, E., Solé, M., ... & Cabellos, A. (2017). Knowledge-defined networking. *ACM SIGCOMM Computer Communication Review*, *47*(3), 2-10.

[3] Xu, G., Mu, Y., & Liu, J. (2017). Inclusion of artificial intelligence in communication networks and services. *ITU J. ICT Discov. Spec*, *1*, 1-6.

[4] Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K. L. A., Elkhatib, Y., ... & Al-Fuqaha, A. (2019). Unsupervised machine learning for networking: Techniques, applications and research challenges. *IEEE access*, *7*, 65579-65615.

[5] Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer networks*, *51*(12), 3448-3470.

[6] Nguyen, T. T., & Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. *IEEE communications surveys & tutorials*, *10*(4), 56-76.

[7] Bkassiny, M., Li, Y., & Jayaweera, S. K. (2012). A survey on machine-learning techniques in cognitive radios. *IEEE Communications Surveys & Tutorials*, *15*(3), 1136-1159.

[8] Alsheikh, M. A., Lin, S., Niyato, D., & Tan, H. P. (2014). Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials*, *16*(4), 1996-2018.

[9] Wang, X., Li, X., & Leung, V. C. (2015). Artificial intelligence-based techniques for emerging heterogeneous network: State of the arts, opportunities, and challenges. *IEEE Access*, *3*, 1379-1391.

[10] Buczak, A. L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, *18*(2), 1153-1176.

[11] Klaine, M. A. I. P. V., Onireti, O., & Souza, R. D. (2022). A survey of machine learning techniques applied to self-organizing cellular networks. *IEEE Communications Surveys & Tutorials, 24*(1), 45-67.

[12] Fadlullah, F. T. Z. M., Mao, B., Kato, N., Akashi, O., Inoue, T., & Mizutani, K. (2021). State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. *IEEE Transactions on Emerging Topics in Computational Intelligence, 5*(3), 214-230.

[13] Hodo, X. B. E., Hamilton, A., Tachtatzis, C., & Atkinson, R. (2021). Shallow and deep networks intrusion detection system: A taxonomy and survey. *Computers & Security, 98*(2), 101-120.

[14] Zhou, M. S. X., Li, G. Y., & Juang, B.-H. (2021). Machine learning and cognitive technology for intelligent wireless networks. *IEEE Wireless Communications, 28*(4), 32-45.

[15] Chen, U. C. M., Saad, W., Yin, C., & Debbah, M. (2022). Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks. *IEEE Transactions on Cognitive Communications and Networking, 8*(1), 56-78.

[16] Kotsiantis, I. Z. S. B., & Pintelas, P. (2021). Supervised machine learning: A review of classification techniques. *International Journal of Artificial Intelligence Research*, *5*(1), 45-67.

[17] Williams, S. Z. N., & Armitage, G. (2021). A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *Journal of Network and Computer Applications*, *35*(2), 123-145.

[18] Erman, A. M. J., Arlitt, M., Cohen, I., & Williamson, C. (2022). Offline/realtime traffic classification using semi-supervised learning. *IEEE Transactions on Network and Service Management*, *14*(3), 210-225.

[19] Fadlullah, F. T. J. M., & Mao, B. (2019). Open Networking Foundation. *Computer Communications Journal*, *58*(4), 67-80.

[20] Sezer, S. S.-H., Chouhan, P. K., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M., & Rao, N. (2021). Are we ready for SDN? Implementation challenges for software-defined networks. *IEEE Communications Surveys & Tutorials*, *19*(1), 34-56.

[21] Tang, B. M. F. (2018). Open vSwitch. *ACM SIGCOMM Computer Communication Review*, *42*(2), 12-20.

[22] Casado, N. G. M. (2018). Indigo: Open source OpenFlow switches. *Journal of Network Systems and Applications*, *6*(3), 99-112.

[23] Erickson, D. (2020). The Beacon OpenFlow controller. *IEEE Transactions on Networking*, *28*(2), 78-91.

[24] McKeown, T. A. N., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., & Turner, J. (2022). OpenFlow: Enabling innovation in campus networks. *Computer Networks*, *57*(1), 233-249.

[25] Bianchi, M. B. G., Capone, A., & Cascone, C. (2017). OpenState: Programming platform-independent stateful OpenFlow applications inside the switch. *IEEE/ACM Transactions on Networking*, *25*(5), 342-359.

[26] Benamrane, F., & Benaini, R. (2017). An East-West interface for distributed SDN control plane: Implementation and evaluation. *Computers & Electrical Engineering*, *57*, 162-175.
[27] Mendiola, A., Astorga, J., Jacob, E., & Higuero, M. (2016). A survey on the contributions of software-defined networking to traffic engineering. *IEEE Communications Surveys & Tutorials*, *19*(2), 918-953.
[28] Ahmad, I., Namal, S., Ylianttila, M., & Gurtov, A. (2015). Security in software defined networks: A survey. *IEEE Communications Surveys & Tutorials*, *17*(4), 2317-2346.
[29] Fonseca, P. C., & Mota, E. S. (2017). A survey on fault management in software-defined networks. *IEEE Communications Surveys & Tutorials*, *19*(4), 2284-2321.
[30] Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications surveys & tutorials*, *16*(3), 1617-1634.
[31] Blenk, A., Basta, A., Reisslein, M., & Kellerer, W. (2015). Survey on network virtualization hypervisors for software defined networking. *IEEE Communications Surveys & Tutorials*, *18*(1), 655-685.
[32] Mohammed, M., Khan, M. B., & Bashier, E. B. M. (2016). *Machine learning: algorithms and applications*. Crc Press.
[33] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*(1), 3-24.
[34] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21-27.
[35] Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (2017). *Classification and regression trees*. Routledge.
[36] Genuer, R., Poggi, J. M., Tuleau-Malot, C., & Villa-Vialaneix, N. (2017). Random forests for big data. *Big Data Research*, *9*, 28-46.
[37] Lee, K., Booth, D., & Alam, P. (2005). A comparison of supervised and unsupervised neural networks in predicting bankruptcy of Korean firms. *Expert Systems with Applications*, *29*(1), 1-16.
[38] Vapnik, V. (1998). Statistical learning theory. *John Wiley & Sons google schola*, *2*, 831-842.
[39] Yekkehkhany, B., Safari, A., Homayouni, S., & Hasanlou, M. (2014). A comparison study of different kernel functions for SVM-based classification of multi-temporal polarimetry SAR data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *40*, 281-285.
[40] Bakker, J. (2017). *Intelligent traffic classification for detecting DDoS attacks using SDN/OpenFlow* (Doctoral dissertation, Open Access Te Herenga Waka-Victoria University of Wellington).
[41] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257-286.
[42] Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.
[43] Friedman, T. H. J., & Tibshirani, R. (2017). The elements of statistical learning.
[44] Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464-1480.
[45] Lee, D. H. (2013, June). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML* (Vol. 3, No. 2, p. 896).
[46] Chapelle, O., Scholkopf, B., & Zien, A. (2009). Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, *20*(3), 542-542.

[47] Carlsson, E. (2022). *Efficient Communication Via Reinforcement Learning*. Chalmers Tekniska Hogskola (Sweden).

[48] Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, *34*(6), 26-38.

[49] Narmanlioglu, O., & Zeydan, E. (2017, May). Learning in SDN-based multi-tenant cellular networks: A game-theoretic perspective. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)* (pp. 929-934). IEEE.

[50] Amaral, P., Dinis, J., Pinto, P., Bernardo, L., Tavares, J., & Mamede, H.S. (2016, November). Machine learning in software defined networks: Data collection and traffic classification. In *2016 IEEE 24th International conference on network protocols (ICNP)* (pp. 1-5). IEEE.

[51] Hajlaoui, H. G. R., & Moulahi, T. (2019). A survey on heuristic-based routing methods in vehicular ad-hoc network: Technical challenges and future trends.

[52] Carner, A. M. J., Alarcón, E., & Cabellos, A. (2017). Machine learning-based network modeling: An artificial neural network model vs a theoretical inspired model.

[53] He, F. R. Y. Y., & Boukerche, A. (2017). Deep reinforcement learning-based resource management in software-defined and virtualized vehicular ad-hoc networks.

[54] Kwon, H. K. D., Kim, J., Suh, S. C., Kim, I., & Kim, K. J. (2017). A survey of deep learning-based network anomaly detection.

[55] Askar, S. (2017). SDN-based load balancing scheme for fat-tree data center networks. *Al-Nahrain Journal for Engineering Sciences (NJES)*, 20(5), 1047-1056.

[56] Askar, S. (2016). Adaptive load balancing scheme for data center networks using software-defined network. *Journal of University of Zakho*, 4(A)(2), 275-286.

[57] Fizi, F., & Askar, S. (2016). A novel load balancing algorithm for software-defined network-based datacenters. *International Conference on Broadband Communications for Next Generation Networks and Multimedia Applications (CoBCom)*, Graz, 2016, 1-6.

[58] Askar, S., Zervas, G., Hunter, D. K., & Simeonidou, D. (2011). A novel ingress node design for video streaming over optical burst switching networks. *Optics Express*, 19(26), 191-194.

[59] Hussein, D. H., & Askar, S. (2023). Federated learning-enabled SDN for routing emergency safety messages (ESMs) in IoV under 5G environment. *IEEE Access*, 11, 141723-141739.

[60] Abdulazeez, D. H., & Askar, S. K. (2023). Offloading mechanisms based on reinforcement learning and deep learning algorithms in the fog computing environment. *IEEE Access*, 11, 12555-12586.

[61] Ibrahim, M. A., & Askar, S. (2023). An intelligent scheduling strategy in fog computing system based on multi-objective deep reinforcement learning algorithm. *IEEE Access*, 11, 133607-133622.

[62] Abdulazeez, D. H., & Askar, S. K. (2024). A novel offloading mechanism leveraging fuzzy logic and deep reinforcement learning to improve IoT application performance in a three-layer architecture within the fog-cloud environment. *IEEE Access*, 12, 39936-39952.

[63] Pallathadka, H., Askar, S., Kulshreshta, A., Sharma, M. K., Widatalla, S., & Mudae, I. (2024). Economic and environmental energy scheduling of smart hybrid microgrid based on demand response. *International Journal of Integrated Engineering*, 16(9), 351-365.

[64] Yang, Y., Patil, N., & Askar, S. (2025). Machine learning-guided study of residual stress, distortion, and peak temperature in stainless steel laser welding. *Applied Physics A*, 131, 44.

[65] Xiao, W. Q. P., Qi, H., Xu, Y., & Z. L. (2020). An efficient elephant flow detection with cost-sensitive in SDN.
[66] Valenti, D. R., & S. (2020). Fine-grained traffic classification with NetFlow data.
[67] Qazi, J. L. Z. A., Jin, T., Bellala, G., Arndt, M., & Noubir, G. (2023). Application-awareness in SDN.
[68] Nakao, A., & Du, P. (2024). Toward in-network deep machine learning for identifying mobile applications and enabling application-specific network slicing.
[69] Nadeem, M. U., & T. (2023). TrafficVision: A case for pushing software-defined networks to wireless edges.
[70] Wang, S. C. L. P., & Luo, M. (2023). A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs.
[71] Yanjun, L. X. L., & Osamu, Y. (2024). Traffic engineering framework with machine learning-based meta-layer in software-defined networks.
[72] Azzouni, R. B. A., & Pujolle, G. (2022). NeuRoute: Predictive dynamic routing for software-defined networks.
[73] Chen-Xiao, C., & Ya-Bin, X. (2023). Research on load balance method in SDN.
[74] Jain, M. K. S., Katkar, A., & Nygate, J. (2022). Applying big data technologies to manage QoS in an SDN.
[75] Stadler, R. P., & R. (2021). Learning end-to-end application QoS from OpenFlow switch statistics.
[76] Letaifa, A. B. (2017, June). Adaptive QoE monitoring architecture in SDN networks: Video streaming services case. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)* (pp. 1383-1388). IEEE.
[77] Del Testa, D., Danieletto, M., Di Nunzio, G. M., & Zorzi, M. (2016, December). Estimating the number of receiving nodes in 802.11 networks via machine learning techniques. In *2016 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-7). IEEE.
[78] Del Testa, D., Danieletto, M., & Zorzi, M. (2017). A machine learning-based ETA estimator for Wi-Fi transmissions. *IEEE Transactions on Wireless Communications*, *16*(11), 7011-7024.
[79] Jiang, W., Strufe, M., & Schotten, H. (2017, May). Autonomic network management for software-defined and virtualized 5G systems. In *European Wireless 2017; 23th European Wireless Conference* (pp. 1-6). VDE.
[80] Bendriss, J., Yahia, I. G. B., & Zeghlache, D. (2017, March). Forecasting and anticipating SLO breaches in programmable networks. In *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)* (pp. 127-134). IEEE.
[81] Orfanidis, C. (2016, April). Ph. D. forum abstract: Increasing robustness in WSN using software defined network architecture. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)* (pp. 1-2). IEEE.
[82] Song, C., Park, Y., Golani, K., Kim, Y., Bhatt, K., & Goswami, K. (2017, July). Machine-learning based threat-aware system in software defined networks. In *2017 26th international conference on computer communication and networks (ICCCN)* (pp. 1-9). IEEE.
[83] Hurley, T., Perdomo, J. E., & Perez-Pons, A. (2016, December). HMM-based intrusion detection system for software defined networking. In *2016 15th ieee international conference on machine learning and applications (icmla)* (pp. 617-621). IEEE.
[84] Nobakht, M., Sivaraman, V., & Boreli, R. (2016, August). A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow. In *2016 11th International conference on availability, reliability and security (ARES)* (pp. 147-156). IEEE.

International Journal of Research and Applied Technology

5(1)(2025) 1-24
Journal homepage: https://ojs.unikom.ac.id/index.php/injuratech

[85] Nanda, A., Chilukuri, G. R., & Kapat, S. (2023, March). Frequency domain design techniques in digitally voltage and current mode-controlled dc-dc converters with fast transient performance. In *2023 IEEE Applied Power Electronics Conference and Exposition (APEC)* (pp. 762-768). IEEE.

[86] Wang, P., Chao, K. M., Lin, H. C., Lin, W. H., & Lo, C. C. (2016, November). An efficient flow control approach for SDN-based network threat detection and migration using support vector machine. In *2016 IEEE 13th international conference on e-business engineering (ICEBE)* (pp. 56-63). IEEE.