Open Access ISSN: 3007-1437 (Online), ISSN: 3007-1429 (Print)

Alkadhim Journal for Computer Science (KJCS)

Journal Homepage: https://alkadhum-col.edu.iq/JKCEAS

A Comprehensive Review of SDN Intrusion Detection Using Ensemble Machine Learning

[1] Batool Khairallah Abd*, [2] Saif Ali Abd Alridha Alsaidi,
[1] College of Computer Science and Information Technology, Wasit university, wasit
[2] College of Computer Science and Information Technology, Wasit university, wasit
[1]stdm.babed@uowasit.edu.iq ,[2]salsaidi@uowasit.edu.iq

Article information

Article history:
Received: April, 28, 2026
Accepted: July, 1, 2026
Available online: September, 25, 2026

Keywords:
Software Defined Networking (SDN),
Intrusion Detection System (IDS),
Machine Learning ,
Ensemble learning,
Network Security.

Corresponding Author:
Batool Khairallha Abd
stdm.babed@uowasit.edu.iq

DOI:
https://doi.org10.61710/fpfehs22

Abstract

Software Defined Networking (SDN) has developed as a potential networking paradigm that improves network programmability, flexibility, and centralized management. The centralized architecture of SDN presents considerable security challenges, rendering Intrusion Detection Systems (IDSs) crucial for detecting and addressing malicious actions. Conventional IDS methodologies frequently experience elevated false alarm rates and exhibit restricted efficacy against advanced and emerging cyber threats. To tackle these issues, machine learning approaches, especially ensemble learning methods, have garnered significant interest for their capacity to enhance detection performance and generalization ability. This article offers a thorough examination of methodologies for intrusion detection using machine learning in software-defined networking contexts, emphasizing ensemble learning strategies. The study investigates prevalent datasets, assesses the advantage and drawbacks of current methodologies, and analyzes contemporary research trends in SDN intrusion detection. The results demonstrate that ensemble learning techniques typically outperform individual machine learning classifiers for detection accuracy, dependability, and resilience. Nonetheless, other obstacles persist, including dataset imbalance, scalability concerns, computational complexity, and constraints in real-time implementation. This analysis identifies existing research gaps and outlines future research paths for the development of more effective, scalable, and reliable IDS systems to secure SDN settings.

1. Introduction

Modern network infrastructure have become increasingly complex and dynamic, particularly with the widespread adoption of Software-Defined Networking (SDN) [1]. While SDN provides flexibility , scalability , and centralized control, it also introduces new security challenges due to its programmable and centralized

architecture [2] .In SDN environments, the centralized controller represents a critical component that may become a single point of failure if compromised [3]. As a result, Intrusion Detection Systems (IDSs) have become essential for monitoring network traffic and identifying malicious activities [4]. However traditional IDS techniques often struggle to detect modern and evolving attacks , especially in dynamic SDN environments [5].

Another critical factor affecting the performances of IDSs in SDN environments is the availability and quality of datasets used for training and evaluation. Several datasets have been widely used in the literature, such as NSL-KDD, CUP 99, CICDDOS2019,CICDDOS 2018, and UNSW-NB15.Despite their popularity , many of these datasets suffer from significant limitations [6] , including outdated traffic patterns , the lack of SDN specific feature, and poor representation of real -world network behavior.

These limitations directly impact the effectiveness and generalization capability of machine learning -based IDS models [7].

As a result, models that achieve high accuracy on these datasets may not perform well in real-world SDN environments. This paper presents a comprehensive review of IDS in SDN environments using machine learning and ensemble learning approaches. It focuses on analyzing commonly used datasets, comparing existing methods, and identifying key challenges and future research directions in this field. Despite the considerable progress achieved in IDS research for SDN environment, several critical issues remain unresolved. These challenges highlighted the need for a deeper investigation into existing limitation and motivation the exploration of more advanced and adaptive solutions. Accordingly, the main motivation of this study can be summarized as follows:

- Despite significant advancements in intrusion detection system (IDS) for Software -Defined Network (SDN), several critical challenges still limit their effectiveness. Traditional IDS approaches often suffer from high false positive rates and limited capability in detecting sophisticated and zero-day attacks.

- In addition, the centralized architecture of SDN introduces new security concerns, as the controller represents a potential single point of failure and an attractive target for attackers [8]. These challenges are further compounded by the reliance on outdated datasets such as NSL-KDD and KPP Cup99, which do not accurately reflect the dynamic and programmable nature of modern SDN environment.

- Recent studies in cyber-physical systems have highlighted that traditional IDSs often struggle with high false positive rates and limited adaptability to evolving cyber threats. To address these limitation , machine learning -based approaches have been widely adopted, offering improved detection performance and the ability to analyze complex patterns in network data[9].

- Therefore, there is a need to explore more advanced and adaptive approaches for IDS development. In this context, machine learning and ensemble learning techniques have emerged as promising solutions to enhance detection accuracy, improve generalization, and reduced false alarm rates in SD- based IDS.

This reviews primary contributions can be encapsulated as follows:

- Delivering an extensive analysis of intrusion detection systems (IDSs) within SDN contexts, focusing specifically n machine learning and ensemble learning methodologies.
- Assessing the strengths, limits, and performance attributes of current ensemble learning methodologies documented in the literature.
- Emphasizing contemporary research challenges and deficiencies, encompassing dataset restrictions, class imbalance problems, scalability issues, computational complexity, and constraints related to real-time deployment.
- Examining recent research trends and prospective approaches that may enhance the accuracy, robustness, and efficiency of intrusion detection system solution for protection of Software -Defined Networking environment.

The rest of this study will be structured as follows: security issues, IDSs, and popular datasets. The third section of this work describes the literature review approach, which consists of stating the research questions, selecting the search strategies, and determining the data sources for study selection purposes. The fourth section analyzes the usage of machine learning algorithms for detecting intrusions with the help of SDNs and different types of classifiers and learners. In turn, the fifth section discusses feature selection methods used to select features in intrusion detection tasks. The sixth section provides an explanation of the concept of ensemble learning, types, benefits, drawbacks, and performance measures of ensembles. Finally, the seventh section presents a thorough discussion of the reviewed literature, the existing research problems, and future work perspectives.

2. Theoretical Background and Related Work

2.1 Software-Defined Networking (SDN)

SDN is defined as a networking paradigm in which the control plane is separated from the data plane ,enabling centralized network management and enhanced programmability [10].In this architecture , network decision making is handled within the control plane , while packet forwarding is performed by the data plane [11]. The SDN controller functions as a centralized entity that oversees network operations and coordinates with forwarding devices t using standardized communication protocols such as Open-Flow [12]. This centralized approach allows for a global view of the network, thus improving traffic management , flexibility , and scalability[13]. However, despite these advantages, SDN faces significant security issues. For instance, SDN's centralized approach makes it vulnerable to single points of failure, and its programmability makes it easy for hackers to launch attacks like DDoS and flow manipulation [14].

2.1.1 SDN Architecture

The SDN architecture is defined as an architecture that enables centralized network management and programmability through the division of network functionality into different layers, as presented in **Figure 1** [3]. SDN architecture comprises four key components: the control plane, data plane, SDN controller, and application layer [14].

- The control plane is responsible for making decisions for the network and maintains a global view of the network.
- The data plane is responsible for forwarding data packets in the network.
- The SDN controller acts as an interface between the application and network devices and facilitates communication between these devices through northbound and southbound interfaces.
- The application defines network policies like routing, security, and intrusion detection.

This layered approach enables SDN to be flexible and dynamic in terms of handling changes in network behavior [13][15].

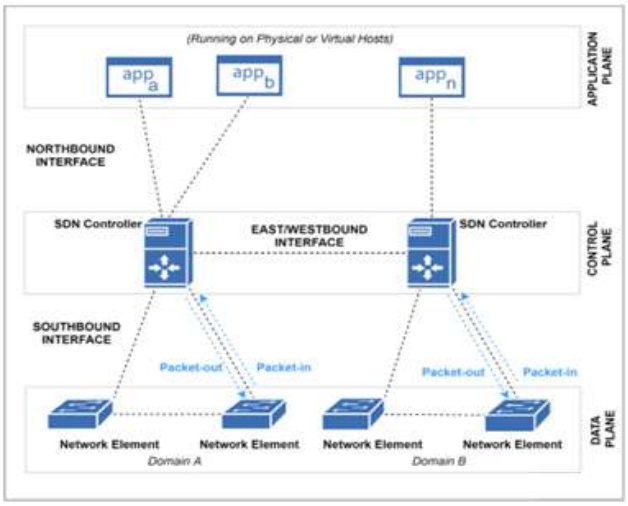

Figure (1): The Architecture of SDN [3].

As illustrated in **Figure 1** the SDN architecture is organized into three main layers: the application plane, control plane, and data plane. The application plane contains network applications that defined policies related to routing, traffic engineering, security, and intrusion detection. These applications communicate with the SDN controller through the northbound interface. The control plane is represented by one or more SDN controller that maintain a global view of the network and make forwarding decision. In multi-controller environments, controllers communicate through southbound/northbound interfaces to exchange information and coordinate network management tasks. The data plane consists of network element such as switches and forwarding devices responsible for handling packet forwarding operations. Communication between the controller and forwarding device is achieved through the southbound interface, commonly implemented using protocols such as Open-Flow. When a switch receives an unknown packet, it may generate a Packet -In message to the controller, which subsequently responds with a Packet-Out intrusion containing the appropriate forwarding rule.

This separation of responsibilities improves network programmability, scalability, and centralized management. However, the reliance on centralized controllers also introduces security challenges, making intrusion detection systems an essential component for protecting SDN environment.

2.1.2 Security Challenges in SDN

The vulnerability that appears within the context of SDN is related to its architecture, primarily, to such concepts as the separation of control plane and data plane, and also the centralized control approach. On the one hand, the described features contribute to increased network flexibility, but on the other hand, they create numerous security gaps that become evident during various attacks [9]. Firstly, as mentioned above, the key vulnerability in SDN refers to the centralized nature of the network control system, represented by a controller [16]. The described element acts as a single point of failure that should be considered by network managers, as compromising the controller leads to total takeover of the network by the attacker [17]. There is evidence indicating that hackers aim to target the controller through such attacks as saturation attacks, packet-in flooding, and topology poisoning, causing network malfunction or inefficiency [18].

Furthermore, another source of vulnerability can be found in communication between the two major network layers. Namely, the SDN control plane communicates with the data plane using the protocol known as Open-Flow [19]. In turn, such communication is vulnerable to man in the middle attacks, DoS/DDoS attacks, and even spoofing attacks, as proven experimentally [2]. Specifically, research indicates that an attacker can launch an efficient DDoS attack against an SDN system, flooding the controller with excessive flow requests and thus resulting in the controller's inability to cope with its responsibilities [20].

Also, the programmability of SDN systems significantly increases the number of possible entry points for attacks ,The reason is that the attacker may manipulate network settings through malicious code or programs by adding unauthorized rules into the system controller[21]. In this respect, the SDN systems become especially vulnerable due to the ease of compromising network routes and intercepting information. Finally, in addition to centralization and programmability issues discussed above, SDN systems face challenges connected with vulnerabilities arising at three different planes [18]. First of all, according to empirical analysis, most attacks occur at the level of the control plane, followed by the data plane and application layers[22]. The types of attacks targeting SDN systems at this stage include, among others, flow- rule manipulation, buffer overflow, TCP SYN flooding, and unauthorized access attempts[23]. Another vulnerability to be taken into account is that of scalability, as the increase in SDN network size results in a greater vulnerability of the controller[24]. This problem has been studied experimentally by experts who concluded that large scale DDoS attacks were able to lead to SDN controller failure and cause dropping of legitimate traffic as well as overall network collapse[25].

All in all, the use of centralization, programmability, and open communication channels in SDN significantly increase its vulnerability. This aspect calls for the development of effective real time attack detection methods.

2.1.3 Intrusion Detection Systems in SDN

Intrusion Detection Systems (IDS) are defined as analytical frameworks aimed at detecting behaviors and anomalies that are characteristic of intrusion and exploitation of network resources and host systems[26]. IDS is considered an auxiliary system that performs the task of detecting and distinguishing between legitimate and illicit activities, providing the relevant personnel with detailed notifications or automatically responding to minimize the impact of the intrusion[27]. Essentially, the fundamental concept behind IDS is the process of monitoring data and extracting the relevant indicators of threats using established rules or models developed through the analysis of historical data[28]. Intrusion Detection Systems are implemented using both active and passive techniques. IDS sensors are installed at critical points of the system infrastructure, which could be at the host level, network level, or using a hybrid and distributed system[29]. Data, which includes network packets and system/application logs, is gathered and processed through a series of stages, which include feature extraction, data reduction and processing, pattern or statistical analysis, machine learning, incident classification, and alerting[30]. IDS is integrated with the Incident Response Systems (IRS) or Intrusion Prevention Systems (IPS) to perform automatic responses, which could include blocking the source IP address, changing the firewall rules, or terminating the session[31] .The classification of IDS and its detection mechanisms is illustrated in **Figure 2**[32].

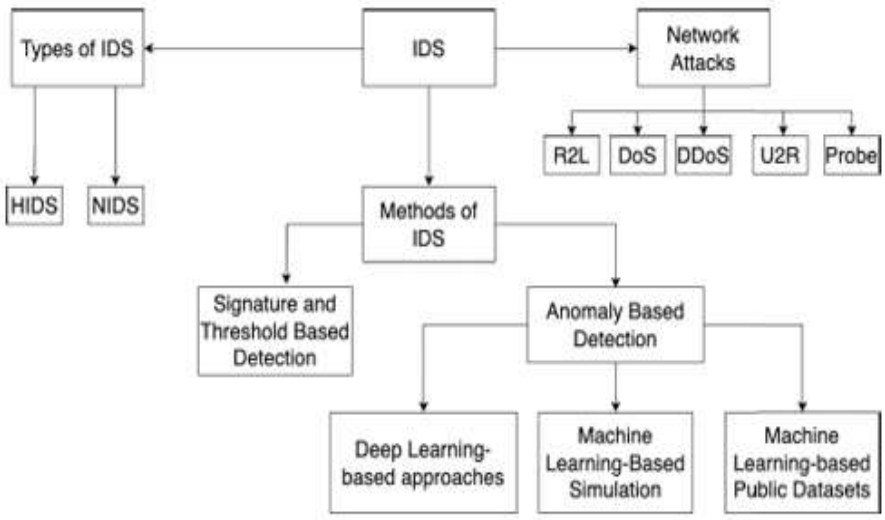

Figure 2. Classification of Intrusion Detection Systems (IDS), Detection Methods, and Network Attacks [32]

Figure 3 illustrates the hierarchical classification of intrusion detection systems (IDSs) according to deployment type, detection methodology, and targeted attack categories. From the deployment perspective, IDSs are categorized into Host-Based Intrusion Detection Systems (HIDS), which monitor activities occurring on individual hosts, and Network-Based Intrusion Detection Systems (NIDS), which analyze network traffic to identify malicious behavior. The figure further classifies IDS methodologies into signature-based detection and anomaly-based detection. Signature-based approaches rely on predefined attack patterns and are effective for detecting known threats, whereas anomaly-based approaches identify deviations from normal network behavior and are therefore more suitable for detecting unknown or zero-day attacks. In addition, the figure highlights common attack categories frequently addressed in IDS research, including Denial of Service (DoS), Distributed Denial of Service (DDoS), Probe, User-to-Root (U2R), and Remote-to-Local (R2L) attacks. Finally, the anomaly-based branch demonstrates the growing adoption of machine learning, deep learning, and simulation-based approaches for enhancing intrusion detection performance in modern SDN environments.

The benefits of using IDS include the ability to identify attacks that do not have signatures, improved detection within the context of the entire system, increased traceability and digital forensic capabilities, and the ability to support regulatory compliance[33]. Intrusion Detection Systems are considered instrumental in the study of attacks and the development of models that are adapted to the changing patterns of attacks. Additionally, IDS is considered intelligent systems that improve network operations through the reduction of incident response times and the enhancement of decision making[34].

2.1.4 Types of Intrusion Detection System

Intrusion Detection Systems can be broadly classified into several categories based on their detection mechanism and underlying methodologies, this types illustrated in **figure 3**[35] .

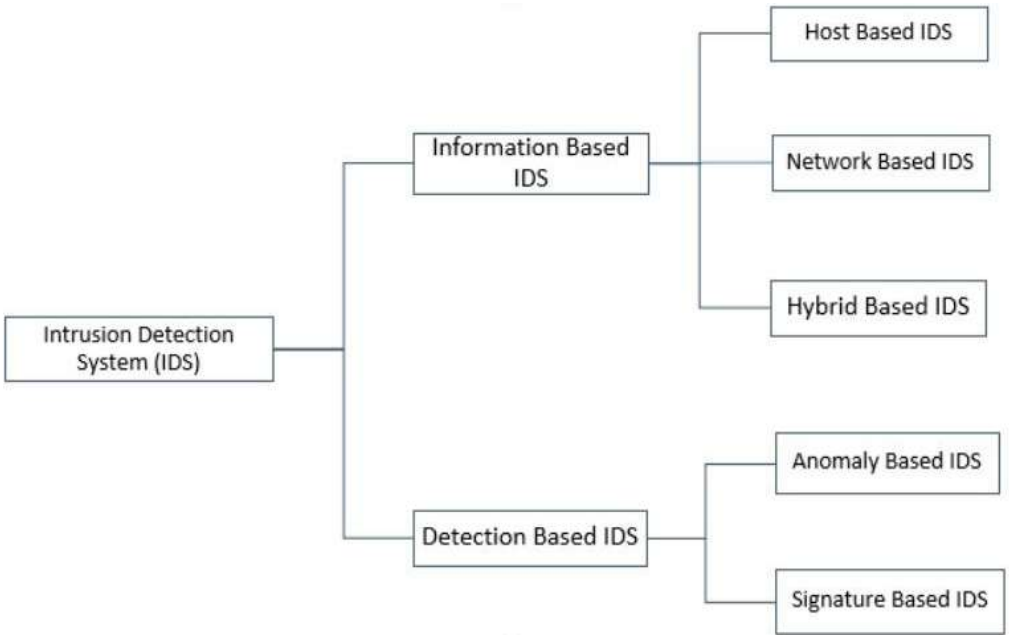

Figure 3. Types of IDS [35].

1.Based on Deployment

1.1 Host Based IDS (HIDS)

The Host-Based Intrusion Detection System (HIDS) is a type of intrusion detection technology that is installed at the host level to examine the internal activity of the system, which includes logs, file modification, and user activity. This system operates through monitoring the activities inside the host and matching them against pre-defined behaviors or rules. The main strength of HIDS is that it can detect any type of malicious activity inside the system, and this capability allows the system to monitor activities inside the host. On the other hand, the main weakness of the HIDS technique is that it cannot be used to detect large scale network attacks as well as generate unnecessary load on the host [36].

1.1 Network Based IDS(NIDS)

Network-based Intrusion Detection System (NIDS) works by analyzing and examining the packets moving through the network. This intrusion detection system is generally located in crucial locations like gateways and switches to be able to examine network traffic coming from various nodes. NIDS functions by intercepting packets and applying various techniques for detecting intrusions, such as signatures and anomalies. The strength of using NIDS lies in its ability to detect massive attacks such as denial of service and distributed denial of service throughout the whole network. It does not burden the host with an extra workload and provides a wider perspective of network operations. Nevertheless, NIDS faces difficulties in handling encrypted traffic and may experience performance issues in fast-paced networks [37].

1.3 Hybrid Based IDS

Hybrid IDS combines both host-based and network-based detection approaches to achieve comprehensive monitoring. It operates by collecting and correlating data from both system level and network level sources. This integrating enhances detection accuracy and provides a more complete view of potential threats. Hybrid systems can effectively detect both internal and external attacks. However, hybrid IDS introduces higher complexity in deployment and management, as well as increased computational and communication overhead [38].

2. Based on Detection Technique

2.1 Signature Based IDS

The principle underlying signature-based intrusion detection systems (IDS) lies in matching the behavior of the attacker against a library of already known attack signatures. Once such a match is made, the system identifies the behavior as an intrusion. The approach is highly effective and very accurate with fewer false positives in recognizing any known attack. The technique has been widely adopted in conventional IDS systems due to its simplicity and effectiveness. However, its biggest drawback is that it cannot detect any unknown attacks, since the detection process depends entirely on the patterns already recognized [39].

2.2 Anomaly Based IDS

The anomaly-based intrusion detection systems identify unauthorized access through deviations from the usual activity or system behaviors. In this regard, the technique involves developing a baseline model of the usual activities and matching the real-time activities against this baseline model. One of the advantages of using an anomaly-based IDS system is that it can effectively detect attacks that have never been encountered before, including zero-day attacks. This makes it more applicable in dynamic and ever-changing environments. However, one challenge with these systems is that they usually generate high levels of false positives due to the variation that exists in usual behaviors. Machine learning techniques are thus applied to minimize this problem[40].

From the above classification, it is apparent that there exist advantages and disadvantages associated with each of these IDS approaches. While traditional approaches like the signatures-based approach are fast reliable, they lack the flexibility needed in a rapidly changing environment. Conversely, anomaly based and machine learning approaches are flexible but come with higher complexities. The aforementioned factors thus make the development of ensemble models ideal in SDN-based IDS [41] [42].

2.1.4 Datasets for SDN

The effectiveness of intrusion detection system in SDN is highly dependent on the quality and representativeness of the datasets used for training and evaluation. In the literature, datasets employed for SDN-based IDS can be categorized into traditional benchmark dataset, modern network datasets, DDoS - focused datasets, and SDN-specific datasets. Each dataset provides different characteristics in terms of traffic realism, attack diversity, and relevance to SDN environments.

1.Traditional Benchmark Dataset

- **KDD Cup 99**

KDD Cup 99 is one of the earliest and most widely used datasets for ID research [43]. It contains a large number of network traffic records labeled as either normal or attack, with attack categories such as DoS, Probe, R2L, and U2R. It has been extensively used due to its availability and its role as a standard benchmark for evaluating classification algorithms. Researchers often use it for baseline comparison due to its widespread adoption. However, KDD Cup99 suffers from several critical limitation. It contains a significant amount of redundant records, which can bias learning algorithms toward patterns. Additionally, the dataset is outdated and does not reflect modern network traffic or advanced attack techniques. Most importantly, it does not capture SDN characteristics such as centralized control or flow-based communication. making it less suitable foe SDN-based IDS evaluation.

- **NSL-KDD**

NSL-KDD was developed as an improved version of KDD Cup99 to address issues such as data redundancy and unbalanced difficulty levels. It provides a more balanced dataset and is widely used for fair performance comparison among different machine learning models. Due to its cleaner structure, it is still frequently adopted in IDS research. Despite these improvements, NSL-KDD still inherits the fundamental limitations of KDD Cup99. It does not represent modern network environments, lacks realistic traffic diversity, and does not include SDN-specific Features. Therefore, while useful for benchmarking , it is not ideal for evaluating IDS in contemporary SDN system [44].

2.Modern Network Intrusion Dataset

- **UNSW-NB15**

UNSW-NB15 is a modern dataset generated using the IXIA perfect storm tool[45], It includes a mixture of normal and malicious traffic and covers multiple attack types such as exploits, reconnaissance. DoS Worm, and shellcode. The dataset is designed to represent contemporary network behavior and provides a more realistic alternative to techniques, However, UNSW-NB15 is not specifically designed for SDN environments, it lacks representation of SDN architecture component such as controllers and flow rules. Therefore, while it improve realism compared to older datasets, it may not fully capture SDN-specific attack patterns[46].

- **CIC-IDS2017**

CIC-IDS2017 is one of the most widely used modern datasets for intrusion detection. It includes realistic network traffic and various attack scenario such as DDoS, brute force, botnet, and web attack. A key advantage of this dataset is the availability of both raw traffic (PCAP) and floe-based features, which makes it suitable for different types of analysis. It also reflects modern user behavior and attack strategies more accurately than legacy dataset. Despite these advantages, CIC-IDS20217 is not specifically tailored for SDN, it does not model SDN- specific interactions such as controller switch communication. Additionally , some studies have pointed

out issues related to data preprocessing and labeling consistency, which may affect reproducibility[47].

- **CSE-CIC-IDS2018**

This dataset extends CIC-IDS2017 by introducing more attack scenarios and larger scale traffic data. It includes various types of DDoS traffic such as SYN floods. UDP floods, DNS amplification, and reflection attacks. This dataset is highly valuable for evaluation IDS models focused on DDoS detection, especially in SDN environments where the controller is vulnerable to such attacks. However, its main limitation is its narrow focus. It only address DDoS attacks and does not provide a comprehensive det of intrusion types. As a result, it is not suitable for evaluating multi-class IDS that aim to detect a wide of attacks[48].

3. DDOS Focused Dataset

- **CIC-DDoS2019**

CIC-DDoS2019 is specifically designed for detecting Distributed Denial of Service (DDoS) attacks. It includes various types of DDoS traffic such as SYN floods, UDP floods, DNS amplification, and reflection attacks. This dataset is highly valuable for evaluating IDS models focused on DDoS detection, especially in SDN environments where the controller is vulnerable to such attacks. However, its main limitation is its narrow focus. It only addresses DDoS attacks and does not provide a comprehensive set of intrusion types. As a result, it is not suitable for evaluating multi-class IDS that aim to detect a wide range of attacks [48].

- **CAIDA DDoS Dataset**

The CAIDA dataset contains real world traffic trace of large-scale DDoS attacks. It is often used for traffic analysis and anomaly detection studies. Its main strength lies in its realism, as it reflects actual attack behavior in operational networks. This makes it useful for validating detection systems under realistic conditions. However, the dataset id not structured for machine learning tasks and often requires significant preprocessing. It also lacks labeled multi-class attack categories and does not represent SDN specific scenarios [49].

4. SDN Specific Dataset

- **INSDN**

INSDN is one of the most relevant datasets designed specifically for SDN intrusion detection. It includes both normal traffic and multiple attack types generated within an SDN environment. Its primary advantage is that it reflects SDN architecture, including floe- based communication and controller-related behavior. This makes it more suitable for evaluating IDS models tailored for SDN compared to traditional and general datasets. Additionally, InSDN supports multi-class classification tasks, enabling the evaluation of models across different attack categories. However, InSDN is relatively less widespread compared to datasets like CIC=IDS2017.This may limit direct comparison with a large number of existing studies. Furthermore, like any synthetic dataset, it may not fully capture a real world SDN deployment scenarios [50].

5. IOT and SDN -IOT Related Dataset

- **BOT -IOT**

BOT-IOT is designed to represent IoT environments with botnet related attacks. It includes realistic traffic patterns and various attack scenarios targeting IoT devices. It is useful for studies involving SDN enabled IoT systems, where SDN is used to manage IoT networks. However, it is not specifically designed for SDN. Therefore , its relevance depends on whether the research focuses on SDN-IoT integration rather than pure SDN environment [51].

- **TON-IOT**

TO-IoT is a recent dataset that includes telemetry data and network traffic for IoT and industrial IoT systems. It provides a comprehensive view of modern cyber physical environments and is useful for evaluating advanced detection systems. Nevertheless. It is not pure SDN dataset. and its applicability to SDN-based IDS must be justified carefully [52].

Dataset	Strengths	Limitations	Research Gap
NSL-KDD	Widely used benchmark dataset and easy comparison with previous studies	Outdated traffic patterns, lacks modern attack scenarios, not designed for SDN environments	Development of updated datasets reflecting contemporary SDN traffic and attack behaviors
KDD Cup 99	Large dataset and extensive use in IDS research	High redundancy, severe class imbalance, outdated network traffic	Creation of realistic datasets with reduced redundancy and improved attack diversity
UNSW-NB15	Includes modern attack categories and realistic traffic features	Limited SDN-specific characteristics and network programmability features	Integration of SDN-oriented traffic and controller-related attack scenarios
CICDDoS2018	Contains large-scale DDoS attack traffic and realistic network behavior	Primarily focuses on DDoS attacks and lacks attack diversity	Inclusion of multiple attack categories relevant to SDN environments
CICDDoS2019	Improved representation of recent DDoS attacks	Limited coverage of non-DDoS attacks and SDN-specific features	Development of comprehensive SDN datasets containing diverse attack types
InSDN	Specifically designed for SDN intrusion detection research and includes SDN-related traffic characteristics	Limited adoption in the literature and fewer comparative studies	Wider validation and benchmarking using different IDS and ensemble learning approaches

Table (1). Limitation and Research Gaps of Commonly Used Datasets in SDN-Based IDS Research

It appears from the study of currently popular data sets that none of the available data sets satisfy all of the criteria of the proposed intrusion detection framework for SDNs. While the older data sets, namely, NSL-KDD and KDD Cup99, continue to enjoy widespread popularity, their shortcomings include the inability to emulate current network activities and to account for the unique features of SDN. Some of the modern data sets, such as UNSW-NB15 and CICDDoS2018/2019, demonstrate better authenticity, but they also suffer from lack of diversity and SDN-relevance.

2.2 Related Work

Recent studies have extensively explored the application of machine learning and ensemble learning techniques for intrusion detection in Software-Defined Networking (SDN). These approaches aim to improve detection accuracy, reduce false alarm rates, and enhance adaptability against evolving cyber threats. The following section critically reviews the most relevant contributions in this field:

Early research efforts explored the integration of traditional machine learning and deep learning models. Dey and Rahman (2020) proposed a flow-based anomaly detection framework using both conventional classifiers and deep learning models, where GRU-LSTM achieved higher accuracy compared to Random Forest [53]. In a similar direction, Alzahrani and Alenazi (2021) demonstrated that tree-based ensemble models such as Random Forest and XGBoost significantly improve IDS performance when combined with effective feature selection [54].

Several studies have focused on DDoS detection in SDN environments. Alamri and Thayananthan (2020) proposed a hybrid detection and mitigation system combining adaptive bandwidth control with XGBoost, achieving high accuracy and extremely low false positive rates [55]. In their subsequent work (2021), they conducted a comparative analysis of multiple classifiers, showing that XGBoost outperforms traditional methods such as SVM, KNN, and Naïve Bayes in detecting DDoS attacks [56].

More recent works have emphasized hybrid and optimization-based approaches. Logeswari et al. (2023) introduced a hybrid feature selection framework combining CFS and RF-RFE with LightGBM, demonstrating improved detection accuracy and reduced computational overhead [5]. Similarly, Omer et al. (2023) proposed a model combining Firefly Optimization with Probabilistic Neural Networks, highlighting the effectiveness of meta-heuristic optimization in enhancing IDS performance [57] .

In addition, research has expanded toward integrating SDN with emerging environments such as Industrial Internet of Things (IoT). Alshahrani et al. (2023) developed an SDN-based IDS using SVM and Decision Tree models, demonstrating high detection accuracy and real-time response capabilities in industrial environments [58] .

Rafin et al. (2026) proposed an SDN intrusion detection framework based on feature selection, class imbalance handling, ensemble learning models, and SHAP-based interpretability. Their study reported a detection accuracy of 99.96% using the InSDN dataset, demonstrating the potential of ensemble methods for developing accurate and interpretable SDN-based intrusion detection systems[59] .Likewise, Ahmed and Abdulkader (2024) proposed a soft voting ensemble combining Random Forest, Gradient Boosting, and Logistic Regression, demonstrating that ensemble methods significantly outperform individual classifiers [60].

Finally, recent advancements have explored SDN-specific datasets and optimization strategies. More and Kachavimath (2025) utilized the InSDN dataset and applied Genetic Algorithm (GA) and Ant Colony Optimization (ACO) for feature selection, achieving high detection accuracy and highlighting the importance of combining feature optimization with machine learning techniques in SDN-based IDS [61].

Saleh and Hamad (2026) introduced the SICA dataset, a comprehensive SDN-IoT intrusion detection dataset containing 22 attack types collected in a multi-controller SDN environment. The study compared the proposed dataset with existing benchmarks such as InSDN and SDN-IoT and demonstrated that modern SDN-oriented datasets can provide richer attack diversity and more realistic traffic characteristics for evaluating deep learning-based IDS models. The authors also highlighted that relying solely on traditional datasets may limit the generalization capability of intrusion detection systems when deployed in real-world SDN environments [62].

Despite the significant progress achieved by these studies, several limitations remain. Most existing approaches rely heavily on traditional datasets such as NSL-KDD and KDD Cup 99, which may not accurately represent the dynamic nature of SDN environments. Additionally, many studies focus primarily on accuracy while overlooking important factors such as scalability, real-time deployment, and generalization.

Furthermore, although ensemble learning techniques have demonstrated superior performance, limited attention has been given to the role of feature diversity and the integration of multiple feature selection strategies within ensemble frameworks, indicating a clear research gap. To provide a clear and structured overview of recent research efforts in SDN-based intrusion detection systems, a comparative summary of the most relevant studies is presented in **Table 2.**

Ref	Year	Dataset	Methodology	Accuracy

[51]	2020	NSL-KDD	RF NB DT BN RBF Network	82%
[53]	2020	CICDDoS2019 NSL-KDD CAIDA	DT RF LR NB	99.9%
[52]	2021	NSL-KDD	DT RF XGBOOST	95.55%
[54]	2021	CICDDoS2019	DT RF NB KNN SVM XGBOOST	99.7%
[55]	2023	NSL-KDD	DT RF KNN SVM XGBoost NB	98.72%
[56]	2023	KDD-CUP99	Decision Tree FFO-PNN KNN ANN	98.99%
[57]	2023	NSL-KDD	Liner SVM Quadratic SVM Fine Tree	99.7%
[58]	2024	UNSW-NB15	GB RF AdaBoost DT	99.87%
[60]	2025	INSDN	NB LR KNN	99.8%

[59]	2026	INSDN	RF, ET, XGBoost, Stacking, SHAP	99.96%
[62]	2026	SICA, InSDN, SDN-IoT	**DNN, RNN, LSTM, 1D-CNN**	97.58%

Table (2): Comparative Analysis of IDS Approaches in SDN.

3. Review Methodology

This section describes the methodology adopted to conduct this review on IDS in SDN environments, with a focus on machine learning and ensemble-based approaches. The review process is guided by a set of research question and follows established guideline for systematic literature review see.

3.1 Research Questions (RQs)

To guide this review, a set of research questions (RQs) was formulated to analyze existing studies on SDN-based intrusion detection system using machine learning and ensemble techniques. These questions focus on identifying commonly used methods, evaluating their effectiveness, and highlighting current research challenges and future directions as show in **table 3**.

ID	Research Question	Expected Outcome
RQ1	What machine learning and ensemble-based techniques have been applied for intrusion detection in SDN environment?	To identify the most frequently used algorithms and ensemble models in SDN-IDS research
RQ2	How effective are ensemble-based approaches compared to traditional single classifiers in improving detection accuracy and reducing false alarms?	To evaluate the comparative performance and advantages of ensemble methods.
RQ3	What are the major research challenges, limitation, and open issues in applying machine learning for SDN intrusion detection?	To highlight current research gaps and suggest direction for future studies.

Table (3): Show the Research Questions.

3.2 Search Terms

Relevant Keywords related SDN and IDS were used to retrieve studies, including "Software Defined Network (SDN)", "Intrusion Detection System (IDS)", "Machine Learning", "Ensemble Learning", "Dataset", "Network Security", and "Cyber Attack". Boolean operators such as AND, OR were applied to refine the search and improve the relevance of the results.

3.3 Data Sources

The literature was collected from several well-known digital libraries including IEEE Explore, Science Direct, Springer Link, ACM Digital Library, and Wiley Online Library. Additional sources such as Google Scholar and Research Gate were used to ensure comprehensive coverage of relevant studies in SDN and IDS.

3.4 Review Framework

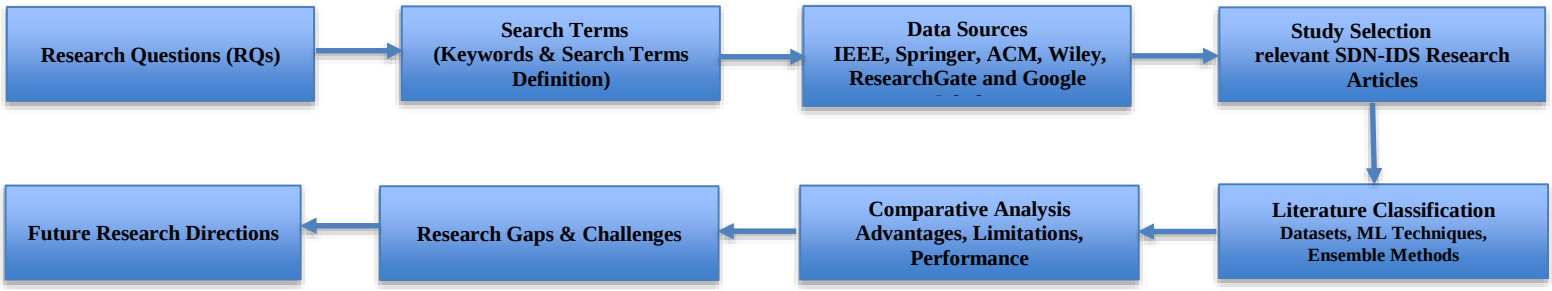

Figure 4. Review Framework Adopted for the Analysis of Ensemble Learning-Based IDS Approaches in SDN Environments.

Figure 4 highlights the review model used in this study. The first step in this procedure includes formulating the research problem along with identifying key search terms that are then used for searching relevant literature from reputable databases. These collected studies are then categorized and analyzed comparatively in order to highlight strengths, weaknesses, potential areas for further investigation, and directions for future research regarding IDS solutions based on ensemble learning within SDN context.

3.5 Study Analysis Process

After conducting the study process to find out the best studies related to the topic, a detailed analysis was carried out to identify and assess the most important findings about intrusion detection systems in software-defined networking (SDN). This review paper is based on the selected research papers which were analyzed in accordance with several criteria, namely, utilized dataset, employed machine learning algorithms, ensemble learning approach, performance measure, and achieved result.

Moreover, a comparative analysis was provided to reveal the advantages, weaknesses, and features of the mentioned methods. Specifically, the attention was paid to ensemble learning since it has become an essential element in enhancing the performance and robustness of intrusion detection systems. The findings presented below reflect the problems and research directions identified in relation to intrusion detection systems in SDNs.

The gathered information was then structured into comparative tables and discussion sections to enhance the comprehension of contemporary research trends and advancements in the subject.

4. Machine Learning Techniques

The Machine Learning approach serves as a fundamental element of the current IDS because it allows for the automated identification of patterns and malicious activity based on network traffic data [63]. Unlike the conventional rule-based methods, which rely on predetermined signatures , machine learning generalizes knowledge and recognizes both known and unknown attacks [64].In SDN the centralized design enables gathering global information about network traffic, including flows statistical information ,packets characteristics, and behavior [31]. Therefore , ML is a powerful technique for IDS in the SDN environment because it utilizes global network information for analysis, making it more efficient compared to conventional approaches that operate on local data only [65]. Various types of machine learning algorithm are widely used in IDS , which include but are not limited to classification, anomaly detection, feature selection, and traffic analysis [66]. ML techniques enable processing massive network traffic datasets, detection patterns that suggest the presence of malicious activity [25].

4.1 Categories of Machine Learning

ML techniques used in IDS can be broadly categorized into three main types : Supervised, Unsupervised, and Semi-Supervised Learning [67].

4.1.1 Supervised Learning

Supervised learning is the most commonly used approach in IDS. In this method, the model is trained using labeled datasets, where each data instance is associated with a predefined class label such as normal traffic or specific attack types [68]. During the training phase, the model learns the relationship between input features (e.g., packet size, flow duration, protocol type) and output labels. Once trained , the model can classify new unseen data into corresponding categories [67]. Supervised learning is widely adopted in IDS due to its high classification accuracy and reliability when sufficient labeled data is available. It is particularly effective for detecting known attack patterns and performing multi-class classification [68]. However, this approach has several limitations. This requires large amounts of labeled data, which can be difficult and time consuming to obtain. Additionally, supervised models may struggle to detect zero-day attacks that were not present in the training dataset [69].

Common supervised learning algorithms include Decision Tree (DT), Random Forest (RF) , Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naïve Bayes (NB), and Artificial Neural Networks [70].

4.1.2 Unsupervised Learning

Unsupervised learning dose not rely on labeled data. Instead , it analyzes the structure of the input data to identify patterns, similarities, or anomalies [71] . In IDS, unsupervised learning is primarily used for anomaly detection. The model learns the normal behavior of network traffic and flags deviations as potential attacks [72]. This makes it particularly useful for detecting unknown or novel threats [30]. Common techniques include clustering algorithms such as K-Means and DSCAN, as well as statistical and dimensionality reduction methods like Principle Component Analysis (PCA) [37].

The main advantage of unsupervised learning is ability to detect previously unseen attacks without requiring labeled data. However , it often suffers from higher false positive rates, as not all anomalies correspond to malicious behavior [73] [74].

4.1.3 Semi-Supervised Learning

Semi supervised learning is a hybrid approach that combines both labeled data and unlabeled data. This method is particularly useful in intrusion detection , where labeled data is limited but large volumes of unlabeled traffic are available [75]. In this approach, the model is initially trained using a small labeled dataset and then refined using unlabeled data. This improves generalization and enables the model to learn representative patterns of network behavior [76]. Semi-supervised learning provides a balance between supervised and unsupervised approaches [77]. It achieves better performance than purely unsupervised methods while reducing the dependency on labeled data [78]. However, it may require more complex training procedures and careful data handling [79].

4.2 Classification Algorithms

The classification algorithms play an important role in intrusion detection systems, carrying out the differentiation process of identifying whether the network traffic is legitimate or not using the extracted attributes from the traffic [25]. The classification algorithms can be grouped according to the mathematical model used, ranging from linear classification, quadratic classification, SVM, kernel methods, decision trees, artificial neural networks, and prototype-based learning techniques [6] [80].

4.2.1 Linear Classifiers

4.2.1.1 Logistic Regression

The Logistic Regression Classifier uses the idea of probabilistic linear classifiers to build up models. The model uses the sigmoid function to represent the relationship between input features and probabilities of classes .[81] Hence, the method can predict the likelihood of a particular class for a given input instance. Although logistic regression classifiers have proven to be extremely efficient in practice due to their ease of implementation and computational convenience in terms of modeling linear relationships, their performance deteriorates on complex non-linear problems [82].

4.2.1.2 Naïve Bayes Classifier

Naïve Bayes is a probabilistic classifier whose main principle comes from Bayes' theorem and the assumptions of conditional independence between features. Naïve Bayes calculates posterior probabilities of classes and predicts the class with the highest probability value. The computational cost is small and hence the efficiency of the classifier increases with the size of the dataset. However, naïve Bayes does not work effectively under conditions of feature dependency [83][84] .

4.2.2 Support Vector Machines

4.2.2.1 Support Vactor Machine (SVM)

Support vector machines are a powerful type of a supervised learning model whose task is to find an optimal separation hyperplane with maximal margin. SVM is efficient in high dimensional spaces and can deal with non-linearity using kernel trick. Possible kernels include RBF and polynomial functions. However, SVM is computationally costly to train and may require careful hyper-parameter selection [85].

4.2.5 Kernel Estimation Methods

4.2.5.1 K-Nearest Neighbor

K-Nearest Neighbor is a very simple instance-based non-parametric classifier. The decision-making process relies on finding similarities with k closest points. Similarities are usually determined using distances such as Euclidean distance. On the one hand, KNN classifiers are very intuitive and easy to implement. On the other hand, prediction is computationally costly and vulnerable to the presence of noise and irrelevant variables [86][83].

4.2.6 Decision Trees

4.2.6.1 Decision Tree (DT)

A decision tree (DT) is an example of a tree-based rule classifier. Such classifiers rely on dividing dataset into subsets based on feature values and forming a hierarchical structure where each node corresponds to some rule and leaves correspond to certain class labels. Decision trees prove to be highly interpretable but easily suffer from overfitting [87] .

4.2.6.2 Random Forest (RF)

Random Forest is an ensemble learning method involving building up multiple decision trees and selecting the most probable prediction by voting for it. RF classifiers solve the overfitting problem and enhance the prediction quality which makes them popular in IDS [88].

4.2.7 Neural Networks

4.2.7.1 Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) is an analog to the brain that consists of layers of interconnected neurons.

ANN can capture complex non-linear relationships in the data and is commonly used in intrusion detection to detect attacks. However, training of ANNs usually requires huge datasets, lots of computation power and precise hyper-parameter selection [89].

5. Feature Selection Techniques

The design of intrusion detection system models based on machine learning requires feature selection in view of the fact that network traffic datasets tend to have a large number of features. Feature selection entails the identification of important features that contribute positively to model performance. Feature selection seeks to obtain an optimal subset of informative features from high dimensional datasets, thereby increasing detection accuracy, decreasing model training time, minimizing overfitting, and enhancing model performance [90]. In SDN based IDS, feature selection becomes even more important in view of the very high number of flow attributes contained in network traffic data. Feature selection allows machine learning algorithms to focus on the most relevant features [91].

There are different techniques used in feature selection. Typically, all feature selection techniques fall under one of three categories, namely; filter techniques, wrapper techniques, and embedded techniques [92].

5.1 Filter Methods

Filter methods involve assessing the relevance of each feature individually using statistical tests. Filter methods do not incorporate any machine learning technique during the evaluation of feature relevance [93].

Examples of commonly used filter methods include the information gain method, chi-square method, correlation coefficient test, and ANOVA. The main advantage of filter methods is that they scale well for large datasets such as network traffic datasets used in intrusion detection applications [81]. Since filter methods involve independent assessment of each feature and are independent of any machine learning algorithm, they cannot result in overfitting, Filter methods can be computationally inefficient depending on the size of the dataset [94].

One major limitation of filter methods is that they tend to ignore feature dependency since each individual feature is evaluated independently. In some cases, ignoring the dependencies among features may result in a suboptimal set of selected features [95].

5.2 Wrapper Methods

Wrapper methods involve selecting an optimal subset of features by training and testing a classification model multiple times [96]. In this approach, a search strategy is used to identify subsets of features whose combination provides the best classification accuracy. Popular search strategies used in wrapper methods include forward selection, backward elimination, and recursive feature elimination [93]. In the case of recursive feature elimination, a set of features is eliminated iteratively until the best subset is obtained. Wrapper methods are computationally inefficient compared to filter methods [97]. However, they are more accurate than filter methods because they consider feature interaction [90]. In SDN- ID models, wrapper methods would perform optimally but may not be useful because of their inefficiency in high-dimensional datasets [81].

5.3 Embedded Methods

Embedded methods use the machine learning algorithm itself to incorporate feature selection. In this case, the machine learning algorithm learns the most relevant subset of features when training [98]. An example of an embedded method includes decision tree algorithms such as Random Forest [99]. Also, regression methods such as Lasso can be used in feature selection since the technique involves eliminating less significant features from a linear model. Embedded methods represent a balance between wrapper and filter methods. In other words, they are more efficient compared to wrapper methods [90]. However, they consider feature dependency just like wrapper methods [100]. Thus, their performance is superior to that of filter methods. In the design of IDS models, embedded methods are quite popular due to their efficiency and performance advantages [101].

6. Ensemble Learning

Ensemble learning can be regarded as a state-of-the-art technique in machine learning which combines a number of different models base learners to achieve improved performance of predictions [102]. Unlike relying solely on one classifier, ensemble learning techniques use a combination of different learners' output predictions in order to make a better decision [103]. The main idea of ensemble learning lies in the fact that different models may detect different patterns in the data. Thus, combining them leads to minimizing the weak sides of each model and maximizing their strong ones [42]. As was noted above, this is especially important when working with IDSs, since data in these cases tend to be complicated, imbalanced, and dynamic [60].
The application of ensemble learning to SDN-based IDSs has become a promising research direction since this technique can help in improving IDS accuracy and reduce the rate of false alarms [25].

6.1 Concept of Ensemble Learning

The general idea behind the concept of ensemble learning lies in building multiple classifiers in order to create a combined predictor outperforming any of the individual learners [104]. As previously mentioned, each base learner is trained using a given dataset, possibly applying different data subsets, features, or algorithms [105]. Next, the prediction made by each model is combined to produce the final result through such techniques as averaging or voting [106]. Thus, one can say that the idea of ensemble learning is to combine the results produced by multiple models to arrive at the prediction which will be superior to the predictions made by individual learners [25].
Such approach prevents overfitting and makes models more stable. In the context of intrusion detection, it helps in combining different detectors specializing in different kinds of attacks.

6.2 Types of Ensemble Methods

There exist different approaches to constructing and combining models in the framework of ensemble learning:

- **Bagging (Bootstrap Aggregating)**

Bagging involves training multiple models using data subsets built by drawing random samples of observations with replacement. Models are then combined into a final classifier via a process of aggregating their predictions through such techniques as voting or averaging. A well-known example of bagging technique is Random Forest algorithm involving multiple decision trees trained on different data subsets and features [42].

- **Boosting**

Boosting is a sequential ensemble learning technique in which models are trained iteratively, where each new model aims to improve the errors of the previous ones. In this approach, misclassified samples are given higher importance during subsequent training phases, allowing the model to focus on difficult cases, Common boosting algorithms include AdaBoost, Gradient Boosting, and XGBoost, which have shown strong performance in many applications. However, these methods can be sensitive to noisy data , as the increased focus on misclassified instances may amplify the effect of outliers [25].

- **Stacking (Stacked Generalization)**

Stacking implies training a meta-classifier aimed at integrating the predictions produced by the base learners into a final prediction. In this case, the prediction made by base learners serves as an input for another model predicting the output value. Staking proves to be a highly effective approach; however, it is relatively complicated to apply in practice [107].

- **Voting Algorithms**

Voting belongs to the simplest ensemble techniques in which multiple models are used to construct the final prediction.
Hard Voting: Final class is selected using voting
Soft Voting: Final class is determined by averaging class probability predictions

In most cases, soft voting leads to higher effectiveness because it takes into account confidence of each model [108].

6.3 Advantages of Ensemble Learning

There are several advantages provided by ensemble learning techniques which make them perfect for intrusion detection problems.

- High Accuracy: Combination of multiple learners leads to enhanced prediction.
- Robustness: Less sensitive to noise and outliers.
- Good Generalization Ability: Low risk of overfitting compared to individual classifiers.
- Handling Complicated Data: Works efficiently with multidimensional nonlinear datasets.
- Fewer False Positives: Important for IDS applications.

The advantages listed above are especially valuable in SDN environment because of its dynamics [109].

6.4 Limitations of Ensemble Methods

However, ensemble learning techniques suffer from the following disadvantages:

- Increased Complexity: Combining multiple classifiers adds up complexity to the system.
- Additional Computation Cost: Training of several models increases computation cost.
- Lower Transparency: Ensemble models are usually not easy to interpret.
- Difficult Implementation: Some techniques, like stacking, require careful implementation.

Considering the necessity of achieving real-time response, some of the disadvantages may negatively impact application of ensemble methods in SDN-based IDSs [110].

Thus, ensemble learning has proved to be a promising technique in developing IDSs based on SDNs. Using techniques like bagging (Random Forest) to increase robustness and boosting to give attention to complicated objects, one can achieve even better results by applying techniques like voting and stacking. However, selection of a proper strategy depends on achieving trade-off between accuracy and computational cost.

6.5 Evaluation Metrics

The evaluation of IDS requires a set of well-defined performance metrics that can effectively reflect the classification capability of the developed models. These metrics are essential for assessing how accurately the system distinguishes between legitimate network traffic and malicious activities [63].

Accuracy is commonly adopted as primary indicator, representing the ratio of correctly classified instances to the total number of samples. Despite its simplicity, relying solely on accuracy may lead to misleading conclusions, particularly in imbalanced datasets where the number of normal instances significantly exceeds attack samples [111].

Precision focuses on the reliability of positive predictions by quantifying the proportion of correctly identified attack instances among all instances predicated attacks. A higher precision value indicates fewer false alarms ,which is crucial in practical deployment to reduce unnecessary system responses [111].

Recall also referred to as the detection rate, evaluates the model's ability to identify actual attack instances. It measures the proportion of true attack samples that are successfully detected. In the context of cybersecurity, achieving a high recall is critical , as undetected attacks may lead to severe system compromise[112].

The F1-Score combines both precision and recall into a single metric using their harmonic mean, offering a balanced evaluation of the model's performance. This metric is particularly valuable in scenarios involving class imbalance, where tradeoffs between false positives and false negatives must be carefully managed [113].

In addition to these core metrics, supplementary measures such as the false positive rate (FRR) and Area Under the Receiver operating characteristic Curve (AUC-ROC) are frequently utilized [114].

These metrics provide deeper insight into the model's discrimination ability and its behavior across different classification thresholds; Therefore, a comprehensive evaluation of IDS performance should not rely on a single metric. Instead , employing multiple complementary metrics ensures a more realistic and robust

assessment of the models effectiveness in detecting network intrusions [50].

7. Discussion

According to previous research, machine learning techniques have proved efficient tools in improving intrusion detection systems (IDS) in SDN. This is primarily because of the inadequacy of the signature-based system in SDN. Due to the unique nature of SDN networks, they require an advanced detection method that guarantees reliability, accuracy, and security. The key challenge posed by SDN to security is the central role played by the controller. It is clear from numerous studies that the performance of an IDS is highly dependent on the test dataset utilized. Researchers have been using common test sets such as the KDD Cup 99 and NSL-KDD test sets for benchmarking. Although these datasets are useful, they cannot appropriately depict contemporary traffic conditions in the SDN network environment. The reason behind this inadequacy is due to the obsolescence of these test sets and lack of SDN-specific traffic characteristics. Therefore, even when an IDS achieves a high level of accuracy on such datasets, it may not necessarily apply in a real-life scenario. Modern test sets such as UNSW-NB15, CIC-IDS2017, CSE-CIC-IDS2018, and CIC-DDoS2019 are more realistic and present different forms of attacks. They provide a wider variety of attacks in comparison to common datasets. Nevertheless, they do not possess SDN-specific traffic characteristics such as interaction between switches and controllers and flow rules behavior. To fill this gap, researchers have created specialized test sets such as InSDN that cater to SDN. Unfortunately, these specialized test sets are less commonly used since there is no established standard dataset. In terms of the techniques used, supervised learning continues to dominate among all techniques due to its high levels of accuracy. However, the usefulness of supervised learning relies on the availability of labeled data, which is hard to come by in practice. On the other hand, the unsupervised and semi-supervised learning techniques, which are better suited for detecting novel attacks, are rarely used in intrusion detection applications in SDN. The use of these two techniques is quite low due to the prioritization of optimization in intrusion detection systems. Classification algorithm plays an important role in enhancing IDS's detection capabilities. Tree-based classification and ensemble algorithms outperform the commonly-used classification algorithms in almost all the studies analyzed. Classification algorithms such as Random Forest, Gradient Boosting, XGBoost, and Light GBM perform best compared to others due to high dimensionality handling capability, pattern recognition, and overfitting mitigation. It should be noted that high levels of accuracy in classification can be attributed to the collaboration of dataset, model, features selected, and evaluation technique used rather than the model itself. Feature selection techniques are paramount in achieving high levels of accuracy while improving model efficiency. Numerous research works show the significance of proper feature selection in enhancing IDS performance in terms of accuracy, processing cost, and interpretability. Filter-based techniques, although efficient and scalable, generate less informative feature subsets. Conversely, wrapper-based techniques generate more relevant features but at the expense of additional computations. Embedded technique allows for the simultaneous feature selection and learning process. Feature selection plays a vital role in IDS performance in conjunction with a classifier.

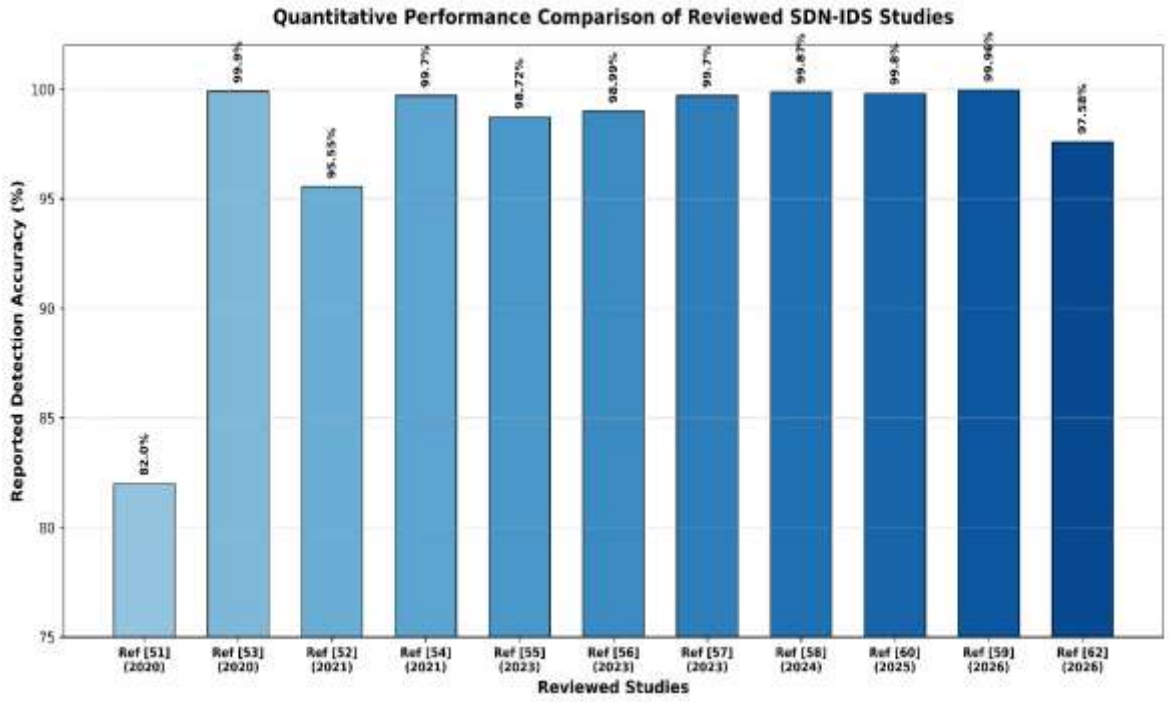

Figure 5. Quantitative comparison of detection accuracy reported by the reviewed SDN-based intrusion detection studies.

Figure 5 is illustrating a quantitative comparison of detection accuracy documented in the examine SDN-based intrusion detection research. Most investigations demonstrated detection accuracies beyond 95% highlighted the efficacy of machine learning and ensemble learning methodologies in SDN-based IDS. Studies utilizing ensemble learning methods, such as Random Forest, Gradient Boosting, and AdaBoost, produced the greatest performance values, routinely attaining accuracy levels near 100%. However, variation in datasets, assault scenarios, feature selection methods, and evaluation strategies necessitate caution when making direct comparisons among research**.**

One of the noticeable trends is the application of ensemble algorithms for improving IDS performance. This includes Bagging, Boosting, and Voting techniques which exploit the complementarity among models. For example, the Bagging technique increases robustness via variance reduction, while Boosting techniques increase accuracy through the exploitation of difficult samples. Nevertheless, regardless of the high level of accuracy achieved by using ensemble learning algorithms, very few authors address challenges related to deployment of models such as computational cost and latency. Moreover, another challenge is that a considerable number of articles focus only on one form of attack; this is the case for DDoS attacks. Even though DDoS attacks are common and applicable in SDN environments due to the involvement of controllers, this concentration limits the development of multi-class IDS. Most of the IDS presented are optimized for particular attack types.

Finally, concerning evaluation metrics, accuracy is widely used in measuring model performance. However, this metric is sometimes inadequate especially for imbalanced datasets. Additional measures such as Precision, Recall, F1-Score, and False Positive Rate can be used. Therefore, depending only on accuracy is misleading.

Overall, this analysis illustrates the considerable advancements in applying machine learning techniques to develop IDS in SDN. However, there are certain challenges including the use of unrealistic test sets, dependence on supervised learning, absence of real time implementation, and different evaluation criteria. Even though the superior detection rates of ensemble learning algorithms in SDN-based intrusion detection systems have been proven, implementing those algorithms into practice presents a challenging

barrier. The overwhelming majority of the current studies focus more on detecting attacks accurately and less on taking care of computational cost and real-time implementation needs.

The main advantage of using ensemble models is related to combining several classifiers to increase the prediction power and robustness of the system. This benefit might result in increased computational cost, increased memory consumption, and prolonged processing times. These factors can negatively affect the speed of detection by SDN-based intrusion detection systems, especially in the case of large networks that require prompt analysis of the data. Besides, the centralized approach of SDN makes these problems even more urgent since the role of the controller in monitoring all the actions taken in the network increases.

In such a way, using advanced detection techniques and high computational costs can delay the process of attack detection. One of the major issues is scalability. Even though a lot of projects prove their effectiveness using benchmark datasets, there are few studies assessing their methods in actual situations or for larger SDN. As a result, the practical value of many ensemble learning solutions for IDSs stays unclear. In this situation, future research should focus on improving the accuracy rate as well as on creating lightweight and scalable approaches to SDN.

8. Conclusion

This study provided through assessment of intrusion detection systems (IDSs) inside Software-Defined Networking (SDN) environments, focusing specifically on machine learning and ensemble learning methodologies. Machine learning algorithms, and contemporary ensemble learning methodologies utilized intrusion detection.

As follows from the study, the quality and representativeness of training and validation datasets used play an important role in the effectiveness of IDS models' performance. Despite the widespread use of conventional datasets, including KDD Cup99 and NSL-KDD, their limited ability to reflect modern SDN traffic makes their usage ineffective for current research. Datasets that better resemble reality, including InSDN, are relatively newer and more appropriate. Nevertheless, there are still many challenges connected to data standardization, attack diversity, and representativeness.

Another point found in the review is that ensemble machine learning methods outperform conventional single classifiers in terms of accuracy, robustness, and generalization. Methods such as bagging, boosting, or voting benefit from the capabilities of multiple models and make predictions even better than in conventional cases. As for the disadvantages of such approach, the increase in computational load, higher hardware requirements, and decreased model interpretability are worth mentioning.

Several gaps in research related to the problem of IDS in SDN environment were revealed during literature survey. The first gap concerns the need for standardized SDN datasets. Another gap is associated with using exclusively supervised learning algorithms without considering alternative approaches. Moreover, real-time limitations are not always examined in papers. Finally, there is a tendency to study mainly a certain attack type, e.g., DDoS attack, despite the variety of existing attack types.

Future works may be conducted to develop SDN datasets, apply unsupervised and semi-supervised learning to detect new threats, create lightweight IDS models for real-time use, and incorporate the XAI into ensemble learning methods.

References

[1] B. Isong, R. R. S. Molose, A. M. Abu-Mahfouz, and N. Dladlu, “Comprehensive Review of SDN Controller Placement Strategies,” *IEEE Access*, vol. 8, pp. 170070–170092, 2020, doi: 10.1109/ACCESS.2020.3023974.

[2] S. Scott-Hayward, G. O’Callaghan, and S. Sezer, “Sdn Security: A Survey,” in *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, IEEE, Nov. 2013, pp. 1–7. doi: 10.1109/SDN4FNS.2013.6702553.

[3] M. B. Jimenez, D. Fernandez, J. E. Rivadeneira, L. Bellido, and A. Cardenas, “A Survey of the Main Security Issues and Solutions for the SDN Architecture,” *IEEE Access*, vol. 9, pp. 122016–122038, 2021, doi: 10.1109/ACCESS.2021.3109564.

[4] I. Rakine *et al.*, "Comprehensive Review of Intrusion Detection Techniques: ML and DL in Different Networks," *IEEE Access*, vol. 13, no. May, pp. 104345–104367, 2025, doi: 10.1109/ACCESS.2025.3579990.

[5] G. Logeswari, S. Bose, and T. Anitha, "An Intrusion Detection System for SDN Using Machine Learning," *Intell. Autom. Soft Comput.*, vol. 35, no. 1, pp. 867–880, 2023, doi: 10.32604/iasc.2023.026769.

[6] J. Xie *et al.*, "A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 1, pp. 393–430, 2019, doi: 10.1109/COMST.2018.2866942.

[7] M. Kaleem, M. A. Mushtaq, S. Rashid, and M. Saleemi, "A Comprehensive Review of Intrusion Detection Systems in IoT Landscape," in *Communications in Computer and Information Science*, vol. 2381 CCIS, no. 2, 2025, pp. 287–302. doi: 10.1007/978-3-031-82931-4_22.

[8] B. Shyryn, T. A. Ahanger, and A. Zhumadillayeva, "Enhancing software-defined network security with deep learning: a comprehensive review," *Int. J. Inf. Secur.*, vol. 25, no. 2, pp. 1–37, 2026, doi: 10.1007/s10207-026-01232-2.

[9] A. A. J. Al-Hchaimi *et al.*, "Enhancing Cybersecurity in Cyber-Physical Systems: an Explainable AI Approach for Intrusion Detection," in *2025 5th International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, IEEE, Aug. 2025, pp. 1–8. doi: 10.1109/eSmarTA66764.2025.11132126.

[10] A. Wani, R. S, and R. Khaliq, "SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT-SDL)," *CAAI Trans. Intell. Technol.*, vol. 6, no. 3, pp. 281–290, Sep. 2021, doi: 10.1049/cit2.12003.

[11] E. Tsogbaatar *et al.*, "SDN-Enabled IoT Anomaly Detection Using Ensemble Learning," in *IFIP Advances in Information and Communication Technology*, Springer International Publishing, 2020, pp. 268–280. doi: 10.1007/978-3-030-49186-4_23.

[12] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A Survey on Software-Defined Networking," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 1, pp. 27–51, 2015, doi: 10.1109/COMST.2014.2330903.

[13] S. Khan, A. Gani, A. W. Abdul Wahab, M. Guizani, and M. K. Khan, "Topology Discovery in Software Defined Networks: Threats, Taxonomy, and State-of-the-Art," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 1, pp. 303–324, 2017, [Online]. Available: https://ieeexplore.ieee.org/document/7534866/

[14] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Comput. Networks*, vol. 62, pp. 122–136, Apr. 2014, doi: 10.1016/j.bjp.2013.10.014.

[15] and A. R. A. R. A. Abduljabbar Ali, A. Muhammed, M. D. H. Abdullah, "Software-Defined Networks Topology Discovery Security and Drawbacks: A Survey of Attacks and Defenses," *IAENG Int. J. Comput. Sci.*, vol. 52, no. 7, pp. 2333–2357, 2025.

[16] A. Shaghaghi, M. A. Kaafar, R. Buyya, and S. Jha, "Software-Defined Network (SDN) data plane security: Issues, solutions, and future directions," *Handb. Comput. Networks Cyber Secur. Princ. Paradig.*, pp. 341–387, 2019, doi: 10.1007/978-3-030-22277-2_14.

[17] A. N. Alhaj and N. Dutta, *Analysis of Security Attacks in SDN Network: A Comprehensive Survey*, vol. 281, no. November. Springer Singapore, 2022. doi: 10.1007/978-981-16-4244-9_3.

[18] A. Abdou, P. C. van Oorschot, and T. Wan, "Comparative Analysis of Control Plane Security of SDN and Conventional Networks," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 3542–3559, 2018, doi: 10.1109/COMST.2018.2839348.

[19] S. Scott-Hayward, "Design and deployment of secure, robust, and resilient SDN controllers," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, IEEE, Apr. 2015, pp. 1–5. doi: 10.1109/NETSOFT.2015.7258233.

[20] L. Wang and Y. Liu, "A DDoS Attack Detection Method Based on Information Entropy and Deep Learning in SDN," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, IEEE, Jun. 2020, pp. 1084–1088. doi: 10.1109/ITNEC48623.2020.9085007.

[21] A. Patwardhan, D. Jayarama, N. Limaye, S. Vidhale, Z. Parekh, and K. Harfoush, "SDN Security: Information Disclosure and Flow Table Overflow Attacks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, IEEE, Dec. 2019, pp. 1–6. doi: 10.1109/GLOBECOM38437.2019.9014048.

[22] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A Survey of Security in Software Defined Networks," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 623–654, 2016, doi: 10.1109/COMST.2015.2453114.

[23] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in Software Defined Networks: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2317–2346, 2015, doi: 10.1109/COMST.2015.2474118.

[24] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 602–622, 2016, doi: 10.1109/COMST.2015.2487361.

[25] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Netw. Appl.*, vol. 12, no. 2, pp. 493–501, Mar. 2019, doi: 10.1007/s12083-017-0630-0.

[26] M. R. Ahmed, S. Islam, S. Shatabda, A. K. M. Muzahidul Islam, M. Towhidul, and I. Robin, "Intrusion Detection System in Software-Defined Networks Using Machine Learning and Deep Learning Techniques-A Comprehensive Survey," *Ieee*, no. Ml, pp. 1–47, 2023, [Online]. Available: https://doi.org/10.36227/techrxiv.17153213.v1

[27] U. S. Musa, S. Chakraborty, M. M. Abdullahi, and T. Maini, "A Review on Intrusion Detection System using Machine Learning Techniques," in *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, IEEE, Feb. 2021, pp. 541–549. doi: 10.1109/ICCCIS51004.2021.9397121.

[28] A. ARQANE, O. Boutkhoum, H. Boukhriss, and A. El Moutaouakkil, "Intrusion Detection System using Ensemble Learning Approaches: A Systematic Literature Review," *Int. J. Online Biomed. Eng.*, vol. 18, no. 13, pp. 160–175, Oct. 2022, doi: 10.3991/ijoe.v18i13.33519.

[29] N. S. Shaji, R. Muthalagu, and P. M. Pawar, "SD-IIDS: intelligent intrusion detection system for software-defined networks," *Multimed. Tools Appl.*, vol. 83, no. 4, pp. 11077–11109, Jan. 2024, doi: 10.1007/s11042-023-15725-y.

[30] D. Jankowski and M. Amanowicz, "Intrusion detection in software defined networks with self-organized maps," *J. Telecommun. Inf. Technol.*, vol. 2015, no. 4, pp. 3–9, 2015, doi: 10.26636/jtit.2015.4.977.

[31] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015, doi: 10.1109/JPROC.2014.2371999.

[32] A. H. Janabi, T. Kanakis, and M. Johnson, "Survey: Intrusion Detection System in Software-Defined Networking," *IEEE Access*, vol. 12, no. October, pp. 164097–164120, 2024, doi: 10.1109/ACCESS.2024.3493384.

[33] S. Dahiya, V. Siwach, and H. Sehrawat, "Review of AI Techniques in development of Network Intrusion Detection System in SDN Framework," in *2021 International Conference on Computational Performance Evaluation, ComPE 2021*, IEEE, 2021, pp. 168–174. doi: 10.1109/ComPE53109.2021.9752430.

[34] B. Laha, D. Basu, S. Biswas, P. Gupta, and B. Sadhukhan, "Intrusion Detection in IoT Systems Using Ensemble Machine Learning Techniques," in *2023 IEEE 4th Annual Flagship India Council International Subsections Conference: Computational Intelligence and Learning Systems, INDISCON 2023*, IEEE, Aug. 2023, pp. 1–7. doi: 10.1109/INDISCON58499.2023.10270505.

[35] N. W. Khan *et al.*, "A hybrid deep learning-based intrusion detection system for IoT networks," *Math. Biosci. Eng.*, vol. 20, no. 8, pp. 13491–13520, 2023, doi: 10.3934/mbe.2023602.

[36] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, Jan. 2013, doi: 10.1016/j.jnca.2012.09.004.

[37] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.

[38] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, p. 20, Dec. 2019, doi: 10.1186/s42400-019-0038-7.

[39] N. Hubballi and V. Suryanarayanan, "False alarm minimization techniques in signature-based intrusion detection systems: A survey," *Comput. Commun.*, vol. 49, pp. 1–17, Aug. 2014, doi: 10.1016/j.comcom.2014.04.012.

[40] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Comput. Networks*, vol. 51, no. 12, pp. 3448–3470, Aug. 2007, doi: 10.1016/j.comnet.2007.02.001.

[41] F. Rahim and A. P. D. S. Ali Abd Alradha Alsaidi, "A Comprehensive Review of Intrusion Detection Systems in IoT networks Using ML and DL Techniques," *AlKadhim J. Comput. Sci.*, vol. 3, no. 2, pp. 84–95, Jun. 2025, doi: 10.61710/kjcs.v3i2.111.

[42] H. Polat, O. Polat, and A. Cetin, "Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models," *Sustain.*, vol. 12, no. 3, 2020, doi: 10.3390/su12031035.

[43] S. Stolfo, W. Fan, W. Lee, and A. L. Prodromidis, "Cost-based Modeling and Evaluation for Data Mining With Application to Fraud and Intrusion Detection : Results from the JAM Project ∗," 2008 . Available: https://api.semanticscholar.org/CorpusID:16061051

[44] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, IEEE, Jul. 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.

[45] J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, *Cost-based modeling and evaluation for data mining with application to fraud and intrusion detection*. 2000.

[46] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, IEEE, Nov. 2015, pp. 1–6. doi: 10.1109/MilCIS.2015.7348942.

[47] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a Reliable Intrusion Detection Benchmark Dataset," *Softw. Netw.*, vol. 2017, no. 1, pp. 177–200, 2017, doi: 10.13052/jsn2445-9739.2017.009.

[48] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*, IEEE, Oct. 2019, pp. 1–8. doi: 10.1109/CCST.2019.8888419.

[49] A. Kaur, C. Rama Krishna, and N. V. Patil, "A comprehensive review on Software-Defined Networking (SDN) and DDoS attacks: Ecosystem, taxonomy, traffic engineering, challenges and research directions," *Comput. Sci. Rev.*, vol. 55, p. 100692, Feb. 2025, doi: 10.1016/j.cosrev.2024.100692.

[50] M. S. Elsayed, N.-A. A. Le-Khac, and A. D. Jurcut, "InSDN: A novel SDN intrusion dataset," *IEEE Access*, vol. 8, no. September, pp. 165263–165284, 2020, doi: 10.1109/ACCESS.2020.3022633.

[51] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Futur. Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019, doi: 10.1016/j.future.2019.05.041.

[52] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and Adna N Anwar, "TON-IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862.

[53] S. K. Dey and M. M. Rahman, "Effects of Machine Learning Approach in Flow-Based Anomaly Detection on Software-Defined Networking," *Symmetry (Basel).*, vol. 12, no. 1, p. 7, Dec. 2019, doi: 10.3390/sym12010007.

[54] A. O. Alzahrani and M. J. F. Alenazi, "Designing a Network Intrusion Detection System Based on Machine Learning for Software Defined Networks," *Futur. Internet*, vol. 13, no. 5, p. 111, Apr. 2021, doi: 10.3390/fi13050111.

[55] H. A. Alamri and V. Thayananthan, "Bandwidth Control Mechanism and Extreme Gradient Boosting Algorithm for Protecting Software-Defined Networks Against DDoS Attacks," *IEEE Access*, vol. 8, no. November, pp. 194269–194288, 2020, doi: 10.1109/ACCESS.2020.3033942.

[56] H. A. Alamri and V. Thayananthan, "Analysis of Machine Learning for Securing Software-Defined Networking," *Procedia Comput. Sci.*, vol. 194, pp. 229–236, 2021, doi: 10.1016/j.procs.2021.10.078.

[57] N. Omer, A. H. Samak, A. I. Taloba, and R. M. Abd El-Aziz, "A novel optimized probabilistic neural network approach for intrusion detection and categorization," *Alexandria Eng. J.*, vol. 72, pp. 351–361, Jun. 2023, doi: 10.1016/j.aej.2023.03.093.

[58] H. Alshahrani, A. Khan, M. Rizwan, M. S. Al Reshan, A. Sulaiman, and A. Shaikh, "Intrusion Detection Framework for Industrial Internet of Things Using Software Defined Network," *Sustain.*, vol. 15, no. 11, pp. 1–18, 2023, doi: 10.3390/su15119001.

[59] S. R. Rafin, M. S. S. Dip, and N. A. Arabi, "Enhancing SDN Security: A Balanced and Interpretable Machine Learning Framework for Intrusion Detection," in *2026 IEEE 15th International Conference on Communication Systems and Network Technologies (CSNT)*, IEEE, Apr. 2026, pp. 576–582. doi: 10.1109/CSNT69054.2026.11502483.

[60] M. M. Ahmed and H. Abdulkader, "An ensemble-based approach for effective distributed denial of service attack detection in software defined networking," *IAES Int. J. Artif. Intell.*, vol. 13, no. 2, p. 2019, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp2019-2026.

[61] S. A. More and A. V. Kachavimath, "SDN Intrusion Detection using Meta-Heuristic Optimization and K-Nearest Neighbors Classifier," *Procedia Comput. Sci.*, vol. 260, pp. 1137–1144, 2025, doi: 10.1016/j.procs.2025.03.299.

[62] M. T. Saleh and A. H. Hamad, "SICA: Generation and Deep Learning-Based Evaluation of a Novel Dataset for Intrusion Detection in SDN-IoT Environments," *J. Eur. des Systèmes Autom.*, vol. 59, no. 1, pp. 69–77, Jan. 2026, doi: 10.18280/jesa.590108.

[63] L. Boukraa, S. Essahraui, Y. Maleh, K. El Makkaoui, I. Ouahbi, and R. Esbai, "Machine Learning-Based Intrusion Detection Systems for Sdn: an Empirical Study Using Knime," *Edpacs*, vol. 69, no. 6, pp. 46–59, 2024, doi: 10.1080/07366981.2024.2360840.

[64] C. Kannan, R. Muthusamy, V. Srinivasan, V. Chidambaram, and K. Karunakaran, "Machine learning based detection of DDoS attacks in software defined network," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 32, no. 3, pp. 1503–1511, 2023, doi: 10.11591/ijeecs.v32.i3.pp1503-1511.

[65] A. H. Abdi, G. S. Member, S. Ahmed, and A. Tahir, "Security Control and Data Planes of SDN : A Comprehensive Review of Traditional , AI , and MTD Approaches to Security Solutions," vol. 12, no. April, pp. 69941–69980, 2024.

[66] G. Kumar and H. Alqahtani, "Machine Learning Techniques for Intrusion Detection Systems in SDN-Recent Advances, Challenges and Future Directions," *Comput. Model. Eng. Sci.*, vol. 134, no. 1, pp. 89–119, 2023, doi: 10.32604/cmes.2022.020724.

[67] M. Learning, *Machine learning Tom Micheal*, vol. 45, no. 13. 2017. [Online]. Available: https://books.google.ca/books?id=EoYBngEACAAJ&dq=mitchell+machine+learning+1997&hl=en&sa=X&ved=0ahUKEwiomdqfj8TkAhWGslkKHRCbAtoQ6AEIKjAA

[68] S. M. Doğan, A. Koçak, and M. Alkan, "Detection and mitigation of cyber-attacks in software defined networks using machine learning/deep learning: a systematic literature review, research challenges and future directions," *Int. J. Inf. Secur.*, vol. 24, no. 5, p. 209, Oct. 2025, doi: 10.1007/s10207-025-01114-z.

[69] F. A. Vadhil, M. L. Salihi, and M. F. Nanne, "Machine learning-based intrusion detection system for detecting web attacks," *IAES Int. J. Artif. Intell.*, vol. 13, no. 1, pp. 711–721, 2024, doi: 10.11591/ijai.v13.i1.pp711-721.

[70] A. A. Abu-Shareha, M. M. Abualhaj, A. Hussein, O. Almomani, A. Amer, and A. Achuthan, "Supervised machine learning intrusion detection review and multi-criteria evaluation," *Sci. Rep.*, Mar. 2026, doi: 10.1038/s41598-026-44773-1.

[71] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *J. Netw. Comput. Appl.*, vol. 60, pp. 19–31, Jan. 2016, doi: 10.1016/j.jnca.2015.11.016.

[72] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, no. 1–2, pp. 18–28, Feb. 2009, doi: 10.1016/j.cose.2008.08.003.

[73] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *2010 IEEE Symposium on Security and Privacy*, IEEE, 2010, pp. 305–316. doi: 10.1109/SP.2010.25.

[74] S. A. Alomari *et al.*, "The Evolution of Machine Learning: From Traditional Algorithms to Deep Learning Paradigms," in *Mastering the Minds of Machines*, Boca Raton: CRC Press, 2025, pp. 9–15. doi: 10.1201/9781003516385-2.

[75] X. Zhu and A. B. Goldberg, "Introduction to Semi-Supervised Learning," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 3, no. 1, pp. 1–130, Jan. 2009, doi: 10.2200/S00196ED1V01Y200906AIM006.

[76] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Inf. Sci. (Ny).*, vol. 378, pp. 484–497, Feb. 2017, doi: 10.1016/j.ins.2016.04.019.

[77] T.-P. Nguyen, J. Cho, and D. Kim, "Semi-supervised intrusion detection system for in-vehicle networks based on variational autoencoder and adversarial reinforcement learning," *Knowledge-Based Syst.*, vol. 304, p. 112563, Nov. 2024, doi: 10.1016/j.knosys.2024.112563.

[78] C. Chen, Y. Gong, and Y. Tian, "Semi-supervised learning methods for network intrusion detection," *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, pp. 2603–2608, 2008, doi: 10.1109/ICSMC.2008.4811688.

[79] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, 2020, doi: 10.1007/s10994-019-05855-6.

[80] Q. Shambour, M. Al-Zyoud, and O. Almomani, "Quantum-Inspired Hybrid Metaheuristic Feature Selection with SHAP for Optimized and Explainable Spam Detection," *Symmetry (Basel).*, vol. 17, no. 10, pp. 1–34, 2025, doi: 10.3390/sym17101716.

[81] A. H. K. Janabi, T. Kanakis, and M. Johnson, "A Survey of Intrusion Detection Systems based Machine Learning Approaches Applied to Software-Defined Networks (SDN): Research Issues and Challenges," 2023, doi: 10.20944/preprints202312.1449.v1.

[82] R. Fu, "Design and Implementation of Network Intrusion Detection System based on Machine Learning," in *2025 International Conference on Intelligent Systems and Computational Networks (ICISCN)*, IEEE, Jan. 2025, pp. 1–6. doi: 10.1109/ICISCN64258.2025.10934502.

[83] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-Based Syst.*, vol. 78, pp. 13–21, Apr. 2015, doi: 10.1016/j.knosys.2015.01.009.

[84] M. Shehab, A. Smerat, and L. Abualigah, "Reinforcement Learning-based Optimization Algorithms: A Survey," in *Mastering the Minds of Machines*, Boca Raton: CRC Press, 2025, pp. 172–177. doi: 10.1201/9781003516385-22.

[85] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 424–430, Jan. 2012, doi: 10.1016/j.eswa.2011.07.032.

[86] T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, 1967, doi: 10.1109/TIT.1967.1053964.

[87] K. Noor, A. L. Imoize, C.-T. Li, and C.-Y. Weng, "A Review of Machine Learning and Transfer Learning Strategies for Intrusion Detection Systems in 5G and Beyond," *Mathematics*, vol. 13, no. 7, p. 1088, Mar. 2025, doi: 10.3390/math13071088.

[88] S. A. Alomari *et al.*, "Supervised Learning: Teaching Machines with Labeled Data," in *Mastering the Minds of Machines*, Boca Raton: CRC Press, 2025, pp. 26–33. doi: 10.1201/9781003516385-4.

[89] Simon Haykin, "Neural Networks - A Comprehensive Foundation - Simon Haykin.pdf," 2005.

[90] K. C. Khor, C. Y. Ting, and S. P. Amnuaisuk, "A feature selection approach for network intrusion detection," *Proc. - 2009 Int. Conf. Inf. Manag. Eng. ICIME 2009*, vol. 3, no. December, pp. 133–137, 2009, doi: 10.1109/ICIME.2009.68.

[91] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Comput. Secur.*, vol. 70, pp. 255–277, Sep. 2017, doi: 10.1016/j.cose.2017.06.005.

[92] A. M. Oyelakin and J. R. G, "A Survey of Feature Extraction and Feature Selection Techniques used in Machine Learning-Based Botnet Detection Schemes," *VAWKUM Trans. Comput. Sci.*, vol. 9, no. 1, pp. 01–07, 2021, doi: 10.21015/vtcs.v9i1.604.

[93] J. Miao and L. Niu, "A Survey on Feature Selection," *Procedia Comput. Sci.*, vol. 91, no. Itqm, pp. 919–926, 2016, doi: 10.1016/j.procs.2016.07.111.

[94] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowl. Inf. Syst.*, vol. 34, no. 3, pp. 483–519, 2013, doi: 10.1007/s10115-012-0487-8.

[95] I. Guyon, "An Introduction to Variable and Feature Selection Isabelle," *J. ofMachine Learn. Res. 3 1157-1182*, vol. 1, pp. 1–26, 2003, doi: 10.1162/153244303322753616.

[96] R. Kohavi, G. H. John, H. Rd, and S. Jose, "Wrappers for Feature Subset Selection 2 Feature Subset Selection," *Focus (Madison).*, pp. 1–43, 2011.

[97] A. Wiliński and S. Osowski, "Gene selection for cancer classification," *COMPEL - Int. J. Comput. Math. Electr. Electron. Eng.*, vol. 28, no. 1, pp. 231–241, 2009, doi: 10.1108/03321640910919020.

[98] M. B. Imani, M. R. Keyvanpour, and R. Azmi, "A novel embedded feature selection method: A comparative study in the application of text categorization," *Appl. Artif. Intell.*, vol. 27, no. 5, pp. 408–427, 2013, doi: 10.1080/08839514.2013.774211.

[99] Z. Jin, J. Shang, Q. Zhu, C. Ling, W. Xie, and B. Qiang, "RFRSF: Employee Turnover Prediction Based on Random Forests and Survival Analysis," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12343 LNCS, pp. 503–515, 2020, doi: 10.1007/978-3-030-62008-0_35.

[100] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," *Data Classif. Algorithms Appl.*, pp. 37–64, 2014, doi: 10.1201/b17320.

[101] V. Deepa, K. M. Sudar, and P. Deepalakshmi, "Design of Ensemble Learning Methods for DDoS Detection in SDN Environment," *Proc. - Int. Conf. Vis. Towar. Emerg. Trends Commun. Networking, ViTECoN 2019*, pp. 17–22, 2019, doi: 10.1109/ViTECoN.2019.8899682.

[102] J. McNeill, *Ensemble Methods Foundations and Algorithms*, vol. 13, no. 2. 1984.

[103] A. Saleem and H. Beitollahi, "An Ensemble-based Machine Learning Framework for Advanced Distributed Denial of Service Attack Detection in Software Defined Networks," *UHD J. Sci. Technol.*, vol. 9, no. 2, pp. 184–197, Oct. 2025, doi: 10.21928/uhdjst.v9n2y2025.pp184-197.

[104] T. G. Dietterich, "Ensemble Methods in Machine Learning," 2000, pp. 1–15. doi: 10.1007/3-540-45014-9_1.

[105] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Min. Knowl. Discov.*, vol. 8, no. 4, Jul. 2018, doi: 10.1002/widm.1249.

[106] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review," *Eng. Appl. Artif. Intell.*, vol. 115, p. 105151, Oct. 2022, doi: 10.1016/j.engappai.2022.105151.

[107] S. S. Systems, "Ensemble Learning Framework for DDoS Detection in," 2024.

[108] T. Ravichandran, K. Gavahi, K. Ponnambalam, V. Burtea, and S. J. Mousavi, "Ensemble-based machine learning approach for improved leak detection in water mains," *J. Hydroinformatics*, vol. 23, no. 2, pp. 307–323, Mar. 2021, doi: 10.2166/hydro.2021.093.

[109] A. Hirsi, L. Audah, A. Salh, M. A. Alhartomi, and S. Ahmed, "Enhancing SDN security using ensemble-based machine learning approach for DDoS attack detection Enhancing SDN security using ensemble-based machine learning approach for DDoS attack detection," no. March, pp. 1073–1085, 2025, doi: 10.11591/ijeecs.v38.i2.pp1073-1085.

[110] Z. Alomari, H. Sadineni, M. B. Taha, and Z. Li, "Hybrid Ensemble Learning Framework for Real-Time DDoS Detection and Mitigation in SDN Environments," in *2025 3rd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*, IEEE, Sep. 2025, pp. 1–8. doi: 10.1109/AIBThings66987.2025.11296148.

[111] R. A. Elsayed, R. A. Hamada, M. I. Abdalla, and S. A. Elsaid, "Securing IoT and SDN systems using deep-learning based automatic intrusion detection," *Ain Shams Eng. J.*, vol. 14, no. 10, p. 102211, Oct. 2023, doi: 10.1016/j.asej.2023.102211.

[112] J. Wang and L. Wang, "SDN-Defend: A Lightweight Online Attack Detection and Mitigation System for DDoS Attacks in SDN," *Sensors*, vol. 22, no. 21, p. 8287, Oct. 2022, doi: 10.3390/s22218287.

[113] A. S. Jose, L. R. Nair, and V. Paul, "Desinging Intrusion Detection System in Software Defined Networks Using Hybrid Gwo-Ae-Rf Model," *Indian J. Comput. Sci. Eng.*, vol. 13, no. 6, pp. 1951–1966, Dec. 2022, doi: 10.21817/indjcse/2022/v13i6/221306129.

[114] A. Abubakar and B. Pranggono, "Machine learning based intrusion detection system for software defined networks," *Proc. - 2017 7th Int. Conf. Emerg. Secur. Technol. EST 2017*, vol. 9, no. 09, pp. 138–143, 2017, doi: 10.1109/EST.2017.8090413.