Evaluating Machine Learning Models for DDoS Detection in SDNs

Özgür Tonkal and Jeremia Anthony Mgungile*

Department of Software Engineering, Samsun University, Samsun, Turkey
Email: ozgur.tonkal@samsun.edu.tr (Ö.T); jayantony01@gmail.com (J.A.M.)
*Corresponding author

***Abstract*—With the increased usage of internet-enabled devices in the contemporary networked world, vulnerability to security attacks has significantly widened. Distributed Denial of Service (DDoS) attacks are particularly critical, as they impact service availability and diminish valuable computing and internet resources. The study aims to enhance fine-grained DDoS identification by evaluating and comparing the performance of machine learning algorithms over Software-Defined Networks (SDNs), to improve accuracy and reduce false positives. We assessed machine learning algorithms, namely Extreme Gradient Boosting (XGBoost), Random Forest (RF), Naive Bayes (NB), and the Hidden Markov Model (HMM), against the labelled DDoS datasets, the LR-HR DDoS 2024 and InSDN. We employed a vaccine-based binary wolf grey optimization feature selection approach to rank data attributes by their respective levels of importance. The measures employed for the purpose included time delay, accuracy, false positive rate, and true positive rate. From the models compared, XGBoost showed the most effective detection, with the best accuracy and reduced false positive instances, especially when employed with an enhanced vaccine-based feature selection. Experimental results confirm that XGBoost, especially when combined with vaccine-based Binary Grey Wolf Optimization (BGWO) feature selection, gives a highly effective solution for detecting DDoS attacks in a Software Defined Network environment.**

***Keywords*—distributed denial of service, software defined networks, cybersecurity, machine learning**

I. Introduction

Contemporary networks employ a new architecture called Software Defined Networking (SDN), where the data plane is separated from the control plane. The former consists of devices such as routers and switches, whereas the latter comprises controllers that manage resources in the data plane. This unique architecture leads to great flexibility. However, this flexibility comes at the cost of new cybersecurity risks, including Distributed Denial-of-Service (DDoS) attacks that target SDN Controllers [1]. DDoS attacks overwhelm a specific server, service, network, or endpoint by sending massive amounts of traffic or requests from multiple machines and botnets during a short time. The consequence can be a deterioration in service performance or even the shutdown of the entire service. These attacks are particularly lethal for SDNs, as the controller is the single point of failure [2].

It has been estimated that DDoS attacks have grown to an unprecedented level, with vast growth in volume, intensity, frequency, and new attack vectors. Being a pivotal component of cloud computing architecture and SDN, at present, controllers may also be victims of attack. Among the various security threats, attacks targeting the controller are particularly critical [3]. Since the controller serves as the central intelligence of the network, an attacker who compromises it can manipulate or disrupt all network traffic, posing a severe risk to the entire SDN environment [4]. However, DDoS attacks not only inflict damage upon networks but also, they also tamper with the authentic application operations, ultimately decreasing the authenticity of the packet.

Therefore, the focus of this article is to evaluate machine learning models for DDoS detection in Software-Defined Networks (SDNs). The goal is to explore the usage of the detection models under varying conditions in an SDN and show the outcomes of employing them in detecting DDoS. Several research objectives need to be accomplished to fulfill this goal. However, the objective of this work is to use machine learning for efficient DDoS detection in an SDN.

Software-Defined Networking (SDN) is a network management and communication paradigm that decouples control and data planes, introducing the logic of network data management. The main idea of SDN is to provide a centralized control for distributed data and to construct dynamic, changeable, and flexible activities. Software-defined networking introduces a programmable global view of a network that helps to visualize traffic patterns, facilitate security, and quickly respond to attacks and system breaches [5].

A typical SDN architecture comprises three planes: application, control, and data plane.

The application plane, situated at the top of SDN architecture, includes traffic engineering, resource allocation, security enforcement, and other management functions.

Manuscript received August 3, 2025; revised November 7, 2025; accepted December 30, 2025; published March 26, 2026.

doi: 10.12720/jait.17.3.596-610

The control plane runs on a central server that accepts policies from applications, translates them into low-level commands, and imposes corresponding actions on the data plane. The control introduces global visibility and network-wide intelligence. The control plane is positioned between the application plane and data plane and serves as the heart of SDN.

The data plane consists of simple network devices such as switches and routers whose essential tasks are to forward packets and carry out actions based on commands received from the control plane [6]. Fig. 1 illustrates the layered structure of SDN architecture.

The communication between the layers happens via the Northbound Application Programming Interface (API) interface, which establishes communication between the Application plane and the Control plane, while the Southbound interface serves between the Control plane and the data plane.

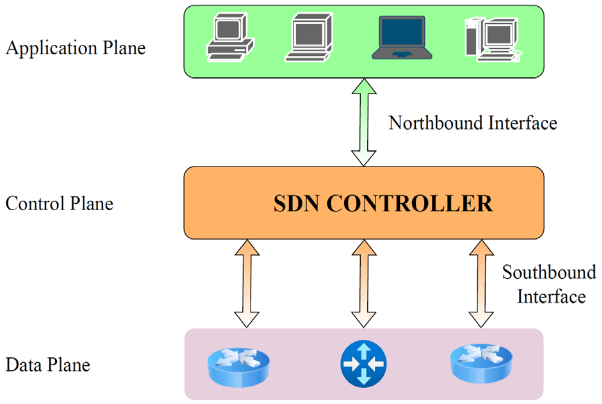


Fig. 1. SDN architecture.

II. Literature Review

The study examined the state of the art related to DDoS attacks in SDNs and machine learning applications in DDoS attacks in SDNs. DDoS attacks degrade network quality of service by swamping the local network server, or Internet link, with a deluge of bogus traffic from multiple compromised sources [7].

Furthermore, the security of SDNs is improved by proper detection and safeguarding against DDoS attacks. A variety of techniques used to cap, detect, and obstruct the DDoS flooding on different layers of networking have emerged in recent years [8]. However, most of these proposed approaches can be sidestepped in SDN due to the lack of resources for complete exploration. Due to limited resources and increasing traffic complexity, existing DDoS prevention approaches often introduce additional overhead and may negatively affect network performance [9].

In comparison to known methods, there is a need for a more resilient technique for DDoS detection. A better option is deep or machine learning in this context, as it is not necessary to have detailed knowledge of the network pattern of normal or adversary traffic [10]. Traditional traffic prediction based on analytical probabilities does not capture fuzzy behavior in traffic; rather, it utilizes heuristics that make the produced model overly fundamental, rendering the model ineffective against DDoS attacks, since the attacks do not conserve the heuristic spectrum enforced.

Software Defined Networks have received significant attention due to their flexibility and centralized control architecture. Although SDNs provide several security benefits, they remain vulnerable to DDoS attacks targeting both the data and control planes. As a result, recent research has focused on the development of intelligent DDoS detection and mitigation mechanisms capable of identifying malicious traffic patterns and improving network resilience. Existing studies primarily investigate supervised learning approaches for detecting known DDoS attack patterns in SDN environments using labeled datasets.

The SDN's unique structure provides attackers with new threats that can prompt DDoS attacks. Layer 2 switches can be programmed to use the OpenFlow protocol [11]. The controller is responsible for programming the switches [12]. It can take initial measurements and redirect suspicious traffic to the controller. Existing SDN-based DDoS detection studies employ a wide range of machine learning and deep learning architectures, including decision tree-based classifiers, ensemble methods, and hybrid deep models such as Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM). However, these studies differ considerably in terms of dataset selection, architectural complexity, and evaluation criteria, making direct comparison difficult. To systematically highlight these differences and position the proposed approach within the current state of the art, Table I summarizes representative SDN-based DDoS detection studies, highlighting the architectures used, datasets adopted, and key methodological characteristics.

Recent studies in Software Defined Networking (SDN) have indeed focused on the real-time and adaptive Distributed Denial of Service detection and mitigation mechanism. Abdallah *et al.* [13] proposed a hybrid CNN-LSTM model for anomaly detection in SDN environments, demonstrating significantly improved accuracy by leveraging both spatial and temporal correlations in network traffic. Likewise, Rajan and Aravindhar [14] applied a CNN-LSTM hybrid for DDoS attack detection and mitigation, attaining high accuracy but at the cost of increased computational complexity. These approaches highlight a growing emphasis on adaptivity and context-aware intelligence in SDN security.

TABLE I. SDN-BASED DDOS DETECTION STUDIES

Study	Architecture(s) used	Dataset(s)	Task	SDN plane focus	Notes
Bajenaid *et al.* [15]	Comparative analysis (survey-style)	InSDN (discussed)	Intrusion Detection System (IDS)	Control	Survey; no implementation
Setitra *et al.* [16]	Optimized Convolutional Neural Network-Multilayer Perceptron (CNN-MLP) Deep Learning (DL)	SDN DDoS dataset	DDoS	Control	High accuracy but high computational cost; latency is not emphasized
Mudgal *et al.* [17]	CatBoost + flow-table features	SDN telemetry (LOFT)	Data-plane attacks	Data plane	Specialized in flow-table overflow attacks
Omer *et al.* [18]	Long Short-Term Memory (LSTM)	Low-Rate/High-Rate (LR-HR) DDoS-2024	Low-rate Denial of Service (LDoS)	Control	Single DL model; no benchmarking or mitigation
Elshewey *et al.* [19]	MLP, CNN, LSTM, Gated Recurrent Unit (GRU), CNN-GRU	SDN traffic dataset	DDoS	Control	Strong DL benchmarks; no latency analysis
Ibrahim *et al.* [20]	Deep Neural Network (DNN), CNN, LSTM	SDN security dataset	IDS/DDoS	Control	Focuses on DL only; no feature optimization
Mehmood *et al.* [3]	Optimizer-equipped CNN-MLP	InSDN	DDoS	Control	DL heavy; real-time feasibility unclear
Khalid and Aldabagh [21]	Hierarchical multi-Class	InSDN	IDS	Control	Review: No implementation
Proposed approach	RF, XGBoost, NB, HMM+Vaccine-BGWO feature selection; benchmarked vs Decision Tree (DT), Logistic Regression, Linear Support Vector Machine (SVM)	LR-HR DDoS 2024, InSDN	DDoS	Data+Control	Optimized for near real-time (ms-level) detection; includes statistical significance testing, latency/throughput analysis.

Besides, these changes are easy to deploy in SDN. Detection techniques require the cooperation of network devices to gather the necessary information to effectively detect an attack. This action leads to a high level of false negatives since only some points can provide all the information needed for detection. It is difficult to guess alerts when the attacker's previous behavior is not controlled. This is because the detectors are determined, and it is difficult to anticipate actions generated by the attacker [22].

Notwithstanding these advances, the existing solutions often have scalability and interpretability limitations. Deep models, while accurate, are resource-intensive and impose high processing overhead on controllers.

A. DDoS Attacks in SDNs

An increasing threat to networks, particularly to Software-Defined Networks (SDNs), is Distributed Denial of Service (DDoS) attacks. The DDoS attack exhausts a network's resources, resulting in unavailable or degraded services. Attacking an SDN with DDoS traffic can cause the entire network's failure since it is controlled by a logically centralized controller [23]. Attacking a part of an SDN using DDoS attacks, such as switches, can reduce network quality and security. Fig. 2 illustrates possible attack points in the SDN architecture. Conventional networks, designed in a distributed and flat manner, lack a general controlling entity that can be affected by a direct DDoS attack.

However, an SDN has a logically centralized control plane that can be disabled by DDoS attacks. Moreover, DDoS attackers can directly or indirectly affect the SDN performance, and ultimately, the SDN is used as a proof of concept and has the potential to be used by more sophisticated attackers. In fact, new technologies like SDNs involve difficulties at the network level; for example, DDoS attacks present challenges in deploying security solutions from a low level to a programmable network level in the northbound interface [24].

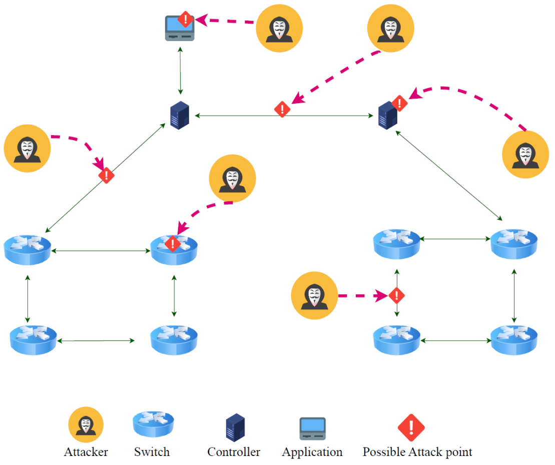


Fig. 2. Possible attack points in SDN architecture.

Moreover, some recent works design only DDoS-resistant network architecture or detection mechanisms in SDN, but they fail to discuss the adaptation or mitigation methods from a security perspective of an SDN. Existing DDoS-resistant mechanisms are mostly static and are unable to detect the daily and ever-changing anomalies [9]. The requirement of the current SDN is to have an adaptive defense with a real-time network monitoring mechanism.

The data layer of an SDN architecture is an important point of entry for attackers, since a compromised or malicious host can send enough traffic to an SDN switch to cause it to fail by either consuming all of the SDN switch's resources or by exhausting its flow tables through User Datagram Protocol (UDP) or other protocol floods [25] This creates two problems for the controller: first, it will not be able to transmit data to the switch; and second, it will not be able to verify whether the traffic going to the switch is legitimate. Therefore, SDN switch data statistics are continuously collected using OpenFlow counters versus OpenFlow telemetry. The data collected will be aggregated at the controller with other data at a higher level into flow features to classify the traffic, after it has been sent through OpenFlow counters. By correlating data plane indicators with control plane events, the framework improves visibility into attacks that primarily manifest at the switch level, such as traffic flooding and flow table saturation.

Recently, with the requirement of sophisticated service availability from users and businesses, numerous techniques, protocols, and architectures have been innovatively developed to provide the services more efficiently [26].

The DDoS attacks may target the SDN planes (Data and Control). Attacks on each plane exploit distinct vulnerabilities and have different operational impacts on network performance and stability. Table II summarizes and contrasts data plane and control plane DDoS attacks in SDN environments.

TABLE II. COMPARISON OF DATA PLANE AND CONTROL PLANE DDOS ATTACKS IN SDN

Aspect	Data Plane DDoS Attacks	Control Plane DDoS Attacks
Targeted SDN Component	SDN switches and forwarding devices	SDN controller and control channel
Primary Objective	Exhaust switch bandwidth, flow tables, or packet-processing capacity	Overload the controller CPU, memory, or control logic
Attack Mechanism	Massive packet flooding, spoofed flows, and table-miss exploitation	Flooding Packet-In messages, fake flow requests
Traffic Characteristics	High-volume data traffic, often spoofed or randomized	High rate of control messages with low data payload
Impact on Network	Packet loss, increased latency, switch malfunction	Controller bottleneck, delayed flow setup, network-wide degradation
Attack Scope	Localized to specific switches or links	Global impact affecting the entire SDN domain
Detection Location	Edge switches or data-plane monitors	SDN controller (control layer)
Detection Complexity	Moderate; relies on traffic volume and flow statistics	High; requires analysis of control traffic patterns
Real-Time Requirement	Very high (packet-level response needed)	High (controller responsiveness critical)
Representative Examples	Traffic flooding, flow table overflow	Packet-In flooding, controller saturation

B. Overview of Machine Learning in DDoS Detection

Machine learning techniques are increasingly being considered as powerful tools to enhance the detection of DDoS attacks [1]. The spectrum of machine learning-based models for DDoS detection ranges widely and includes decision trees, support vector machines, deep belief networks, clustering, naive Bayes, hidden Markov models, and various computing classifiers with easy state transitions like boosting trees and random forests [27]. These models can be employed in either supervised or unsupervised learning. Building a model for DDoS detection requires identifying essential features of DDoS attack patterns to support proper classification. These can include various combinations of volume, rate, and other descriptive statistics for network traffic data, as well as packet payload details, port numbers, and byte and packet rates. That is, detecting DDoS attacks involves training models with numerous reliable traffic-related characteristics to adequately capture the distinct behavior of DDoS attack traffic [27].

The use of machine learning-based models has become an increasingly popular research trend in the study and application of DDoS detection methods. These models can adapt to different kinds of DDoS behavior and network environments, as well as provide high classification accuracy and low latency. They have shown their ability to effectively distinguish between attack and benign traffic [27], especially for classifying dense and sparse DDoS attackers. In a dynamic network environment, using adaptive learning reduces the need for ongoing parameter tuning, which is a well-known characteristic of DDoS protection models. This is where the need for in-depth research into classification characteristics becomes apparent.

1) XGBoost algorithms

XGBoost has received great acceptance due to its wide use and robust implementation. It represents an optimized distributed gradient boosting algorithm. XGBoost natively provides a parallel tree boosting algorithm based on gradient descent [27]. It is highly scalable in terms of standardization, time efficiency, and memory utilization, thanks to its sparse-aware algorithm [28]. Recently, the development of other improved versions has been presented. In the context of our study, where a multi-level approach to DDoS detection is considered, we found that XGBoost achieves the best rates in the Adaptive Neuro-Fuzzy Inference System (ANFIS) context, formed by a set of trained experts designed based on the complexity of each specific DDoS attack.

2) Random forest

Random Forest is a popular and robust ensemble learning method that is applicable for classification [8]. It is frequently chosen by organizations and data scientists when classifying network or flow problems, such as DDoS detection [29]. This algorithm can be described as a robust version of bagging and classification via improving on the bootstrap theory. The Random Forest model technology uses several decision trees as a basis for its purpose of creating a “forest” [30]. This can be very accommodating when the size of the data is enormous, with several dimensions. When a Random Forest is trained, some trees are tested on input data. The output produced is determined by using a technique such as “majority vote” to apply statistical analysis within decision trees.

3) Naïve Bayes

Naive Bayes is a simple but efficient classification algorithm based on the Bayesian rule [12]. According to probability theory, the Bayesian rule is shown in Eq. (1).

$$P(ci|X) = \frac{P(ci) \times P(X|c_i)}{P(X)} \quad (1)$$

where $P(c_i|X)$: posterior probability of class c_i given the data X, $P(c_i)$: Prior probability of class c_i, $P(X|c_i)$ refers to the probability of data X given class c_i, $P(X)$: total probability of data X.

Then, if $P(X)$ does not change between classes, which means they are constants, the formula can be simplified as shown in Eq. (2):

$$P(c_i|X) \propto P(c_i) \times P(X|c_i) \quad (2)$$

Naive Bayes assumes that features are conditionally independent given the class, which means the Naive Bayes classifier is a probabilistic classifier using the Bayes theorem with an assumption of independence among the events [31].

4) Hidden Markov models

Hidden Markov Models (HMMs) have also shown their significance in evaluating DDoS threats in an SDN environment due to their ability for temporal modeling. The general trends and behavior of data can be modeled well using suitable models. However, HMMs are more suited to the problem domain due to their probabilistic representation. A Hidden Markov Model is a trellis diagram representing an observer who is recording evidence regarding a source generating a sequence of symbols. At a specific time, the source is in one of a finite number of system states emitting the symbol that identifies the state.

This is the reason why HMM is interesting when considering a cybersecurity environment where a change of state happens over time, and these states are not seen directly. This fact has contributed to the popularity of HMM in network intrusion detection systems.

III. Materials and Methods

This study conducts a comprehensive evaluation of a set of machine learning models, including Random Forest, Naive Bayes as traditional and temporal-based detection models, the Hidden Markov Model (HMM), as well as XGBoost classifiers comparable to each other in terms of various metrics: accuracy, precision, recall score, and F1-Score. These models are selected based on qualitative and quantitative criteria to reduce false positives and false negatives during their evaluations. A methodology for structural testing that is suitable for both models and data is designed to evaluate the selected models. The tests are then executed, and the results are analyzed to identify the machine learning model that gives the best performance with fewer resources [1].

A methodology was carefully designed in multiple stages, starting from data preprocessing, where the datasets are split into training and testing. Then the models are independently trained while fine-tuning their hyperparameters. The testing set is used to evaluate the performance of the trained models on unseen data.

To validate the robustness of the proposed vaccine-based (BGWO) DDoS detection model, different experiments were carried out that involved comparing the model evaluated without feature selection and the one with vaccine-based BGWO feature selection. Fig. 3 displays how the events are staged in this study.

To analyze the results and summarize them, multiple visualizations were used, and a comprehensive comparison of the models on each evaluation metric was conducted.

The datasets LR-HR DDoS 2024 and InSDN were used. The dataset included normal traffic (Benign) and malicious attacks, including UDP flooding, Internet Control Message Protocol (ICMP) flooding, Synchronize (SYN) flooding, Acknowledgement (ACK) flooding, TCP flooding, and other attacks classified as slow attacks, and they are labeled 0 and 1, respectively.

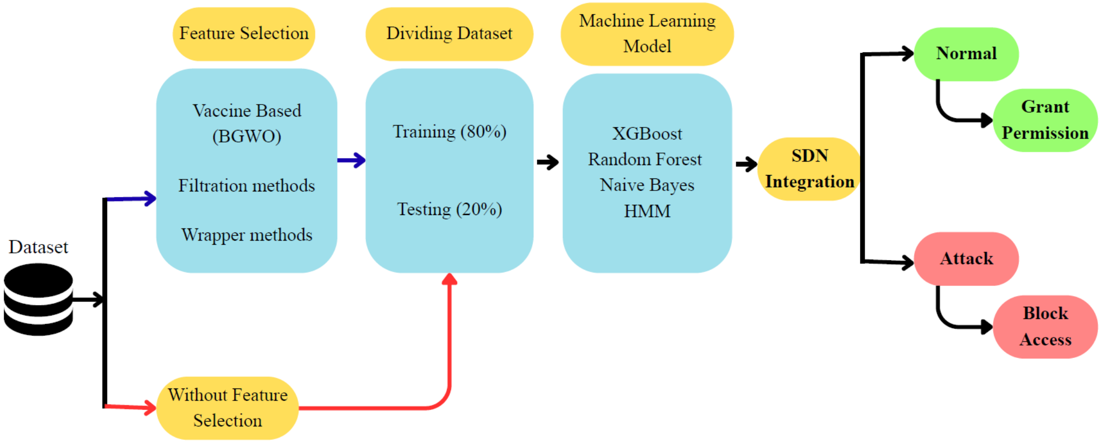


Fig. 3. Flow of events.

C. Dataset

LR-HR DDoS dataset 2024 [32] and InSDN [33] are publicly available datasets for SDN-Based Networks, and they were used as a baseline for our experimentation. The dataset is simulated using a DDoS attack-defense simulator. The dataset encompasses wire and wireless topologies, combining normal and malicious traffic. LR-

HR DDoS dataset comprises 145,614 packets collected over the simulation, with 78,733 normal and 61,881 malicious packets [32]. While the total number of normal and attack traffic instances in the InSDN dataset is 343,939 for, with a total of 68,424 normal data and 275,515 attack traffic instances. Then, data preprocessing was done through exploratory data analysis, data transformation, and data selection. Finally, classical machine learning classifiers, including Random Forest, Naïve Bayes, HMM, and XGBoost classifiers, were selected and implemented to find the best classifier against DDoS attacks. The model obtained was implemented in a simulated environment for validation in the SDN environment.

D. Vaccine-Based Feature Selection

This section details the vaccine-based feature selection for DDoS detection using the BGWO process. The high-level idea was borrowed from the concept of vaccination in the human immune system, whereby promising feature subsets act as "vaccines" to bias the search toward beneficial regions of the feature space. This helps reduce redundant features, speeds up convergence, and retains the discriminative attributes for accuracy in the classifier.

Operational steps that were applied include; Initialization, Vaccine repository, Vaccine injection, Grey Wolf Update (BGWO), local refinement, followed by termination.

Initialization; Initial population was generated, where each wolf encodes a binary feature selection vector (1 selected and 0 discarded) Each wolf was then evaluated using classification performance measures.

Vaccine repository: Acted as a small repository *V* of top-k vaccine vectors representing historically best feature subsets (ranked by weight). The repository is updated every. T_v Iterations by inserting new weight and evicting the weakest vaccine if capacity is exceeded.

Vaccine injection: At regular intervals, the weighted wolf receives a controlled injection.

Grey Wolf update (BGWO core): Standard BGWO position update rules were applied using α, β, and δ leaders to guide search exploitation scheduling. The vaccine injections occur either before or after the BGWO update, but we observed better stability when injections are applied before the BGWO move.

$$X(t+1) = (1-\alpha) \times X_{GWO}(t+1) + \alpha \times V,\ V \in \boldsymbol{\mathcal{V}},\ \alpha \in [0, 1] \tag{3}$$

$$X_i(t+1) = \begin{cases} V_i, & \text{if } m_i = 1 \\ round\left(\sigma\left(X_{GWO}, i(t+1)\right)\right), & \text{otherwise,} \end{cases} \text{ for } i = 1, \dots, N$$

Every T_v iterations, the vaccine repository $\boldsymbol{\mathcal{V}}$ is updated by inserting the top-performing wolves and removing the lowest-performing ones, maintaining the top k vaccine patterns. Full algorithm for the vaccine BGWO is as follows.

Algorithm 1: Algorithm for the vaccine BGWO

```
Input:
  $D$: Training dataset
  $F$: Total number of features
  $N$: Number of wolves
  $T$: Maximum iterations
  $\alpha$: Feature penalty coefficient

Output:
  $F_{best}$: Optimal feature subset

Initialize wolf population $W_i \in \{0,1\}$^F randomly
Evaluate fitness of each wolf using:
  Fitness = Accuracy − $\alpha$ × (Number of selected features)
Identify $\alpha$-wolf, $\beta$-wolf, $\delta$-wolf

for $t$ = 1 to $T$ do
  for each wolf $W_i$ do
    Update wolf position using GWO equations
    Apply the sigmoid function to convert to binary
    Apply vaccine-based mutation:
      Randomly flip bits with adaptive mutation rate
  end for
  Update $\alpha$, $\beta$, $\delta$ wolves
end for

Return feature subset from α-wolf
```

Eq. (4) confines how selection balances detection was performed.

$$S = \lambda \times \text{ErrorRate}_{\text{val}}(S) + \frac{(1-\lambda) \times |S|}{N} \tag{4}$$

where: *S* is the Selected feature, *N* is the total number of features, λ weight classification, α is the vaccine injection fraction, *X* denotes a candidate solution.

The Binary Grey Wolf Optimization (BGWO) algorithm was chosen for feature selection due to its converging behavior, which is actually the best among all other nature-inspired optimization techniques, and to its adaptability in the exploration/exploitation balance that other methods cannot match. BGWO, in contrast to Genetic Algorithm (GA) or Particle Swarm Optimisation (PSO) [34], which usually demand very intricate tuning of crossover and mutation parameters, applies very basic coefficient updates that are derived from the natural hierarchy of grey wolf leadership (α, β, δ, ω). This leads to a much quicker and, at the same time, more stable convergence.

BGWO has the capacity to solve binary feature selection problems efficiently, even if the search space is very large and non-linear [35]. It not only looks for the global optima but also exhausts the search where it is near the best areas, thus minimizing the chances of falling into local minima.

In this study, we propose a Vaccine BGWO out of which BGWO is a variant, and within it, we place an adaptive vaccine injection that retains high-quality feature subsets over the iterations as the main technique. This upgrade not only boosts diversity, classification accuracy, and decreases feature redundancy but also suits DDoS detection in SDN environments where timely and accurate decision-making is of utmost importance.

Various methods were used for pre-processing the dataset. Data cleaning for missing values and normalization was done. A set of 25 features is derived from the combined records of the LR-HR DDoS 2024 and InSDN dataset. The features are regarded as the global feature set and include the protocol type, flow duration, bytes in traffic, and Fwd protobufs [27]. Fig. 4 displays the frequency distribution of features BGWO selected features.

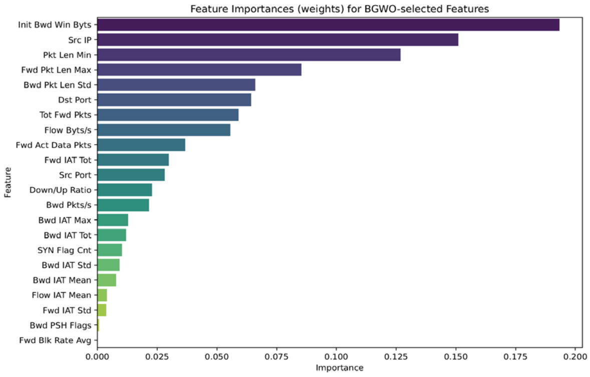


Fig. 4. Feature selection (BGWO selected features).

The traffic data for DDoS attacks and normal conditions was collected in an SDN environment. The data includes some concerning global features like protocol type, flow duration, label, and Fwd bytes, collected in normal conditions and under different DDoS attacks. The collected dataset is divided into training and testing sets according to the proportions of 80% and 20%, respectively. The training set includes 112,491 records, and the testing set includes 28,123 records. Each record in the dataset includes features and a label where normal is denoted as 0 and the DDoS attack is denoted as 1.

The preprocessing included data cleansing, conversion, and feature extraction and assortment [36]. The dataset was cleaned by removing unusable records with null values and transformed by changing the data type of the flow duration from float64 to integer and the bytes in traffic from int64 to float. The enhanced binary grey wolf optimization algorithm was employed to extract and select the optimal feature subset from the global feature set, which includes five selected features: Flow ID, min bytes in traffic, max bytes in traffic, mean bytes in traffic, and std bytes in traffic. Four different machine learning classification algorithms, including XGBoost, Random Forest, Naive Bayes, and the Hidden Markov Model (HMM) are trained and tested using the optimal feature subset. The classification performance was evaluated based on accuracy, precision, recall, and F1-Score for each classifier are compared [2].

E. Model Selection and Evaluation

Basic criteria used for model selection included accuracy, time to train the model, time to detect test instances, and competency in a multi-class scenario. To evaluate the performance of machine learning models in detecting Distributed Denial of Service (DDoS) attacks in Software Defined Networks (SDNs), various models were implemented with different parameters, and their performance was evaluated using different evaluation metrics. Eq. (3) shows the formula for calculating the Accuracy. The model with the highest accuracy was then selected for further analysis. By layering the network and separating the data traffic of the Control Plane (CP) from the Data Plane (DP), the impact of DDoS attacks on the CP of the SDN network was analyzed [8]. The CP is secured by deploying the selected model to detect DDoS attacks in SDN networks. The dataset was used to train and test machine learning models. It was the most comprehensive dataset for network-based Distributed Denial of Service (DDoS) attack detection and is the first to support research on detecting and mitigating DDoS attacks in SDNs [1].

Four machine learning models were implemented to compare the performance. The models include Random Forest (RF), Extreme Gradient Boosting (XGBoost), Naïve Bayes, and Hidden Markov Model (HMM). Each model was implemented with different parameters, and accuracy was used to evaluate the performance of each model [2]. In Table III, the feature selection parameters are displayed in relation to their values used in the experimentation.

TABLE III. FEATURE SELECTION PARAMETERS

Parameters	**Vaccine_bgwo**	**Value**
Number of iterations	iterations	30
Search agents (k)	n_wolves	20
Alpha (α)	mutation_rate	0.01–0.1

The model with the highest accuracy was selected as the best model for DDoS attack detection. Table IV displays common parameters that were used during the training.

TABLE IV. COMMON PARAMETERS

Component	**Control Parameters**	**Value**
Data Splitting	test_size	0.2
	random_state	42
XGBoost	n_estimators	200
	learning_rate	0.1
	max_depth	6
	random_state	42
	eval_metric	logloss
RandomForest	n_estimators	200
	max_depth	5
	random_state	42
HMM	n_iter	100
	random_state	42
	sequence_length	10
Naive Bayes	GaussianNB	-

Furthermore, recall measures the fraction of all positive instances a model correctly identifies. The F1-Score is the harmonic mean of precision and recall [27]. Also, the area under the receiver operating characteristic curve will be measured. Therefore, the above metrics focus only on the number of correct binary classifications, ignoring the type, which means they do not consider false negatives or false positives. The selection of these metrics is reasonable in binary classification tasks in which the primary interest is the model’s ability to recognize all actual positives.

To evaluate the proposed approach against existing SDN-DDoS detection systems, a benchmarking framework was designed using representative machine learning architectures commonly reported in the literature. The benchmark set includes Decision Tree, Naïve Bayes, Linear Support Vector Machine, Logistic Regression, Random Forest, and Extreme Gradient Boosting, which are widely used for SDN traffic classification and intrusion detection.

All benchmark models were trained and evaluated under identical experimental conditions, including the same preprocessing steps, feature sets, and stratified cross-validation protocol. This design ensures a fair and reproducible comparison and allows the performance gains of the proposed method to be attributed to the underlying model design and feature optimization strategy rather than experimental bias.

F. SDN Environment and Controller Module Interaction Flow

The experimental evaluation was conducted in an emulated Software Defined Networking environment by using the Mininet 2.3.0 network simulator. The SDN control logic was managed by the Ryu controller (as the control plane), which supports the protocol for dynamic control data plane communication. The topology consisted of virtual switches and host nodes, organized to mimic practical traffic flows in the SDN infrastructure. Benign and malicious traffic was generated to simulate DDoS scenarios targeting the controller via the data plane.

The Vaccine-Based Grey Wolf Optimization (Vaccine BGWO) feature selection module was implemented in the ML model and then integrated into SDN. Fig. 5 illustrates the SDN integration testbed architecture. Flow level statistics gathered from the data plane were analyzed, and this contributed to detecting and classifying real-time DDoS attack traffic.

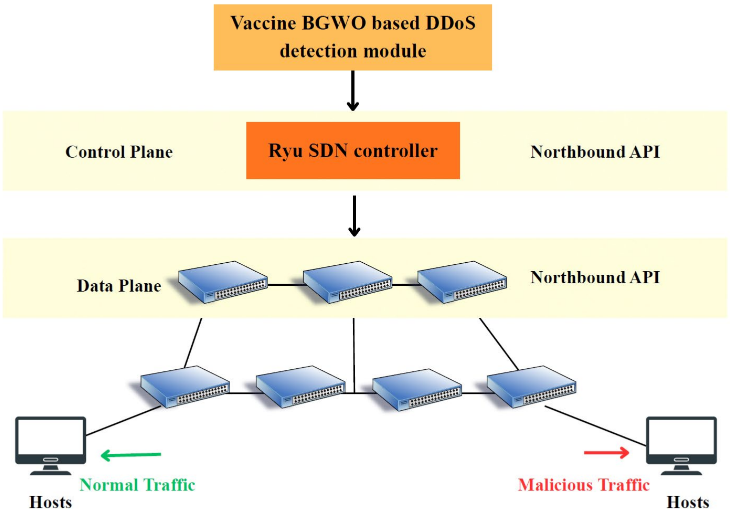


Fig. 5. SDN integration testbed architecture.

The experimentation was benchmarked by LR-HR DDoS 2024, and InSDNdatasets. The datasets were integrated into the Mininet framework with Python-based scripts to enable the simulation of different network loads and attack intensities.

In the suggested architecture, instead of deploying the ML-based DDoS detection model equipped with Vaccine BGWO feature selection directly on the controller itself, both components are placed as part of the application plane, alleviating the control plane from overload and ensuring detection and analysis happen asynchronously from control plane functions related to the network.

The SDN controller (Ryu) periodically exports flow statistics like flow duration, packet count, byte count, and protocol type to the application plane using the Northbound API. The ML model equipped with Vaccine BGWO processes these features to discover the best subset, hence reducing data dimensionality before classification. This approach reduces computational complexity and communication overhead between the controller and the application plane.

This modular integration conserves controller responsiveness and scalability while allowing for intelligent, feature-efficient DDoS detection.

IV. Result and Discussion

The results derived from the evaluations of the selected machine learning classifier models are presented here. The strengths and weaknesses of each model are extensively analyzed, and comprehensive evidence is presented to help researchers and practitioners assess the extent to which the resulting criteria are reliable for DDoS detection. The dataset, scaled as presented and then pre-processed and evaluated, was chosen for this experiment.

The analysis is divided into two parts. The first part presents the evaluation metric scores for each model. Initially, we plotted the Receiver Operating Characteristics (ROC) graph of the area graph for the evaluation of the classifiers and ranked them from top to bottom. The true positive rate and false positive rate relationships are used to create the graph. The higher the score, the more capable the classifier is of distinguishing an imbalance of the two classes. This is a non-negligible aspect because, from the perspective of security practices, it tends to have significant consequences if attacks are falsely reported. The graph provides a comprehensive overview of the model's capability to segregate DDoS from normal traffic.

A. Performance Comparison of Models

Evaluating the models' performance for DDoS detection in an SDN environment using different evaluation metrics. For each evaluation metric, we compared the results of each model using visual comparison. Various performance metrics, including accuracy (A), precision (P), recall (R), and F1-Score (F1), are considered to provide a comprehensive understanding of the models' strengths and weaknesses.

The definitions and mathematical equations for evaluating the proposed machine learning models and classifiers are provided; these evaluation scales are based on True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) [1].

Accuracy is the total number of instances classified correctly (DDoS and Benign) divided by the total number of instances in the dataset, as calculated using Eq. (5). It depicts the proportion of DDoS attacks accurately identified in the dataset.

Depicts the proportion of DDoS attacks accurately identified in the dataset.

$$Ac = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Recall (Sensitivity) is the Recall counter and is the ratio of correctly classified DDoS traffic instances to the total number of DDoS traffic instances. It quantifies the ability of a model to correctly classify the DDoS traffic. Eq. (6) shows mathematically how recall metrics are evaluated.

$$R = \frac{TP}{TP + FN} \tag{6}$$

Precision (Specificity) measures the number of correctly categorized DDoS attacks over the total number of instances classified as DDoS and is described in Eq. (7).

$$P = \frac{TP}{TP + FP} \tag{7}$$

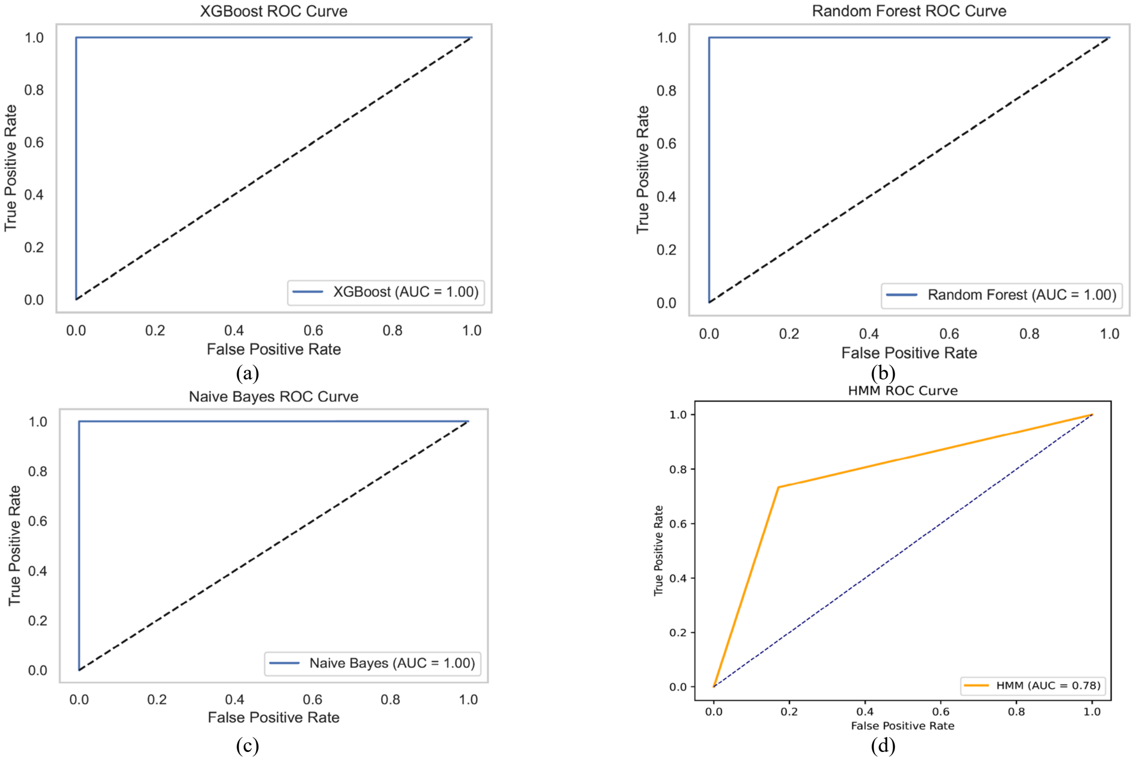
Eq. (8) conveys how the F1-Score is achieved. The F1-Score is defined as the weighted average of Precision and Recall (with a range of 0 to 1).

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{8}$$

TP: True Positives, TN: True Negatives, FP: False Positives, and FN: False Negatives.

B. ROC, F1-Score, and Confusion Matrix Ring

Receiver Operating Characteristic (ROC), F1-Score curves, and confusion matrix are graphical representations of a binary classifier's performance, plotting true positive rates (or sensitivity) against false positive rates. Fig. 6 shows the ROC curves results from trained models. To assess performance, the area Under the ROC Curve (AUC) was computed. The AUC reveals the ability to distinguish between a positive class and a negative class [37]. The area can be interpreted as the probability that queries randomly selected from the positive class will receive higher scores than queries randomly selected from the negative class.

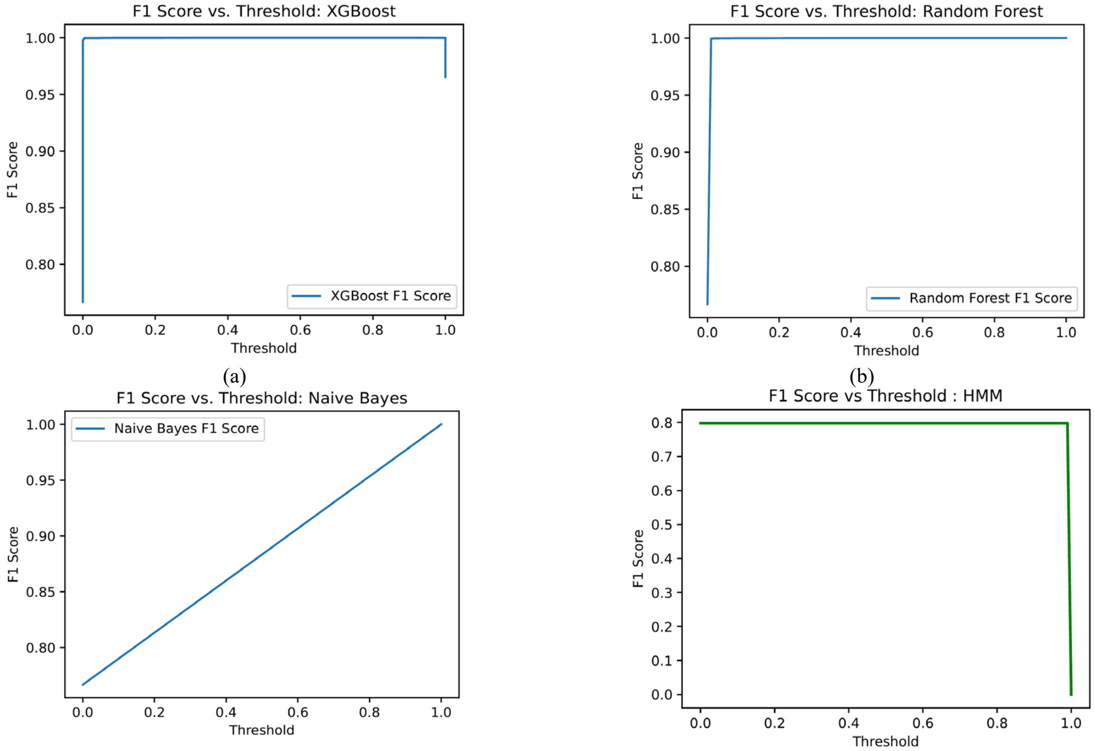


(a) (b) (c) (d)

Fig. 6. ROC curves: (a) XGBoost (b) Random Forest (c) Naïve Bayes (d) HMM.

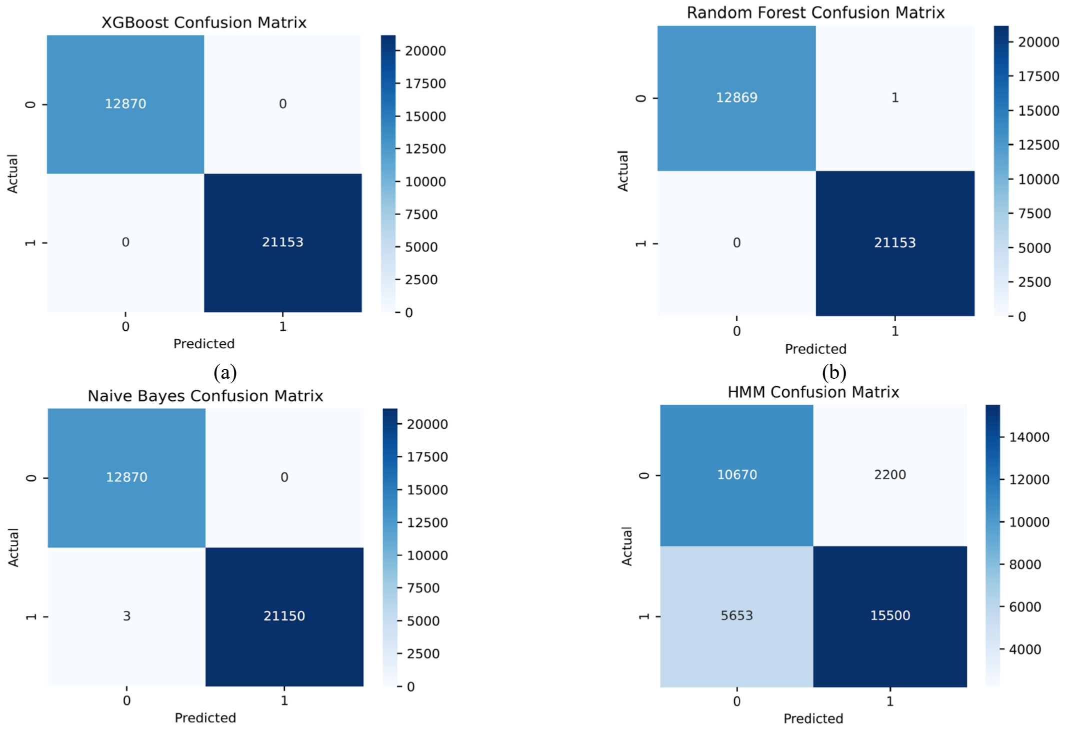
The higher the AUC value, the better the model performance. AUC values near 0.5 represent random chance, while as the values approach 1, the models become ideal. If the values are less than 0.5, it implies that the model is not effective in classification tasks. While it is possible to consider more advanced performance measures based on the ROC curve behavior, it is usually a standard practice to report the AUC along with the ROC curves. The ROC curve is also robust against class imbalance and works well with data with varying distributions.

The F1-Score is defined as the weighted average of Precision and Recall (with a range of 0 to 1). Fig. 7 shows the F1-Score curve for the model results.



(a) (b) (c) (d)

Fig. 7. F1-score curves: (a) XGBoost (b) Random Forest (c) Naïve Bayes (d) HMM.



(a) (b) (c) (d)

Fig. 8. Confusion matrix: (a) XGBoost (b) Random Forest (c) Naïve Bayes (d) HMM.

A confusion matrix evaluates the performance of the classification models. Fig. 8 shows the confusion matrix results. It consists of True Positives (TP), correctly predicted positive cases. True Negatives (TN): Correctly predicted negative cases. False Positives (FP), incorrectly predicted positive cases (actual is negative). False Negatives (FN), incorrectly predicted negative cases (actual is positive).

Furthermore, presenting the trade-offs among models, as well as the features that have the most significant impact on the model's performance, based on our evaluation metrics.

The XGBoost, Random Forest, and Naïve Bayes have shown the best model performance with accuracy of 100%, 99.99%, and 99.98%, respectively. Also, based only on precision, recall, and F1-Score, they have also performed well. Nonetheless, the proposed models' performance based on the Area Under the Curve (AUC) and specificity differed in varying degrees. The difference in size and feature selection is one of several reasons to employ multiple evaluation metrics to measure the model's performance in order to gain a better interpretation of the model's performance. In practice, another consideration in selecting the best deployment model is the trade-off between accuracy and time efficiency in classifying DDoS and non-DDoS attacks. Table V shows the classification results of machine learning models without the equipped vaccine-based BGWO. In terms of time efficiency, the proposed models took longer to perform the calculations, as the models are based mainly on decision trees and perform numerous iterations.

TABLE V. Classification Results of ML Models Without Vaccine-Based (BGWO)

ML	Accuracy/%	Precision/%	Recall/%	F1-Score/%
XGBoost	99.80	99.99	100.0	100.0
RandomForest	99.99	99.99	99.99	99.99
Naïve Bayes	99.98	99.98	99.98	99.98
HMM	76.93	87.57	73.25	79.69

The proposed models' performance also depends on the complex model predictions, i.e., the decision tree-based approaches (RF) are more accurate at classifying DDoS attack traffic than other models. Implementing compression and feature selection to reduce complexity improved the predictive performance of deep learning approaches in general.

The ML detection models equipped with vaccine-based BGWO have improved the performance of models by increasing accuracy and detection latency as well as response time. In defending against DDoS, we must detect attacks as swiftly as possible to facilitate mitigation and the stabilization of the controller. While decision tree classifiers require iteratively evaluating rules, the Vaccine BGWO feature selection approach can effectively reduce the input dimensionality, which will then reduce computation time during model inference. Also, to ensure minimal impact on the controller performance, the experiment was conducted offline. Table VI shows the classification results of ML models equipped with vaccine-based (BGWO).

TABLE VI. Classification Results of ML Models Equipped with Vaccine-Based (BGWO)

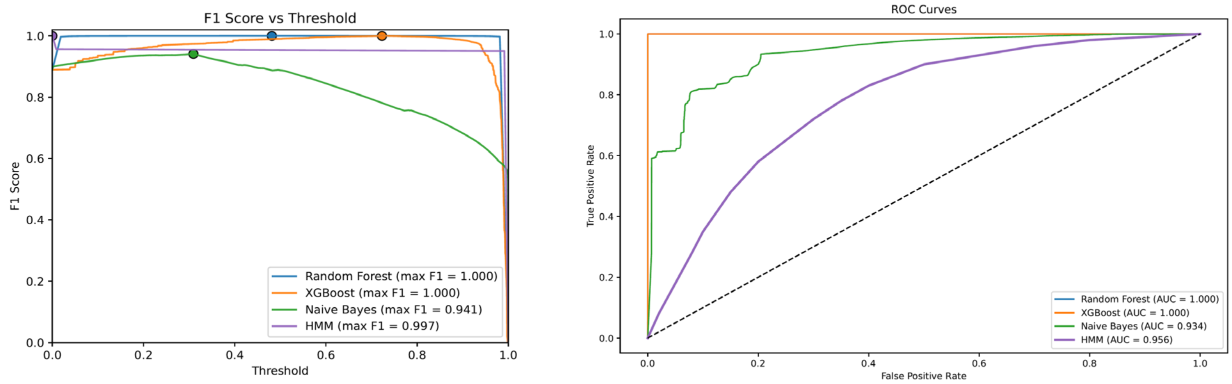
ML + Vaccine-based BGWO	Accuracy/%
XGBoost	100.0
RandomForest	99.99
Naïve Bayes	89.66
HMM	90.02

From Fig. 9(a) and (b), which display F1-Score and ROC Curves respectively of DDoS detection models equipped with vaccine-based BGWO, from the observation we have seen that the F1-Score comparison clearly indicates that tree-based ensemble models (Random Forest, XGBoost) outperform probabilistic (Naive Bayes) and sequence-based (HMM) models for this dataset.

They are ideal for DDoS detection, where false negatives (missed attacks) and false positives (wrong alarms) must both be minimized. But also, on the side of ROC Curves, both Random Forest and XGBoost achieve flawless detection under ROC evaluation, consistent with their F1 results.

HMM performed well but may occasionally misclassify certain traffic flows, while Naive Bayes shows decent but less robust separation.

XGBoost and Random Forest are the best-performing models, offering near-zero false positives and false negatives.



(a) (b)

Fig. 9. F1-Score (a) and ROC curves (b) for ML Models equipped with BGWO.

HMM remains valuable for time-dependent or flow-based traffic modeling in SDN, but slightly underperforms.

Naive Bayes is simple and fast but less precise, making it less suitable for real-time DDoS detection in SDN environments.

Table VII presents a benchmark contrast of classical ML models and the proposed Vaccine BGWO classifiers on the LR-HR DDoS 2024 dataset. The results indicate that all evaluated models achieve high detection accuracy; however, notable differences are observed in terms of false positive rate and real-time performance.

TABLE VII. BENCHMARK COMPARISON (LR-HR DDOS 2024 DATASET)

Model	Accuracy (mean±std)	F1 (mean±std)	AUC (mean±std)	FPR (mean±std)	Latency (ms/sample)	Throughput (samples/s)
DT	$0.999947 \pm 3.7\times10^{-5}$	$0.999957 \pm 3.0\times10^{-5}$	$0.999944 \pm 4.32\times10^{-5}$	$7.0\times10^{-5} \pm 1.04\times10^{-4}$	0.000519±0.000230	$2.19\times10^{6} \pm 7.81\times10^{5}$
Linear SVM	$0.999427 \pm 9.9\times10^{-5}$	$0.999539 \pm 7.9\times10^{-5}$	$1.000000 \pm 9.82\times10^{-8}$	$1.352\times10^{-3} \pm 3.65\times10^{-4}$	0.000802±0.000490	$1.57\times10^{6} \pm 6.95\times10^{5}$
Logistic Regression	$0.999277 \pm 1.01\times10^{-4}$	$0.999419 \pm 8.2\times10^{-5}$	$1.000000 \pm 1.23\times10^{-7}$	$1.842\times10^{-3} \pm 3.23\times10^{-4}$	0.001117±0.000655	$1.26\times10^{6} \pm 8.12\times10^{5}$
Naïve Bayes	$0.999489 \pm 1.15\times10^{-4}$	$0.999589 \pm 9.2\times10^{-5}$	$0.999472 \pm 5.69\times10^{-5}$	$1.142\times10^{-3} \pm 4.24\times10^{-4}$	0.000659±0.000341	$1.76\times10^{6} \pm 6.15\times10^{5}$
Random Forest	$0.999965 \pm 3.7\times10^{-5}$	$0.999972 \pm 3.0\times10^{-5}$	$1.000000 \pm 3.99\times10^{-8}$	$2.3\times10^{-5} \pm 5.2\times10^{-5}$	0.005978±0.001855	$1.81\times10^{5} \pm 5.55\times10^{4}$
XGBoost	$0.999974 \pm 3.9\times10^{-5}$	$0.999979 \pm 3.2\times10^{-5}$	$1.000000 \pm 1.95\times10^{-7}$	0.0±0.0	0.003664±0.001698	$3.11\times10^{5} \pm 1.05\times10^{5}$

The results show that all evaluated machine learning models achieve very high detection accuracy on the LR-HR DDoS 2024 dataset. Throughout all iterations of testing, Random Forest and XGBoost had consistently been rated the highest for both metrics of accuracy and F1-Score, including reaching an FPR of 0 when using the XGBoost algorithm across all folds of the dataset. In contrast to the overall very low latency (<0.001 ms/sample) and high throughput of both Decision Tree and Naive Bayes algorithms, which would be ideal for lightweight application deployment, XGBoost would be a good choice to strike a balance between the efficiency of detection and minimal latency; this would provide close to 100% accuracy for this type of algorithm while also completing inference in milliseconds, making it viable for use as an SDN-based, real-time DDoS defense technique.

Crucially, compared to their full-feature counterparts, the Vaccine-BGWO-optimized models maintain near-perfect detection performance while lowering computational overhead, resulting in lower inference latency and higher throughput. This shows that the suggested feature optimization approach increases deployability in practice without compromising detection efficacy.

On the other hand, Table VIII presents the benchmark evaluation of ML models on the InSDN dataset. After evaluating the models, the ensemble-based classifiers, particularly Random Forest and XGBoost, achieved near-perfect detection performance, with F1-Scores and AUC values approaching unity, while Decision Tree, Logistic Regression, and Linear SVM demonstrate lower inference latency and higher throughput; however, they exhibit higher false positive rates compared to ensemble models. Naïve Bayes performs poorly on the InSDN dataset, yielding significantly lower detection accuracy and an unacceptably high false positive rate.

TABLE VIII. BENCHMARK COMPARISON (INSDN DATASET)

Model	Accuracy (mean±std)	F1 (mean±std)	AUC (mean±std)	FPR (mean±std)	Latency (ms/sample)	Throughput (samples/s)
DT	$0.999834 \pm 7.3\times10^{-5}$	$0.999897 \pm 4.6\times10^{-5}$	$0.999682 \pm 1.96\times10^{-4}$	$5.7\times10^{-4} \pm 4.2\times10^{-4}$	$0.00135 \pm 1.2\times10^{-5}$	$7.39\times10^{5} \pm 6.53\times10^{3}$
Linear SVM	$0.997098 \pm 3.8\times10^{-5}$	$0.998191 \pm 2.4\times10^{-4}$	$0.999079 \pm 3.18\times10^{-4}$	$1.36\times10^{-2} \pm 1.56\times10^{-3}$	$0.00177 \pm 6.4\times10^{-5}$	$5.65\times10^{5} \pm 2.06\times10^{4}$
Logistic Reg.	$0.996967 \pm 9.7\times10^{-5}$	$0.998110 \pm 6.0\times10^{-4}$	$0.999102 \pm 2.82\times10^{-4}$	$1.38\times10^{-2} \pm 6.22\times10^{-4}$	$0.00208 \pm 2.75\times10^{-4}$	$4.87\times10^{5} \pm 6.75\times10^{4}$
Naïve Bayes	$0.898781 \pm 7.8\times10^{-4}$	$0.940009 \pm 4.4\times10^{-3}$	$0.761501 \pm 1.76\times10^{-3}$	$0.468388 \pm 3.82\times10^{-3}$	$0.00254 \pm 2.65\times10^{-4}$	$3.97\times10^{5} \pm 4.58\times10^{4}$
Random Forest	$0.999916 \pm 5.4\times10^{-5}$	$0.999947 \pm 3.4\times10^{-5}$	$0.999993 \pm 1.6\times10^{-5}$	$1.9\times10^{-4} \pm 1.5\times10^{-4}$	$0.00818 \pm 9.43\times10^{-4}$	$1.23\times10^{5} \pm 1.40\times10^{4}$
XGBoost	$0.999921 \pm 5.6\times10^{-5}$	$0.999951 \pm 3.5\times10^{-5}$	$0.999995 \pm 1.1\times10^{-5}$	$2.2\times10^{-4} \pm 1.6\times10^{-4}$	$0.00601 \pm 1.20\times10^{-3}$	$1.71\times10^{5} \pm 2.70\times10^{4}$

Despite their higher computational cost, Random Forest and XGBoost maintain inference times within the millisecond range while achieving the lowest false positive rates. These characteristics make them particularly suitable for SDN controller-level deployment, where detection reliability is critical.

Overall, the InSDN results confirm the trends observed on the LR-HR DDoS 2024 dataset, demonstrating the robustness and consistency of ensemble-based detection across different SDN traffic distributions.

Tables IX and X present a benchmark comparison between full-feature ensemble classifiers and their Vaccine BGWO optimized counterparts on the LR-HR DDoS 2024 and InSDN dataset respectively. The full-feature Random Forest and XGBoost models achieve near-perfect detection performance, with accuracy and F1-Scores approaching unity.

When Vaccine BGWO is applied, the average number of selected features is reduced from 24 to approximately 5, corresponding to an 80% reduction in feature dimensionality. Despite this substantial reduction, detection performance remains high, with only a marginal decrease in accuracy and F1-Score.

Importantly, feature reduction leads to significant improvements in real-time performance. The Vaccine BGWO optimized XGBoost model achieves sub-millisecond inference latency (≈0.0016 ms/sample) and the highest processing throughput, exceeding 6.6×10^{5} samples per second. These results confirm that the proposed approach satisfies the near real-time requirements of SDN-based DDoS defense while maintaining reliable detection capability.

TABLE IX. BENCHMARK PERFORMANCE WITH AND WITHOUT VACCINE-BGWO (LR-HR DDOS 2024 DATASET)

Model	Features (mean±std)	Accuracy (mean±std)	F1 (mean±std)	AUC (mean±std)	FPR (mean±std)	Latency (ms/sample)	Throughput (samples/s)
RF (Full)	24.0±0.0	0.9999647 ±3.69×10^{-5}	0.9999716 ±2.97×10^{-5}	0.9999998 ±4.0×10^{-8}	2.33×10^{-5} ±5.21×10^{-5}	0.0045977 ±0.0010032	225,459.7 ±45,601.7
RF + Vaccine-BGWO (Proposed)	4.8±1.095	0.9984745 ±0.0013155	0.9987756 ±0.0010558	0.9980302 ±0.0018025	0.0039395 ±0.0036051	0.0045724 ±0.0019268	242,593.3 ±72,470.1
XGBoost (Full)	24.0±0.0	0.9999753 ±3.94×10^{-5}	0.9999787 ±3.17×10^{-5}	0.9999991 ±1.95×10^{-7}	0.0±0.0	0.0027880 ±0.0015887	422,852.9 ±143,574.4
XGBoost + Vaccine-BGWO (Proposed)	4.8±1.095	0.9984745 ±0.0013155	0.9987756 ±0.0010558	0.9980302 ±0.0018025	0.0039395 ±0.0036051	0.0015830 ±0.0003556	660,658.2 ±161,346.2

TABLE X. BENCHMARK PERFORMANCE WITH AND WITHOUT VACCINE-BGWO (INSDN DATASET)

Model	Features (mean±std)	Accuracy (mean±std)	F1 (mean±std)	AUC (mean±std)	FPR (mean±std)	Latency (ms/sample)	Throughput (samples/s)
RF (Full)	24±0	0.99996 ±3.7×10^{-5}	0.99997 ±3.0×10^{-5}	0.9999998 ±4.0×10^{-8}	2.3×10^{-5} ±5.2×10^{-5}	0.00460 ±0.00100	2.25×10^{5} ±4.56×10^{4}
RF + Vaccine-BGWO (Proposed)	4.8±1.1	0.99847 ±0.00132	0.99878 ±0.00106	0.99803 ±0.00180	0.00394 ±0.00361	0.00457 ±0.00193	2.43×10^{5} ±7.25×10^{4}
XGBoost (Full)	24±0	0.99998 ±3.9×10^{-5}	0.99998 ±3.2×10^{-5}	0.9999991 ±2.0×10^{-7}	0±0	0.00279 ±0.00159	4.23×10^{5} ±1.44×10^{5}
XGBoost + Vaccine-BGWO (Proposed)	4.8±1.1	0.99847 ±0.00132	0.99878 ±0.00106	0.99803 ±0.00180	0.00394 ±0.00361	0.00158 ±0.00036	6.61×10^{5} ±1.61×10^{5}

V. DISCUSSION

Distributed Denial of Service (DDoS) attacks are serious threats in Software-Defined Networks (SDNs). The main outcomes of this research showed that increasing datasets reflects a substantial improvement in the performance of the predictive models.

Furthermore, the ROC AUC values of the Gradient Boost (XGBoost) are higher than those of Moving Averages, whereas they are about the same or lower in other models. Such an evaluation of results is also highlighted by several studies that have stated the superior performance of Gradient Boost (XGBoost) in DDoS detection.

According to Table VI, Vaccine BGWO-optimized models produced better performance than none vaccine based BGWO models (Table V) in terms of accuracy and latency. The feature selection procedure gave us a significant dimensionality reduction in detection latency. This finding means that the detection process can take place under the constraints of “real-time” SDN implementations without causing the SDN controller overload.

The high performance of GB models clearly presents the suitability of substituting the HMM in DDoS detection, provided that they are trained on the application of a small dataset. Despite some limitations, future work could be performed to implement this replacement and monitor their performance according to the performance of HMM, serving for accuracy in different situations.

In experiment, it was found that the performance of the models increased independently using Gradient Boost (XGBoost), RF, Naive Bayes, and HMM models when the number of observations increased in the none vaccine based BGWO but on the other hand the improved performance of some models like HMM showed us that the proposed vaccine based BGWO feature selection method have produced significant results and hence can we tested to the real time environment.

The poor performance observed in HMM models, even after adapting the dataset to simulate a sequence, is due to the nature of the dataset; the HMM models work very well with time-ordered data, sequence data, and our data was based on tabular as well as non-sequential datasets. The significant classification model can also be utilized by financial, educational, and other institutions to stop cyber attackers in SDN. Even if the tested system performance is accurate, there are some limitations related to the implementation. The experimental part can be used to compare the performance of other DDoS detection systems, fixed feature harms, and reconstruct the system in different scenarios where attack events and benign events gather together may lead to. In addition, the evaluation results have been analysed in detail using the performance of the ROC curve [36].

VI. CONCLUSION

This study has evaluated different machine learning models, intending to identify the most effective model approaches for DDoS detection in an SDN environment. Based on extensive experiments conducted on SDN-Based datasets, this article offers three main contributions:

First, this study has provided insight into the performance of machine learning models for DDoS detection using SDN-based datasets. The results indicate that Gradient algorithms (XGBoost) and Random Forest, as well as Naïve bayes have the best accuracy in the aggregated data scenario. However, in the individual scenarios, this article shows that HMM enhances DDoS detection improve in FTP, HTTP, and UDP datasets. But it has shown no significant advantage over Naïve Bayes in the HTTP and UDP dataset scenarios. Overall, the observed performance variations can be attributed to differences in dataset size and traffic characteristics, where smaller and more controlled datasets tend to yield better real-time detection results.

Second, the results offer valuable insights into the strengths and limitations of supervised machine learning models for DDoS detection in SDN environments. While the evaluated models demonstrate promising detection performance, the experiments were conducted using a limited number of SDN datasets with a relatively small feature set, which may constrain the generalizability of the findings. The study evaluates both aggregated traffic scenarios and individual DDoS traffic scenarios; however, real-world SDN environments often involve more diverse and dynamically evolving traffic patterns.

Accordingly, this work does not claim to address previously unseen or zero-day attack detection. Instead, it focuses on benchmarking supervised learning models for detecting known DDoS attack patterns using labelled SDN datasets, with particular emphasis on detection accuracy, false positive rate, and real-time feasibility.

Future research directions include extending the experimental evaluation to additional SDN datasets and environments to further validate the robustness of the proposed models. Re-training with new normal and attack samples combined could split the line more accurately. Furthermore, deep learning techniques may be explored to complement the supervised learning models and enhance detection performance under complex traffic scenarios.

The outcomes of this work promote several research opportunities in the area of DDoS detection at the SDN infrastructure. For example, one promising direction would be to evaluate the DDoS detection capabilities of the combination of several machine learning algorithms in terms of data fusion and use the fused data to feed and re-train another machine learning algorithm to improve the detection accuracy.

Another potential direction would be to investigate the possibility of using more complex machine learning techniques to distinguish between benign and DDoS-induced network flows in SDN environments. Yet, attacks on sophisticated SDN infrastructure can occur in the future that require advanced transformation of the network structure itself to incorporate real-time demand, beyond traditional DDoS mitigations.

Expanding the dataset, the models have mostly been validated by features captured in the test dataset under normal conditions, which may not catch the attacks that actually occurred but were not detected in the test data period. As an alternative, one can look at using a more diverse set of training data, such as multiple network transits, international data, or government organizational data that could better adapt to the pragmatic conditions. As security problems will constantly change, we need to build a community of shared practice and approach-based detection techniques to overcome DDoS attacks in the era of IoT and gigabit networks. The rapid and dynamic nature of attack trends in the network today can lead researchers and practitioners to develop DDoS methods based on shared and combined scientific results.

While the overall system design supports the concept of filtering malicious traffic at the SDN ingress, the current study focuses exclusively on the detection and classification stage, leveraging Vaccine-Based Grey Wolf Optimization (BGWO) for feature selection and efficiency improvement. The implementation of mitigation mechanisms, such as dynamic flow-rule installation at ingress switches guided by controller feedback, is identified as a future extension of this work. As well as this study does not address online or adaptive detection mechanisms for evolving attacker behavior and focuses instead on evaluating supervised learning models under static, labeled SDN traffic conditions.

Finally, this study contributes to the growing body of research on real-time DDoS detection in SDN and IoT-oriented environments. By providing a systematic benchmarking framework and highlighting efficiency-oriented optimization strategies, the results support practical deployment considerations for SDN security systems. The findings may assist network operators, security teams, and researchers in selecting appropriate detection models based on their performance, efficiency, and deployment objectives.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Ö.T.: Conceptualization, Methodology, Supervision, Writing—Original Draft Preparation, Writing—Review and Editing, Visualization. J.A.M.: Conceptualization, Methodology, Software, Data curation, Writing—Original draft preparation, Visualization, Investigation. All authors had approved the final version.

References

[1] A. Hamarshe, H. I. Ashqar, and M. Hamarsheh, "Detection of DDoS attacks in software defined networking using machine learning models," in *Proc. International Conference on Advances in Computing Research*, 2023, pp. 640–651.

[2] Z. Liu, Y. Wang, F. Feng *et al.*, "A DDoS detection method based on feature engineering and machine learning in software-defined networks," *Sensors*, vol. 23, no. 13, Jul. 2023. doi: 10.3390/s23136176

[3] S. Mehmood, R. Amin, J. Mustafa *et al.*, "Distributed Denial of Services (DDoS) attack detection in SDN using optimizer-equipped CNN-MLP," *PLoS One*, vol. 20, no. 1, 2025. doi: 10.1371/journal.pone.0312425

[4] Ö. Tonkal, H. Polat, E. Başaran *et al.*, "Machine learning approach equipped with neighbourhood component analysis for DDoS attack detection in software-defined networking," *Electronics*, vol. 10, no. 11, Jun. 2021. doi: 10.3390/electronics10111227

[5] F. M. Salem, H. Youssef, I. Ali, and A. Haggag, "A variable-trust threshold-based approach for DDoS attack mitigation in software defined networks," *PLoS One*, vol. 17, no. 8, e0273681, Aug. 2022. doi: 10.1371/journal.pone.0273681

[6] A. B. Dehkordi, M. R. Soltanaghaei, and F. Z. Boroujeni, "The DDoS attacks detection through machine learning and statistical methods in SDN," *Journal of Supercomputing*, vol. 77, no. 3, Mar. 2021. doi: 10.1007/s11227-020-03323-w

[7] C. Singh and A. K. Jain, "A comprehensive survey on DDoS attacks detection & mitigation in SDN-IoT network," *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 8, Jun. 2024, doi: 10.1016/j.prime.2024.100543

[8] J. Singh and S. Behal, "Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future

directions," *Computer Science Review*, vol. 37, 2020. doi: 10.1016/j.cosrev.2020.100279

[9] J. Mgungile and Ö. Tonkal, "Scalable intrusion detection in IoT networks: A big data analytics approach," *Turkish Journal of Engineering*, vol. 10, no. 1, pp. 230–243, Dec. 2025. doi: 10.31127/tuje.1793847

[10] E. Alhajjar, P. Maxwell, and N. Bastian, "Adversarial machine learning in network intrusion detection systems," *Expert Syst. Appl.*, vol. 186, Dec. 2021. doi: 10.1016/j.eswa.2021.115782

[11] U. B. Clinton, N. Hoque, and K. R. Singh, "Classification of DDoS attack traffic on SDN network environment using deep learning," *Cybersecurity*, vol. 7, no. 1, Dec. 2024. doi: 10.1186/s42400-024-00219-7

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

[13] M. Abdallah, N. A. L. Khac, H. Jahromi *et al.*, "A hybrid CNN-LSTM based approach for anomaly detection systems in SDNs," in *Proc. the 16th International Conference on Availability, Reliability and Security*, Aug. 2021. doi: 10.1145/3465481.3469190

[14] D. M. Rajan and D. J. Aravindhar, "Detection and mitigation of DDOS attack in SDN environment using hybrid CNN-LSTM," *Migr. Lett.*, vol. 20, pp. 407–419, 2023.

[15] A. Bajenaid *et al.*, "Towards robust SDN security: A comparative analysis of oversampling techniques with ML and DL classifiers," *Electronics*, vol. 14, no. 5, Feb. 2025. doi: 10.3390/electronics14050995

[16] M. A. Setitra, M. Fan, B. L. Y. Agbley, and Z. E. A. Bensalem, "Optimized MLP-CNN model to enhance detecting DDoS attacks in SDN environment," *Network*, vol. 3, no. 4, pp. 538–562, Dec. 2023. doi: 10.3390/network3040024

[17] A. Mudgal, A. Verma, M. Singh *et al.*, "FloRa: Flow table low-rate overflow reconnaissance and detection in SDN," *IEEE Transactions on Network and Service Management*, vol. 21, no. 6, pp. 6670–6683, Dec. 2024. doi: 10.1109/TNSM.2024.3446178

[18] S. Z. Omer, F. Hashim, A. Sali *et al.*, "Binary classification of low-rate DoS attacks using Long Short-Term Memory Feed-Forward (LSTM-FF) Intrusion Detection System (IDS)," *Engineering Science and Technology, an International Journal*, vol. 66, 102049, Jun. 2025. doi: 10.1016/j.jestch.2025.102049

[19] A. M. Elshewey, S. Abbas, A. M. Osman *et al.*, "DDoS classification of network traffic in software defined networking SDN using a hybrid convolutional and gated recurrent neural network," *Sci. Rep.*, vol. 15, no. 1, Dec. 2025. doi: 10.1038/s41598-025-13754-1

[20] S. Ibrahim, A. M. Youssef, M. Shoman, and S. Taha, "Intelligent SDN to enhance security in IoT networks," *Egyptian Informatics Journal*, vol. 28, 100564, Dec. 2024. doi: 10.1016/j.eij.2024.100564

[21] H. Y. I. Khalid and N. B. I. Aldabagh, "A survey on the latest intrusion detection datasets for software defined networking environments," *Engineering, Technology and Applied Science Research*, vol. 14, no. 2, pp. 13190–13200, 2024. doi: 10.48084/etasr.6756

[22] S. A. Madoune *et al.*, "A novel approach for real-time DDoS detection in SDN using dimensionality reduction and ensemble learning," *Journal of Information Security and Applications*, vol. 94, 104195, Nov. 2025. doi: 10.1016/j.jisa.2025.104195

[23] E. P. E. Cuesta, J. C. M. Quintero, and J. D. A. Palma, "DDoS attacks detection in SDN through network traffic feature selection and machine learning models," *Telecom.*, vol. 6, no. 3, 69, Sep. 2025. doi: 10.3390/telecom6030069

[24] Z. Latif, K. Sharif, F. Li *et al.*, "A comprehensive survey of interface protocols for software defined networks," *Journal of Network and Computer Applications*, vol. 156, 2020. doi: 10.1016/j.jnca.2020.102563

[25] Z. Mustafa, R. Amin, H. Aldabbas, and N. Ahmed, "Intrusion detection systems for software-defined networks: A comprehensive study on machine learning-based techniques," *Cluster Comput.*, vol. 27, no. 7, pp. 9635–9661, Oct. 2024. doi: 10.1007/s10586-024-04430-6

[26] J. Halladay *et al.*, "Detection and characterization of DDoS attacks using time-based features," *IEEE Access*, vol. 10, pp. 49794–49807, 2022. doi: 10.1109/ACCESS.2022.3173319

[27] A. Hirsi *et al.*, "Detecting DDoS threats using supervised machine learning for traffic classification in software defined networking," *IEEE Access*, 2024. doi: 10.1109/ACCESS.2024.3486034

[28] P. Kumari, A. K. Jain, and A. Sharma, "An adaptive framework for real-time detection and mitigation of DDoS attacks in software-defined networks," *Peer-to-Peer Networking and Applications*, vol. 19, no. 1, 31, 2026. doi: 10.1007/s12083-025-02180-9

[29] R. Ma, Q. Wang, X. Bu, and X. Chen, "Real-time detection of DDoS attacks based on random forest in SDN," *Applied Sciences*, vol. 13, no. 13, Jul. 2023. doi: 10.3390/app13137872

[30] M. A. Mohsin and A. H. Hamad, "Performance evaluation of SDN DDoS attack detection and mitigation based random forest and K-nearest neighbors machine learning algorithms," *Revue d'Intelligence Artificielle*, vol. 36, no. 2, pp. 233–240, Apr. 2022. doi: 10.18280/ria.360207

[31] I. Masud, K. Kusrini, and A. B. Prasetio, "Distributed Denial of Service (DDoS) attack detection on Zigbee protocol using naive bayes algorithm," *International Journal of Artificial Intelligence Research*, vol. 5, no. 2, Jun. 2021. doi: 10.29099/ijair.v5i2.214

[32] A. Ahmed. (Mar. 2024). LR-HR DDoS 2024 dataset for SDN-based networks. [Online]. Available: https://www.kaggle.com/datasets/abdussalamahmed/lr-hr-ddos-2024-dataset-for-sdn-based-networks

[33] M. S. Elsayed, N.-A. Le-Khac, and A. D. Jurcut, "InSDN: A novel SDN intrusion dataset," *IEEE Access*, vol. 8, pp. 165263–165284, 2020. doi: 10.1109/ACCESS.2020.3022633

[34] A. M. El-ashry, M. F. Alrahmawy, and M. Z. Rashad, "Enhanced quantum inspired grey wolf optimizer for feature selection," *Matrix*, 2020. doi: 10.5815/ijisa.2020.03.02

[35] J. Y. Khaseeb, A. Keshk, and A. Youssef, "Improved binary grey wolf optimization approaches for feature selection optimization," *Applied Sciences*, vol. 15, no. 2, 489, Jan. 2025. doi: 10.3390/app15020489

[36] M. I. Rizaldi, D. R. Chandranegara, and D. R. Akbi, "Comparison of machine learning techniques for classification of distributed denial of service attacks based on feature engineering in SDN-based Networks," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 9, no. 3, pp. 1180–1197, Aug. 2024. doi: 10.29100/jipi.v9i3.5262

[37] S. Rajabi, S. Jamali, and J. Javidan, "An intrusion detection system in computer networks using the firefly algorithm and the fast learning network," *International Journal of Web Research*, vol. 3, no. 1, pp. 50–56, 2020.