IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

IEEE SYSTEMS, MAN AND CYBERNETICS SOCIETY SECTION

Received 13 March 2024, accepted 29 March 2024, date of publication 3 April 2024, date of current version 16 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3384398

RESEARCH ARTICLE

Enhancing DDoS Attack Detection and Mitigation in SDN Using an Ensemble Online Machine Learning Model

ABDUSSALAM AHMED ALASHHAB[1,2], MOHD SOPERI ZAHID[2], (Member, IEEE), BABANGIDA ISYAKU[3], ASMA ABBAS ELNOUR[4], WAMDA NAGMELDIN[5], ABDELZAHIR ABDELMABOUD[6], TALAL ALI AHMED ABDULLAH[2], AND UMAR DANJUMA MAIWADA[2]

[1]Department of Computer Science, Faculty of Information Technology, Alasmarya Islamic University, Zliten, Libya
[2]Department of Computer and Information Science, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia
[3]Department of Computer Science, Faculty of Computing and Information Technology, Sule Lamido University, Kano, Jigawa 700271, Nigeria
[4]Computer Department, Applied College, Girls Section, King Khalid University, Muhayel, Aseer 62529, Saudi Arabia
[5]Department of Information Systems, College of Computer Engineering, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[6]Humanities Research Center, Sultan Qaboos University, Seeb 123, Oman

Corresponding author: Abdussalam Ahmed Alashhab (abdussalaam91@ieee.org)

This work was supported by the Deanship of Scientific Research, King Khalid University, through the Large Research Groups Project RGP.2/127/44.

ABSTRACT Software Defined Networks (SDN) offer dynamic reconfigurability and scalability, revolutionizing traditional networking. However, countering Distributed Denial of Service (DDoS) attacks remains a formidable challenge for both traditional and SDN-based networks. The integration of Machine Learning (ML) into SDN holds promise for addressing these threats. While recent research demonstrates ML's accuracy in distinguishing legitimate from malicious traffic, it faces difficulties in handling emerging, low-rate, and zero-day DDoS attacks due to limited feature scope for training. The ever-evolving DDoS landscape, driven by new protocols, necessitates continuous ML model retraining. In response to these challenges, we propose an ensemble online machine-learning model designed to enhance DDoS detection and mitigation. This approach utilizes online learning to adapt the model with expected attack patterns. The model is trained and evaluated using SDN simulation (Mininet and Ryu). Its dynamic feature selection capability overcomes conventional limitations, resulting in improved accuracy across diverse DDoS attack types. Experimental results demonstrate a remarkable 99.2% detection rate, outperforming comparable models on our custom dataset as well as various benchmark datasets, including CICDDoS2019, InSDN, and slow-read-DDoS. Moreover, the proposed model undergoes comparison with industry-standard commercial solutions. This work establishes a strong foundation for proactive DDoS threat identification and mitigation in SDN environments, reinforcing network security against evolving cyber risks.

INDEX TERMS DDoS attacks, LDDoS attacks, SDN, OML, ensemble, detection and mitigation.

I. INTRODUCTION

In the dominion of modern networking, Software Defined Networks (SDN) have emerged as a transformative paradigm, offering dynamic reconfigurability and scalability that stands in stark contrast to traditional networking architectures [1]. This shift towards SDN has ushered in an era of unparalleled agility and efficiency in network management, enabling organizations to swiftly adapt to evolving demands and architectural requirements. However, amidst the promise and potential of SDN, one enduring challenge remains as formidable as ever: the mitigation of Distributed Denial of Service (DDoS) attacks.

DDoS attacks, characterized by a barrage of malicious traffic orchestrated from multiple sources, continue to pose a significant threat to the availability and integrity of network

The associate editor coordinating the review of this manuscript and approving it for publication was Huiyan Zhang.

services [2], [3]. This threat is not confined to traditional networks alone; rather, it extends its ominous shadow over SDN-based networks as well. The inherent flexibility and programmability of SDN, while highly advantageous in many respects, also introduce novel attack vectors and complexities in DDoS mitigation [4], [5], [6].

Recognizing the pressing need to strengthen SDN against persistent DDoS threats, the integration of Machine Learning (ML) techniques into SDN architecture has emerged as a promising strategy. Recent studies have showcased the effectiveness of ML models in accurately distinguishing legitimate network traffic from malicious attacks [7]. However, as the DDoS landscape evolves with the rise of low-rate and zero-day attacks, the availability of features for model training becomes increasingly restricted [8], [9], [10]. posing challenges to the model's adaptability and efficacy against evolving attack tactics. In response, we present an innovative solution aimed at enhancing DDoS detection and mitigation within SDN environments through an ensemble online machine-learning model, meticulously designed to overcome the limitations of traditional static models.

This model employs principles of online learning, ensuring continuous adaptation and refinement by assimilating emerging attack patterns. It can continuously train on real-time network traffic, employing refined feature selection processes to promptly detect DDoS attacks and update itself as attack patterns evolve. Ensemble methods, renowned for their superior performance compared to individual classifiers, achieve this by amalgamating diverse models, resulting in reduced errors and enhanced detection accuracy [11], [12]. Challenges such as high bias (underfitting) or high variance (overfitting) typically faced by individual classifiers [13], are addressed by our ensemble method, which harmonizes multiple models to strike a balance between bias and variance, ultimately elevating overall performance.

The proposed ensemble method incorporates four OML classifiers: BernoulliNB [14], Passive-Aggressive [15], SGD [16], and MLP [17]. This approach significantly enhances attack identification, adeptly manages concept drift, and strengthens SDN security. In summary, our work contributes in three main aspects:

- Proposing an ensemble online machine-learning model designed for detecting and mitigating evolving DDoS attacks, including zero-day, high-rate, and low-rate attacks, in SDN environments.
- Enhancing the model's adaptability and resilience in the dynamic DDoS landscape through continuous updates to effectively counter evolving threats.
- Rigorously validating the proposed model's performance through experiments conducted in SDN simulation environments using diverse datasets, including CICDDoS2019, InSDN, and slow-read-DDoS.

The subsequent sections of this paper are structured as follows: Section II delves into the preliminaries, offering a comprehensive overview of key concepts such as Software-Defined Networking (SDN), Low-rate DDoS attacks, Zero-day DDoS attacks, Online Machine Learning (OML), and ensemble Machine Learning. Moving on to Section III, we conduct a thorough review of various prominent approaches designed for the detection and mitigation of DDoS attacks within SDN environments. These approaches leverage diverse machine learning techniques to bolster the security of SDN networks. In Section IV, we present our proposed comprehensive system architecture, providing an in-depth exploration of its core components tailored specifically for the detection and mitigation of DDoS attacks within SDN environments. Sections V and VI detail the evaluation and experiment setup, encompassing information on datasets and traffic generation utilized for model training. Section VI presents the findings of the results, accompanied by a detailed discussion. Finally, the concluding Section VII wraps up this paper, summarizing key insights and paving the way for future directions in this domain.

II. PRELIMINARIES

This section provides an overview of Software-Defined Networking, Low-rate DDoS attacks, Zero-day DDoS attack, Online Machine Learning, and Ensemble Model.

A. SOFTWARE DEFINED NETWORKS (SDN)

Software Defined Networks (SDN) represent a revolutionary shift in network architecture, fundamentally altering the way networks are designed, managed, and operated. In traditional network infrastructures, the control plane and data plane functions are tightly integrated into network devices such as switches and routers. SDN decouples these functions, enabling centralized control of network resources through a logically centralized controller, often referred to as the SDN controller [18].

The SDN architecture comprises three layers: the Data layer, Control layer, and Application layer, as illustrated in Figure 1. At the core of SDN is the SDN controller, which communicates with network devices through a standardized

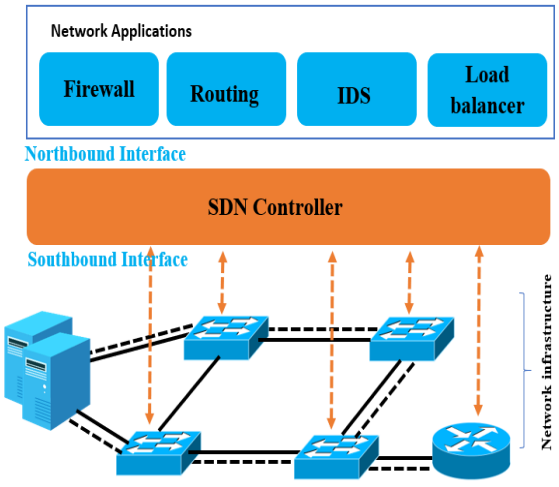

FIGURE 1. Software defined networking architecture.

protocol like OpenFlow [19]. The controller maintains a global view of the network and utilizes this knowledge to make dynamic decisions regarding traffic routing, quality of service (QoS) policies, and security enforcement. SDN switches, typically situated at the network's edge, forward traffic based on instructions received from the controller. This centralized control and programmability empower network administrators to adapt quickly to changing network conditions and traffic patterns [20].

B. LOW-RATE DISTRIBUTED DENIAL OF SERVICE (LDDoS) ATTACKS

DDoS attacks pose a pervasive threat to network availability and performance. While traditional DDoS attacks flood a target with an overwhelming volume of traffic, low-rate DDoS attacks employ a subtler approach [21]. In a LDDoS attack, malicious traffic is injected at a lower rate, making it less conspicuous and more challenging to detect using traditional threshold-based methods [22].

LDDoS attacks often involve techniques such as Slowloris [23] and RUDY [24], which exploit vulnerabilities in web servers by establishing connections but sending minimal, legitimate-looking requests over extended periods. These attacks aim to exhaust server resources, leading to a degradation of service quality without the extreme bandwidth consumption associated with traditional DDoS attacks [25].

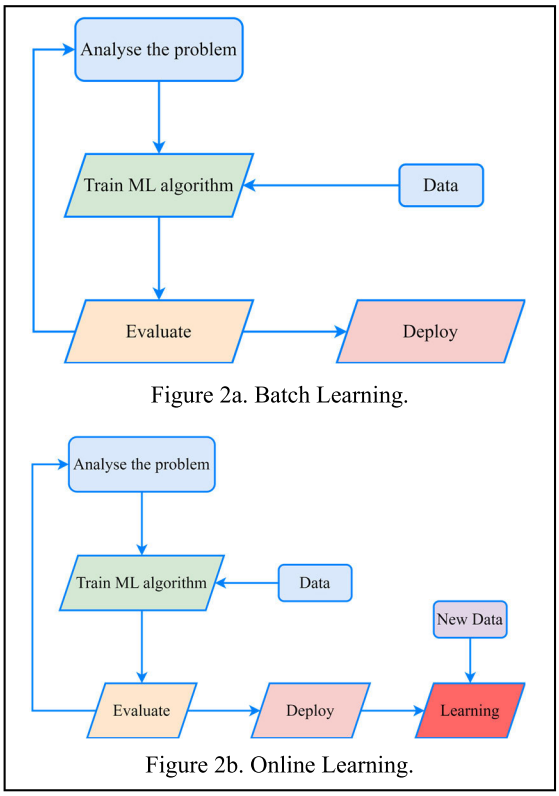

FIGURE 2. Batch and online machin learning.

C. ZERO-DAY DDoS ATTACKS

Zero-day DDoS attacks exploit vulnerabilities in the network on the same day the vulnerability becomes known to the public, or even before a fix or patch is available. In other words, these attacks take advantage of security flaws that are "zero days old," meaning there are no official patches or solutions to protect against them because they are newly discovered. Zero-day DDoS attacks pose significant challenges for organizations and security experts, as they often require immediate and creative solutions to mitigate the impact [26].

Traditional Machine learning mechanisms can struggle to detect zero-day DDoS attacks because they rely on historical data to learn patterns and anomalies [27]. Zero-day attacks, by definition, have not been previously observed, so there is no historical data to train the model. Therefore, machine learning models may not recognize these attacks as anomalous behavior. Machine learning models require carefully engineered features to make predictions. For zero-day attacks, it's challenging to define relevant features because their characteristics are unknown. Models trained on features specific to known attacks may not generalize well to unknown threats. also, attackers continually evolve their tactics, making it challenging for static machine learning models to keep up with rapidly changing attack strategies.

D. ONLINE MACHINE LEARNING (OML)

Machine Learning (ML) has emerged as a potent tool for enhancing network security, equipping us with the capability to scrutinize vast troves of network data and promptly detect anomalies in real-time. A specialized variant of ML, known as Online Machine Learning (OML), accentuates adaptability and perpetual learning from streaming data streams [28].

Key Characteristics of OML involve its intrinsic design for processing data incrementally. It meticulously updates its models as new data streams in. This innate real-time adaptability positions OML as an ideal choice for tasks such as network intrusion detection and the mitigation of DDoS attacks, especially crucial in contexts where threats can metamorphose with remarkable celerity.

To differentiate between the two paradigms, we turn to Figure 2a which illustrates batch learning. In this conventional approach, a machine learning model is meticulously trained using the entire corpus of available data. Subsequently, the trained model is archived and deployed without further learning. This practice, albeit reliable, can be time-consuming, particularly when processing substantial datasets. It is imperative to note that while the model can be updated post-deployment, it remains static during the learning process.

In contrast, Figure 2b depicts Online Learning, a dynamic paradigm wherein a model undergoes continuous updates with small, incremental increments of new data as it becomes accessible. This affords the model the capacity to perpetually learn and adapt to the ever-evolving data landscape. The key steps in Online Learning encompass:

1) Initializing the model with an initial dataset.

2) As new data emerges, the model incrementally absorbs this data, accommodating either single data points or mini-batches.
3) The model remains in a state of continual learning and adapts promptly to shifting data patterns, even after deployment.

Online learning is crucial for dynamic DDoS attack detection, adapting swiftly to changing data patterns. A robust detection system requires a model that continuously improves and adapts in real-time to stay current.

OML excels in mitigating the challenges posed by ever-changing data distributions and concept drift, frequent adversaries in network security. It effortlessly adapts to shifting network behavior and extends the capability of proactive threat detection, even when faced with previously unseen attack patterns.

E. ENSEMBLE MACHINE LEARNING MODEL

Ensemble learning, a potent approach in machine learning, involves the incorporation of multiple models to create a more resilient and accurate predictive system [35]. The core concept revolves around leveraging the strengths of diverse models to compensate for individual weaknesses, ultimately enhancing the overall predictive performance. This collaborative strategy reduces overfitting risks and captures nuanced patterns in the data, making ensemble models particularly effective across a range of applications [36].

Ensemble models come in various types, each with its unique methodology. Bagging, exemplified by Random Forests, concurrently trains multiple models on different subsets of the training data to generate a robust aggregate prediction. Boosting techniques, such as AdaBoost and Gradient Boosting, sequentially train models, emphasizing instances that were previously misclassified to improve overall accuracy. Additionally, stacking combines predictions from multiple models using a meta-learner, aiming to leverage the diverse strengths of individual models for a more comprehensive outcome [36].

Despite their effectiveness, ensemble models pose challenges such as computational expenses and the need for hyperparameter tuning, requiring careful balance between model complexity and available resources [37]. Evaluation metrics like accuracy, precision, and recall ensure robust generalization performance across various domains, highlighting ensemble learning's versatility in real-world applications such as image recognition, finance, and healthcare.

In the context of DDoS attack detection in SDN environments, ensemble methods offer stability, effective handling of imbalanced data, and adaptability to changing network conditions. Leveraging model diversity allows for capturing a broad range of relevant features crucial for detection. In this study, we opted for ensemble method over other machine learning approaches for DDoS detection in SDN environments due to its robustness and adaptability. By combining multiple base learners, ensemble method provides a reliable detection mechanism capable of handling diverse and evolving DDoS attacks. It adapts well to changing network conditions, ensuring real-time response capabilities for rapid attack detection and mitigation. Additionally, ensemble method effectively handles imbalanced data and integrates diverse modeling approaches, enhancing overall predictive performance and enabling detection of a wide range of DDoS attack patterns.

III. RELATED WORK

In this section, we review several notable approaches for detecting and mitigating DDoS attacks in SDN environments. These approaches employ various machine learning techniques, to enhance the security of SDN networks.

Ribeiro et al. [29], introduce an adaptable architecture that combines SDN and Moving Target Defense (MTD) to combat DDoS attacks. Their solution redirects attack traffic to a controlled server, ensuring uninterrupted service for legitimate users. A sensor employing ensemble modeling with Gaussian Naive Bayes (GNB), Support Vector Machine (SVM), Random Forest (RF), and Multilayer Perceptron (MLP) algorithms facilitates flow classification. Key contributions include MTD integration with SDN, machine learning-based DDoS diagnosis, enhanced detection using various models, and proactive defense rules. However, its performance sensitivity to network conditions and reliance on secondary servers in practical scenarios warrant further investigation.

Tonka et al. [30], present an Ensemble machine learning approach with Neighbourhood Component Analysis (NCA) for DDoS attack detection in SDN, achieving high accuracy, with Decision Tree (DT) reaching 100% classification success. However, this approach is limited to predefined features, making it less effective against novel or emerging attack patterns.

Deepa et al. [31], propose an Ensemble Learning Methods approach using multiple ML algorithms (KNN, Naive Bayes, SVM, and SOM) to detect anomalous traffic behavior in the SDN controller. The ensemble method outperforms single algorithms in terms of accuracy, detection rate, and false alarm rate, improving DDoS detection in SDN environments.

Other research papers propose methods for detecting and mitigating LDDoS attacks in SDN environments.

Jess et al. [31], introduce a modular architecture utilizing six machine learning models to train an intrusion detection system (IDS). With a 95% detection rate, their approach proves effective. However, these methods can increase controller overhead and reduce response efficiency in large networks.

Khamkar et al. [33], propose an LDDoS attack identification and defense framework employing the SVM algorithm. With an accuracy of 99%, the framework successfully identifies and mitigates LDDoS attacks. However, the process of identifying features for effective rule creation is not addressed, limiting its efficiency and network connectivity.

Sudar et al. [34], present a flow-based detection and mitigation framework with 93% accuracy in traffic classification,

TABLE 1. Comparison of existing approaches using ML algorithms.

Reference	Year	Detection	Mitigation	DDoS	LDDoS	Method	Dataset	Limitation
Ribeiro et al [29]	2023	✓	✓	✓		GNB, SVM, RF, MLP	CICDDOS2019	Sensitivity to network conditions and unclear scalability. Limited training diversity.
Tonka et al[30]	2021	✓		✓		kNN, DT, ANN and SVM	DDoS attack SDN Dataset	Fixed features limit adaptation to new DDoS patterns. Variable effectiveness and false positives.
Deepa et al [31]	2019	✓		✓		KNN, NB, SVM and SOM	CAIDA 2016	High computational requirements, potential for false positives, and data requirements.
JESÚS et al[32]	2020	✓	✓		✓	J48, RT, RF, MLP	CIC DoS 2017	Reliance on dataset features. Limited evaluation on evolving attacks and practical deployment challenges
Khamkar et al[33]	2021	✓	✓		✓	SVM	KDD99	Computational limitations and challenges detecting evolving attacks.
K. Sudar et al[34]	2022	✓	✓		✓	SVM, DT, NB	CICDDOS2019	Challenges detecting new attack variations and limited evaluation scope.

reducing resource consumption. Nonetheless, it exhibits a high false positive rate for certain traffic flows, such as ICMP.

Table 1 provides a summary overview of the main points of these existing approaches, highlighting their strengths and limitations for DDoS detection and mitigation in SDN environments.

Traditional classification machine learning algorithms often struggle to detect evolving DDoS attacks, including zero-day attacks, because they rely on historical data to learn patterns and anomalies. Zero-day attacks, by definition, have not been previously observed, rendering historical data ineffective for training models. Consequently, these approaches may fail to recognize zero-day attacks as anomalous behavior. Furthermore, these models require carefully engineered features to make predictions, posing a significant challenge for zero-day attacks where relevant characteristics are unknown. Models trained on features specific to known attacks may struggle to generalize to unknown threats. Additionally, attackers continuously evolve their tactics, posing challenges for static machine learning models to keep pace with rapidly changing attack strategies.

To address these challenges, our model dynamically adapts in real-time to emerging threats without relying on known signatures. By combining multiple classifiers, our approach improves accuracy and enhances the ability to detect evolving threats. Our approach to DDoS detection distinguishes itself through the utilization of online ensemble learning, which dynamically adjusts to evolving network conditions and concept drift in real-time. In contrast to traditional methods that rely on static datasets and offline training, our approach continuously learns from streaming data, ensuring an immediate response to emerging threats. The comparison table provided in Table 2 outlines the innovations of our approach.

The proposed model employs a dynamic feature selection mechanism that continuously adjusts its feature set based on the evolving characteristics of network traffic. This dynamic approach enables the model to adapt swiftly to changes in network conditions and emerging attack patterns, enhancing its ability to detect novel and evolving threats effectively. By selecting relevant features in real-time, the model optimizes detection efficacy while minimizing computational complexity, ensuring efficient utilization of resources in rapidly changing environments.

TABLE 2. Comparison of proposed approach with current approaches.

Aspect	Our Approach	Current Approaches
Learning Paradigm	Online ensemble learning	Offline batch learning
Adaptability to Concept Drift	Dynamically adjusts to concept drift	May require periodic retraining
Real-time Response	Immediate response to streaming data	Delayed response due to batch processing
Scalability	Handles large volumes of streaming data	Limited by batch processing capabilities
Resource Efficiency	Efficient utilization of resources	Potentially high resource requirements for batch processing
Flexibility in Deployment	Suitable for dynamic network environments	May require reconfiguration for changes

Our approach offers superior adaptability to concept drift, enabling immediate responses to streaming data, scalability for handling large volumes of data efficiently, and flexibility in deployment for dynamic network environments. This contrasts with traditional approaches that may struggle with delayed responses, scalability limitations, and resource inefficiencies associated with batch processing.

IV. ARCHITECTURAL DESIGN

This section introduces our comprehensive system architecture, providing an in-depth exploration of its core components tailored for detecting and mitigating both high and low-rate

DDoS attacks within SDN environments. Our system encompasses three primary modules:

1) **Traffic Collector Module:** Responsible for capturing and recording network packets as they traverse the SDN environment. This module serves as the initial data source for subsequent analysis.
2) **Online Machine Learning-based Intrusion Detection System (OML-based IDS):** Utilizes machine learning techniques to process and classify incoming traffic in real-time, identifying potential DDoS attacks.
3) **Online Machine Learning-based Intrusion Prevention System (OML-based IPS):** Takes immediate and precise actions to mitigate identified threats based on the information received from the intrusion detection system.

Our model is intelligently designed to continually adapt to evolving attack patterns through an online learning approach. Figure 3 offers a comprehensive overview of the entire DDoS detection and mitigation process, emphasizing the seamless integration of our system with SDN architecture.

System Workflow:

1) The traffic collector, connected to the SDN controller, efficiently copies network traffic from SDN edge switches. This process involves periodic requests for flow entries from all flow tables on OpenFlow switches, ensuring that each packet is switched only once.
2) To enhance security, these requests and responses are transmitted over a secure and isolated channel, preventing exposure to connected hosts. The speed of data collection significantly impacts threat detection and response times.
3) The OML-based IDS module plays a pivotal role in the detection process. It conducts traffic standardization, variable analysis, and employs trained machine learning models to classify preprocessed input traffic as either suspicious or benign.
4) Decisions made by the OML-based IDS are promptly communicated to the mitigation module, which employs the OML-based IPS strategy to take appropriate mitigation actions based on the information received from the IDS.
5) The ML-based IPS module translates these actions into flow rules, which are then implemented by the controller in network devices.

One of the strengths of our architecture lies in its modular design. Each module can be optimized independently, providing flexibility and adaptability. For instance, the IDS module can be replaced with alternative security products, such as new-generation firewalls. Additionally, while our experiments use the Ryu controller, our modular approach is controller-agnostic, as the OML-based IDS and OML-based IPS modules seamlessly integrate with the SDN controller through its northern interface (Rest API).

In the subsequent sections, we will offer detailed insights into the functionality and optimization of each module.

A. TRAFFIC COLLECTOR

The Traffic Collector module stands as a pivotal component within our DDoS detection system. While various network traffic flow generator and analysis tools like Flowba [38] and CICFlowMeter [39] are available, their compatibility with the OpenFlow protocol and SDN controllers varies. Some tools introduce delays or affect network traffic flow, hindering their seamless integration [40]. Moreover, certain tools primarily focus on offline processing, which does not align with our real-time packet processing approach. Consequently, we have meticulously amalgamated the strengths and advantages of these tools to craft our robust traffic collector module.

Our traffic collector module operates within the Ryu controller, implemented through a Python script. This module communicates with the controller's APIs to retrieve and mirror flow traffic information, capable of issuing periodic requests for flow statistics. The operation sequence of the traffic collector is as follows:

Authentication of all connected OpenFlow switches by generating a unique ID for each switch. Authentication ensures that hosts connected to the switch are permitted to exchange packets.

Packet headers are matched against flow entries in the switch's flow table upon packet arrival at an OpenFlow switch. If a successful match occurs, the entry's statistics are updated, encompassing information like the number of bytes and packets.

If a packet header fails to match any flow entry within the switch's flow table, it is forwarded to the controller. The controller, in response, can add a new flow entry to the switch's flow table to enforce the defined policy. This process ensures that traffic generated by all hosts connected to a specific OpenFlow switch populates the switch's flow table.

To construct the collecting module, we define a class known as ''DDOSMLApp,'' extending the RyuApp class from the ''app_manager'' module. The DDOSMLApp class serves as the foundation for implementing a Ryu application designed to monitor packet flow within a network. The class includes the ''OFP_VERSIONS'' class variable, specifying the OpenFlow protocol versions it is compatible with.

The ''init'' method, functioning as the constructor for the DDOSMLApp class, initializes essential class variables such as ''mac_to_port,'' ''datapaths,'' and ''mlobj.'' Furthermore, it initiates a new thread using the ''hub.spawn'' method. The ''monitor'' method operates within this new thread.

The ''monitor'' method is responsible for continuously observing the packet flow within the network. It operates in a loop, with intervals of sleep, and iterates through the datapaths. For each datapath, it sends an ''OFPFlowStatsRequest'' message, prompting the datapath to provide information about the current packet flow. The inclusion of print statements aids in outputting debugging messages to the controller.

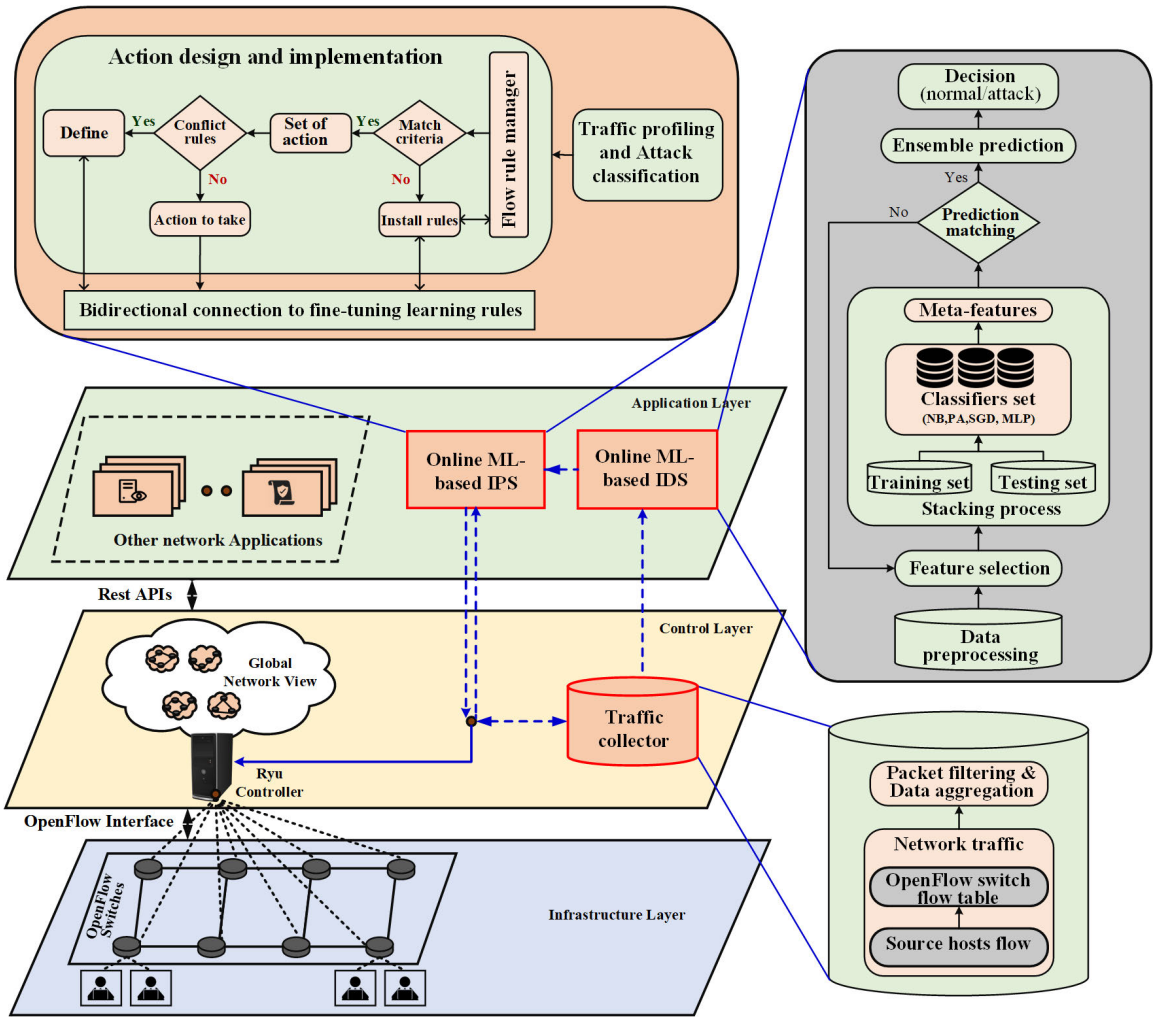

FIGURE 3. Overview of proposed system architecture.

Algorithm 1 outlines the pseudocode detailing the construction of the flow collector module, meticulously crafted to monitor and handle flow entries within OpenFlow switches.

B. OML-BASED IDS

The Online Machine Learning-Based Intrusion Detection System (OML-based IDS) has been designed and deployed specifically to detect evolving DDoS attacks. The OML-based IDS represents a promising solution, employing ensemble Machine Learning models that fuse multiple ML algorithms to assess network traffic in real-time. This dynamic analysis identifies patterns indicative of intrusions or malicious attacks.

The selection of BernoulliNB, Passive-Aggressive, SGD Classifier, and MLP Classifier for the ensemble model is based on several key criteria tailored to the demands of DDoS detection in SDN environments. Firstly, diversity in learning algorithms ensures that each classifier captures unique data facets, thereby enhancing the ensemble's overall predictive power. Secondly, computational efficiency is crucial, especially for real-time adaptation to streaming data. This necessitates classifiers with low computational overhead to minimize processing latency. Moreover, adaptability to concept drift and robustness to imbalanced data are essential attributes. These ensure that the model can effectively handle dynamic network conditions and accurately differentiate between attack and normal traffic instances.

To achieve this, we harness the capabilities of streaming machine learning, capitalizing on a specialized incremental library called "scikit-multiflow" [41]. This extension of the widely recognized "scikit-learn" [42] caters to multi-output/multi-label and stream data classification and regression tasks. "Scikit-multiflow" empowers us to seamlessly handle data streams, where instances continuously flow, whether one-by-one or in mini-batches [43]. This allows us to engage in incremental learning, dynamically updating our model in real-time as new data streams in. The library offers a spectrum of learning algorithms proficient in acquiring knowledge from incoming data, eliminating the need to reprocess previously encountered instances. Our model can continuously train and adapt to new network

Algorithm 1 Construction of the Flow Collector Module

1. Initialize DDOSMLApp_FlowCollector
2. AddFlow:
    a. Set priority, match, and actions
    b. Generate flow_id
    c. Parse match and actions
    d. Add flow to flows dictionary
    e. Send packet out message with flow_id
3. GetFlows:
    a. Return flows dictionary
4. UpdateFlowExpiration:
    a. If flow_expiration_timeout exists, set new flow_expiration_timeout to 30 seconds from now
    b. Else, set flow_expiration_timeout to current time
5. CollectFlows:
    a. If flow_expiration_timeout exists and current time is greater than or equal to flow_expiration_timeout:
        i. For each flow in flows dictionary:
        ii. If difference between current time and flow ['expiration_time'] is greater than or equal to 30 seconds:
        Remove flow from flows dictionary
    b. Update flow expiration timeout
6. Set connection and table_id arguments
7. Create a FlowCollector instance
8. Add a flow entry with priority, match, and actions
9. Collect flows every 30 seconds

traffic, enhancing its ability to detect novel threats, including zero-day DDoS attacks.

The key steps involved in constructing our OML-based IDS are as follows:

1. Collect and Preprocess Data: In this initial step, data is received from the traffic collector module. The data is cleaned and preprocessed to eliminate irrelevant or noisy information, rendering it suitable for our online ensemble ML model. This entails data cleaning, transformation into a format compatible with the model's algorithms/classifiers, and storage in the database and the data stream.

2. Feature Selection and Engineering: Next, we embark on feature selection and engineering. This phase involves analyzing the data to pinpoint relevant features and transforming them into a format suitable for the model's classifiers.

3. Model Initialization: Before we commence real-time predictions in the online stage, we initialize the model with initial data during an offline stage. This typically involves selecting a subset of existing datasets, such as CICIDS2019 dataset, InSDN, and slow-read-DDoS-attack-in-SDN, to train classifiers like BernoulliNB, Passive-Aggressive, SGD, and MLP as multiple base learners, each with its own set of hyperparameters.

4. Ensemble Learning: After the base learners are initialized, the model begins making predictions on new data as it arrives. The ensemble model is created by combining the predictions of these base learners through weighted combination, culminating in the final prediction.

5. Attack Detection: As the ensemble model generates predictions on incoming data, it can flag data points that appear anomalous or suspicious. This determination is typically based on the selected features, contrasting incoming data against the characteristics of what is considered normal behavior.

6. Model Monitoring and Updating: The final step involves keeping our model up to date. Over time, the accuracy of base learners may decline due to changing data distributions. To counteract this, we implement a drift detection technique to monitor the performance of each base learner over time, periodically updating them as new data streams in.

C. OML-BASED IPS

In a network's normal operation, real-time responsiveness plays a pivotal role as traffic necessitates continuous online monitoring and processing. The proposed OML-based IPS is engineered to avert DDoS attacks before they can inflict harm on the network or its resources. The distinctive feature of our OML-based IPS in DDoS prevention lies in its capacity to perpetually learn and adapt to shifting traffic patterns and evolving attack methods.

This adaptability empowers the IPS to proficiently identify and prevent DDoS attacks, even if assailants employ novel or unfamiliar tactics. It's crucial to emphasize that the IDS and IPS function in tandem, mutually reinforcing each other's effectiveness. The IDS detects the attack, and the IPS swiftly enacts the appropriate actions to thwart or mitigate it.

To robustly avert DDoS attacks, our model can:

1. Monitor Real-Time Network Traffic: The model scrutinizes network traffic in real-time, actively seeking unusual patterns or anomalies that might signify an attack.

2. Apply Machine Learning Algorithms: The system employs machine learning algorithms to scrutinize traffic patterns and identify potential threats, responding to notifications originating from the OML-based IDS.

3. Enforce Appropriate Countermeasures: Upon identifying potential threats, the IPS initiates suitable countermeasures, such as obstructing malicious traffic or diverting it to a designated sinkhole, thus forestalling the attack's success.

4. Continuously Learn and Adapt: The IPS is continually evolving, adjusting to shifting traffic patterns and emerging attack techniques, thus preserving its efficacy in detecting and thwarting attacks.

The architectural framework of the proposed OML-based IPS comprises four key phases:

1. Traffic Profiling: This phase involves analyzing and profiling network traffic in real-time to pinpoint unusual patterns or indications of an attack.

2. Action Mechanism Design: In this step, we design the appropriate actions and countermeasures to be employed upon detecting a threat.

3. Implementing the Action: The IPS executes the designated actions, such as blocking malicious traffic or redirecting it, to prevent the attack from succeeding.

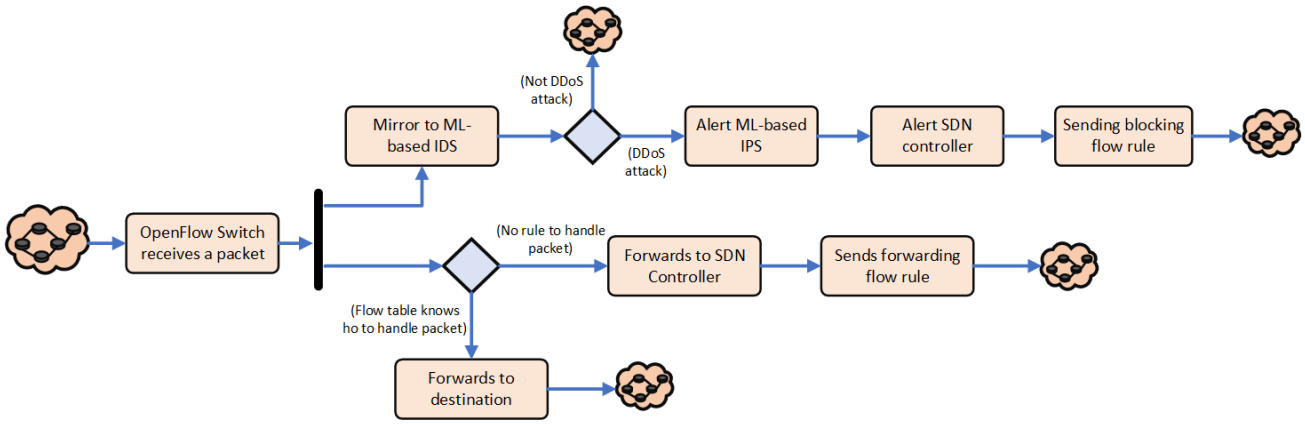

FIGURE 4. Workflow for processing each received packet.

4. Monitor and Update the Model: The IPS continuously monitors its performance and adapts over time to ensure its accuracy and effectiveness in thwarting evolving attacks.

The OML-based IDS and IPS collaborate harmoniously to ensure the network's security, with the IDS serving as the vigilant watchdog and the IPS swiftly taking action against any identified threats.

The novel aspects of the model architecture outlined in this section can be summarized as follows, complementing the streamlined flow processing depicted in Figure 4:

In the architecture workflow for processing each received packet, the traffic collector model operates within the Ryu controller framework using a Python script. This model interacts with the controller's APIs to gather or mirror flow traffic information and can initiate periodic requests to retrieve flow statistics. During the traffic collection phase, the initial step entails the Ryu controller authenticating all connected OpenFlow switches by assigning a unique ID to each switch. Only upon successful authentication are the connected hosts permitted to exchange packets.

As packets arrive at an OpenFlow switch, their headers undergo matching against the flow entries stored in the switch's flow table. In the event of a successful header match, the corresponding flow entry's statistics, such as byte and packet counts, are promptly updated. However, if no flow entry matches the packet's header, the packet is forwarded to the controller for further processing. A crucial facet of our model's architecture lies in its feature engineering methodology, which incorporates a novel selection process. This process aims to optimize detection efficacy while simultaneously reducing computational complexity. By carefully selecting pertinent features, our model achieves heightened efficiency and accuracy in identifying anomalous network behavior.

Moreover, our model incorporates robust mechanisms for handling concept drift, ensuring dynamic model updates based on streaming data. This adaptive approach enables continuous adaptation to changing network conditions, bolstering the model's resilience and efficacy against evolving threats. Together, these innovative architectural elements contribute to the model's effectiveness in real-time DDoS detection and mitigation, enhancing its adaptability and performance in dynamic network environments.

V. EXPERIMENT SETUP

This section outlines experiments assessing the proposed approach's effectiveness in detecting and mitigating DDoS attacks within SDN-based networks. It details the experiment methodology, covering traffic gathering, OML-based IDS, and OML-based IPS. The model is evaluated on various datasets, including a self-generated one with LDDoS/DDoS attacks. Using Mininet and Ruy SDN controller, experiments simulate DDoS attacks with tools like iperf, Hping3, and Scapy. The section also highlights ensemble model training to classify DDoS attacks and evaluates the model on benchmark datasets like CICIDS2019, InSDN, and slow-read-DDoS-attack-in-SDN, showcasing its robustness and real-world relevance.

A. EXPERIMENT ENVIRONMENT

The experiments were executed on a system with a 64-bit processor and 16 GB RAM running on the Windows 10 platform. Oracle's VirtualBox hosted Ubuntu 20.04 as the guest OS to establish the experimental environment. Mininet, supporting OpenFlow 1.3, was employed within this virtualized space, along with the Ryu SDN controller. MiniEdit facilitated the creation of virtual network topologies, and Wireshark was used for network traffic analysis. Table 3 summarizes the setup specifications, while Figure 5 visually represents the overall configuration.

VirtualBox ensured a controlled and reproducible simulated network environment. Mininet explored diverse networking configurations, and Ryu SDN offered efficient network management. Rigorous testing and evaluation of the proposed solution's performance were conducted within this experimental environment, affirming its effectiveness in SDN-based networks.

TABLE 3. Experiment environment specifications.

Component	Description
Processor	64-bit
RAM	16 GB
Operating System	Windows 10
Virtualization Software	Oracle's Virtual Box
Host Operating System	Ubuntu 20.04
Network Emulator	Mininet
SDN Controller	Ryu

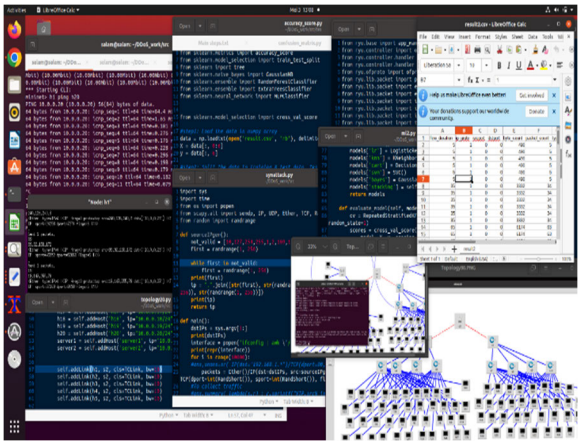

FIGURE 5. Overall experiment environment setup.

In a fat-tree network topology (depicted in Figure 6), the setup included a Ryu controller with two backbone switches (1Gbps) and eight side switches (100Mbps). All switches were interconnected for robustness. Within this setup, 80 emulated hosts were deployed, with specific roles assigned to generate benign or DDoS attack traffic. Python scripts were developed to introduce novel threats, aiding the observation of the model's adaptive response to evolving attack scenarios. Security modules were integrated into the Ryu controller, and a Python script automated experiment execution across various network scenarios. Key functions included topology creation, dataset generation, monitoring, and policy updates.

This comprehensive experimental setup provides the foundation for the subsequent sections' analysis of the proposed system's performance and effectiveness.

B. DATASETS AND TRAFFIC GENERATION FOR MODEL TRAINING

To assess the proposed ensemble model's efficacy, a classification of network traffic was performed using data from a simulated topology experiment depicted in Figure 6. The processed dataset encompasses 145,614 network traffic instances, with 61,881 representing abnormal traffic, accounting for approximately 40.4% of the total samples.

This dataset encapsulates a variety of low and high-rate DDoS attacks, alongside realistic normal traffic patterns

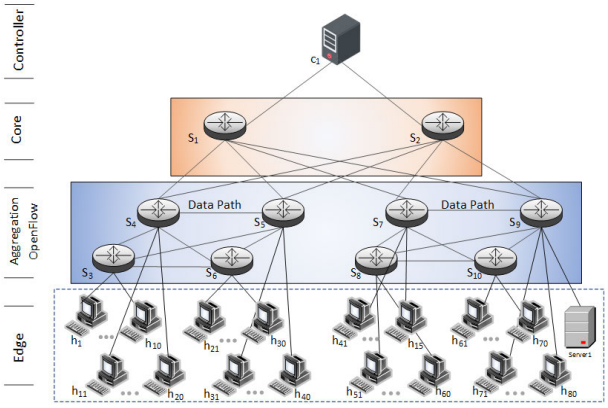

FIGURE 6. Customized fat-tree network topology scenario.

captured at 30-second intervals. Network traffic generation leveraged tools including iPerf, Scapy, and Hping3, in conjunction with the unique characteristics of the SDN architecture, facilitating the generation of network flows. These generated flows were calculated bidirectionally, with the direction forward/backward determined by the first packet in the flow.

The output of the generated flow comprises 22 statistical features in CSV file format, including Flow duration, IP proto, Number of bytes, Number of packets, SYN Flag Count, etc. These 22 collected features were further categorized into eight main groups, each serving distinct purposes in the characterization of network traffic:

1. Packet-Based Attributes: Including details about packets, such as the total number of packets in both forward and backward directions.

2. Network Identifiers Attributes: Encompassing common information defining the source and destination flow, such as IP addresses, port numbers, and protocol types.

3. Sub flow Descriptors Attributes: Presenting information specifically related to sub flows, such as the count of packets and bytes in both forwarding and backward directions.

4. Interarrival Time Attributes: Showcasing information concerning interarrival times in both forward and backward directions.

5. Bytes-Based Attributes: Relating to byte-specific data, encompassing the total number of bytes transmitted in both the forward and backward directions.

6. Flow Timers Attributes: Containing information regarding the duration of each flow, including active and inactive periods.

7. Flow Descriptors Attributes: Comprising traffic flow details, such as the count of packets and bytes in both forward and backward directions.

8. Flag Attributes: Encompassing information related to flags like SYN Flag, RST Flag, Push Flag, and others.

The comprehensive process employed for generating the dataset involved a series of detailed steps, meticulously outlined and executed to ensure accuracy and inclusivity of various network traffic features, thereby laying the groundwork

TABLE 4. Recorded features of the dataset.

Feature	Contents	Description
1	Flow duration	Duration of the flow
2	Ip proto	IP protocol used in the flow
3	Src port	Source port of the flow
4	Dst port	Destination port of the flow
5	Byte count	Total number of bytes transferred
6	Packet count	Total number of packets transferred
7	Total Fwd Packets	Total number of forward packets
8	Total Backward Packets	Total number of backward packets
9	Total Length of Fwd Pakts	Total length of forward packets
10	Total Length of Bwd Pakts	Total length of backward packets
11	Fwd Packet Length Max	Maximum length of forward packets
12	Fwd Packet Length Min	Minimum length of forward packets
13	Packet Length Variance	Variance of packet lengths
14	FIN Flag Count	Number of FIN flags observed
15	SYN Flag Count	Number of SYN flags observed
16	RST Flag Count	Number of RST flags observed
17	Average Packet Size	Average size of packets
18	Fwd Header Length	Length of forward header
19	Bwd Header Length	Length of backward header
20	Init Win bytes forward	Initial window size of forward direction
21	Init Win bytes backward	Initial window size of backward direction
22	Act data pkt fwd	Actual data packets in forward direction

for an exhaustive dataset reflective of the contemporary SDN environment. Following the meticulous outlining, the dataset generation process followed the steps outlined below:

1. Create different network topology scenarios using Mininet with Ryu controller, OpenFlow Switch, hosts, and servers and then send normal and malicious traffic among the network devices.

2. Create a Python file to collect the flow and port statistics for the duration of the monitoring interval.

3. Save the flow and port statistics in a CSV file.

4. Generate normal traffic using the iPerf tool (TCP, UDP, and ICMP traffic). Use Scapy and Hping3 tools to generate malicious traffic (low and high-rate DDoS).

5. Collect the flow statistics every 30 seconds. The SDN Controller sends Openflow flow stats request message to all the switches, and the switches reply with flow stats.

6. Collect a total of 22 features in the dataset extracted from the controller and the network traffic.

7. The last column in the dataset indicates the class label, whether the traffic type is normal (0) or malicious (1).

8. Run the network simulation for 2 hours, generating about 2,900,000 instances (124,000 rows and 24 columns in the CSV file).

The dataset was generated through various experiment scenarios. It includes a combination of normal and malicious network traffic, with normal traffic generated using iPerf and Ping, and attack traffic generated using Scapy and Hping3. Table 4 outlines the recorded features during the dataset generation process.

To validate the model's generalization ability, a comprehensive evaluation was conducted using benchmark datasets, including CICIDS2019, InSDN, and slow-read-DDoS-attack-in-SDN datasets. Table 5 serves as a comprehensive

TABLE 5. Comparison of our dataset with existing datasets.

Dataset	Year	Attack tools	Attack instances	Normal instances	Number of attributes
CICIDS 2019	2019	GoldenEye, LOIC, slowhttptest, HULK	272,0000	1,328,000	83
Slow-DDoS	2022	Slowhttptest	276,000	10,000	12
InSDN	2020	Hping3, LOIC, slowhttptest, HULK, and Nping	275,515	68,424	83
Own Dataset	2023	Scapy, iPerf and Hping3	61,881	78,733	22

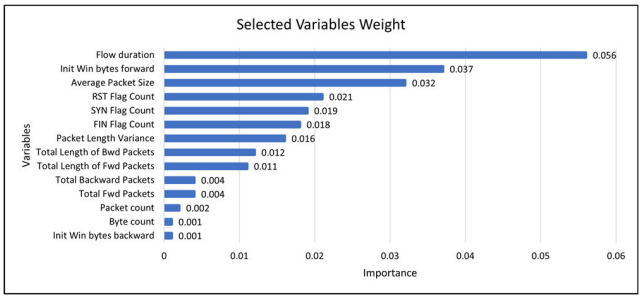

FIGURE 7. Selected features by the proposed feature selection method.

representation of the comparison between the meticulously curated dataset and existing datasets that have been employed in related works. The strategic incorporation of both the proprietary dataset and pre-existing datasets serves to enrich and augment the scope of attack scenarios used to train and evaluate the detection model. This synergy introduces a heightened level of complexity and authenticity into the training data, crucial for ensuring the effectiveness of the approach across a multitude of potential real-world scenarios.

To enhance the effectiveness of our generated dataset, we employed a feature selection process. Our proposed method refined the Chi2 feature selection as a filter method, reducing the dimensionality of the dataset by selecting only the most informative features. This reduction, from 22 to 14 features, led to remarkable accuracy rates and helped mitigate overfitting, where the model learns irrelevant patterns from the training data. Focusing on the most informative features allows the model to generalize better to unseen data. Figure 7 illustrates the selected features. The Chi-squared statistic was computed for each feature to measure the disparity between observed and expected counts. Comparing this statistic against a critical value helped determine the significance of the relationship between features and DDoS attacks. The prioritized features, based on their Chi-squared values, formed a curated subset seamlessly integrated into the proposed ensemble model, thereby enhancing the efficacy of DDoS attack detection.

C. ASSESSMENT OF THE OML-BASED IDS

The OML-based IDS model, designed for online training and testing using streaming data, undergoes a thorough evaluation process. The dataset comprises both normal and malicious

network traffic, divided into training and testing sets at a 70:30 ratio. Employed classifiers include Online Stacking Ensemble, BernoulliNB, Passive-Aggressive, SGD, and MLP Classifiers.

The training and testing procedure involves importing the dataset, splitting it into training and test sets, converting data into streams, creating classifiers, initializing models, testing in an online fashion, and plotting results. Performance metrics like accuracy, precision, recall, and F1 score are calculated for each classifier. The steps are detailed below:

1) IMPORTING THE DATASET

- The required libraries are imported, and the source of the dataset is mounted from Google Drive to access the data.
- The dataset is read using the pandas library and stored in a DataFrame named "df".

2) SEPARATING TRAINING AND TEST DATASETS

- The dataset is split into input features (X) and the target variable (y).
- Further division is done to create separate training and test sets using appropriate indexing.

3) CONVERTING TRAIN AND TEST DATA INTO STREAMS

- The train and test sets are converted into data streams using the DataStream class.
- The prepare_for_use() method is called to initialize and prepare the data streams.

4) MODEL CREATION

- Four different classifiers, namely nb_classifier, PA_ classifier, sgd_classifier, mlp_classifier, voting_classifier, and ensemble (Online Stacking), are initialized.

5) INITIALIZING THE MODEL ON INSTANCES

- The next sample of instances is fetched from the training data stream using the next_sample() method.
- Each classifier is fitted or trained using the obtained training data.

6) TESTING THE MODEL IN AN ONLINE FASHION AND RECORDING PERFORMANCES

- A loop is set up to iterate over chunks of test data. The next sample of test data is fetched from the test data stream.
- Predictions are made using each classifier on the test data, and the results are stored. Performance metrics such as accuracy, precision, recall, specificity, F1 score, and confusion matrix are calculated for each classifier.
- The partial_fit() method is called to update the model with the new test data.
- The counts of class labels (zeros and ones) and the imbalance ratio are stored for each chunk.

7) PLOTTING RESULTS

- The performance results of the different classifiers are plotted using the matplotlib library.

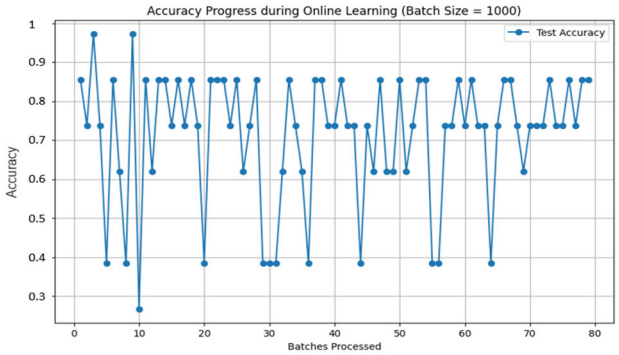

FIGURE 8. Validation accuracy progress during online learning.

- The plotted results provide insights into the performance comparison among the classifiers

This methodology enables the training and testing of the OML-based IDS model using streaming data, evaluating classifiers' performance with various metrics. The visualized accuracy progression in Figure 8 demonstrates the model's adaptability to evolving traffic patterns in real-time.The stability and improvement of accuracy over time signify the model's proficiency, especially in handling unknown traffic representing new DDoS attacks. This online learning approach ensures continuous adaptation to the dynamic nature of network traffic, making the model a potent defense against emerging cybersecurity threats.

D. ASSESSMENT OF OML-BASED IPS

This section assesses the OML-based IPS, examining its performance, effectiveness, and adaptability against various DDoS attacks, especially in real-time scenarios. By combining real-time intrusion prevention and online machine learning, the OML-based IPS establishes a dynamic defense mechanism that efficiently counters DDoS threats. This mechanism also incorporates strategies for legitimate communication recovery, ensuring scalability in fluctuating network traffic. At its core, the system continuously scrutinizes incoming network traffic, adeptly identifying potential anomalies and intrusions. In response, it promptly initiates measures to mitigate risks, safeguarding the network from potential threats.

The OML-based IPS combines real-time intrusion prevention and machine learning for a strong defense against DDoS attacks, adapting dynamically for network stability and security. The assessment relies on key metrics from carefully orchestrated experiments, offering insights into the system's efficacy in mitigating DDoS attacks. Subsequent sections explore the details of this evaluation, revealing how the synergy of real-time intrusion prevention and machine learning creates a resilient shield against evolving DDoS threats.

1) ATTACK STRATEGY AND TESTS' DESCRIPTION

DDoS attacks involve coordinated assaults from various locations in the network, targeting a single victim and exhausting

TABLE 6. Normal network performance without DDoS attack.

Test Id.	Host 1 - Average RTT (ms)	Host 1 - Packet Loss (%)	Host 71 - Average RTT (ms)	Host 71 - Packet Loss (%)
1	0.588	0	0.439	0
2	0.756	0	1.274	0
3	0.493	0	0.463	0
4	0.766	0	0.904	0
5	2.004	0	0.402	0
Average	0.922	0	0.696	0

its resources, leading to the denial of service for legitimate hosts. In the provided scenario (Figure 6), a single controller manages 10 switches and 80 hosts using the OpenFlow protocol. The attack strategy simulates network activities and scenarios, configuring 24 hosts as attackers out of 80, while the remaining 56 hosts are assigned as legitimate.

To gauge the system's response, benign traffic is generated using iPerf and Ping commands, and high-rate DDoS attacks are simulated with Hping3 and Scapy tools. The experiment duration is 5 minutes, with traffic collected every 30 seconds.

The IPgen function is employed to intensify the attack, altering the source IP address within a specified range. The attack targets a specific server with the IP address 10.0.0.23.

Verification of network connectivity and exploration of normal behavior involve exchanging packets using Pingall, iPerf, and Ping commands. Additionally, OpenFlow table rules are excluded, showcasing the typical traffic trend before DDoS attacks launch. Normal network performance metrics, without DDoS attacks, are quantitatively measured, providing a baseline for comparison.

2) NORMAL NETWORK PERFORMANCE WITHOUT DDoS ATTACK

Table 6 presents the latency of Quality of Service (QoS) metrics during normal network performance, showing average Round Trip Time (RTT) and packet loss percentage for each host during ping tests. The values indicate stable network conditions without DDoS attacks. These results serve as a benchmark for assessing network performance without DDoS attacks.

During the DDoS attack, when the proposed Defense System was disabled, significant differences in results were observed. Figure 9 illustrates the impact on the OpenFlow Switch's Rule Table and network traffic behavior, showcasing the surge in traffic during the DDoS attack. The Defense Model's absence allowed for the injection of malicious traffic, leading to an unmanageable peak traffic rate near 50,000 packets per second, rendering the server unreachable for legitimate users.

This comprehensive evaluation provides valuable insights into the OML-based IPS's efficacy in handling DDoS attacks, emphasizing its adaptive nature and real-time response capabilities. The subsequent analysis further dissects the model's accuracy during online learning, showcasing its resilience and proficiency in addressing emerging DDoS attacks.

VI. RESULTS AND DISCUSSION

To evaluate the ensemble model's performance, which incorporates four distinct OML algorithms on the acquired dataset, we employ several primary performance indicators: accuracy, precision, recall, F1 score, and false alarm rate. These metrics rely on values derived from true positives, true negatives, false positives, and false negatives. Each machine learning technique possesses unique characteristics for learning, predicting, and evaluating data points to classify and detect attacks based on applied tuning parameters. The aim of our proposed model is to achieve high accuracy, precision, recall, and F1-measure while maintaining a low false alarm rate. Accuracy is computed using the formula:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

Here, TP represents correctly classified malicious flows, TN denotes correctly classified normal flows, FN refers to incorrectly classified normal flows, and FP indicates incorrectly classified attacking flows.

Precision is calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall is determined by:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

The F1-score is derived from:

$$F1 - score = \frac{2 * recall * precision}{recall + precision} \tag{4}$$

The false alarm rate (FAR) is computed as:

$$FAR = \frac{FP}{(FP + TN)} \tag{5}$$

A. EVALUATION METRICS OF THE DETECTION PHASE

1) PREDICTION ACCURACY

Evaluating the predictive accuracy of the ensemble model, which amalgamates outputs from four distinct machine learning algorithms (BernoulliNB, Passive-Aggressive, SGD Classifier, and MLP Classifier), involves scrutinizing the individual classifier accuracies and emphasizing the supreme accuracy achieved by the ensemble. These evaluations were conducted on the meticulously crafted dataset. Figure 10 delineates the performance metrics of individual classifiers and the ensemble. This experiment rigorously scrutinized the ensemble model and its constituent classifiers using a purposely diversified dataset, thoroughly testing their capacity to generalize.

The ensemble model showcased remarkable performance, surpassing the individual classifiers with an accuracy of 0.9926. This substantiates the efficacy of amalgamating diverse classifiers to forge a more precise predictive system. The visual depiction in Figure 10 effectively illustrates each classifier's accuracy, offering a comprehensive comparative

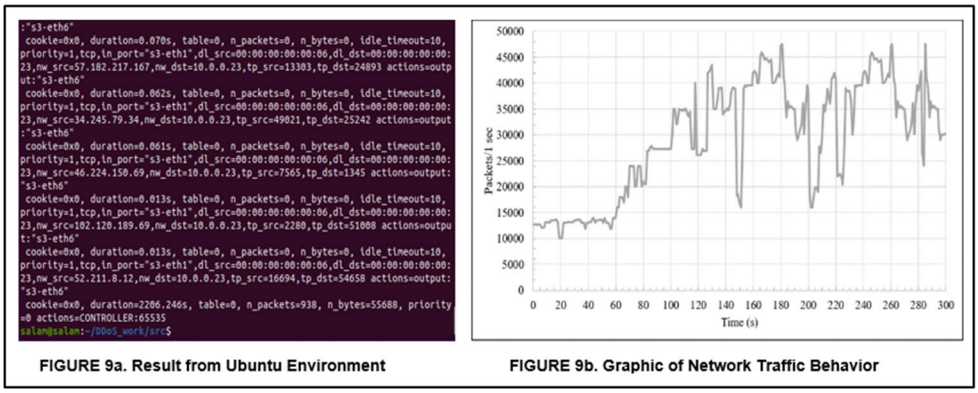

FIGURE 9a. Result from Ubuntu Environment

FIGURE 9b. Graphic of Network Traffic Behavior

FIGURE 9. OpenFlow switch's rule table and network traffic before DDoS attacks.

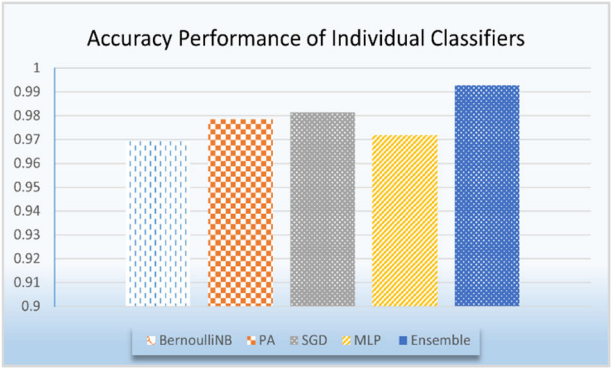

FIGURE 10. Accuracy performance of individual classifiers and the ensemble.

FIGURE 11. Precision performance of individual classifiers and the ensemble.

analysis of their performances. It unequivocally demonstrates the consistent superiority of the ensemble model over individual classifiers, underscoring the benefits of harnessing multiple algorithms for predictive tasks. This graphical representation substantiates the ensemble's exceptional performance and fortifies its applicability in practical real-world scenarios.

These discoveries offer invaluable insights into the distinctive strengths and limitations of each classifier and accentuate how the ensemble approach can harness their collective prowess to yield more accurate predictions. These results underscore the ensemble model's potential deployment across various domains, including but not limited to finance, healthcare, and diverse data-driven applications.

2) PRECISION

Precision stands as a crucial metric unveiling the accuracy of anomaly predictions within the model, denoting the ratio of true positive predictions to all positive predictions. This section presents precision results for both individual classifiers and the ensemble model. Figure 11 visually represents the precision of individual classifiers alongside the ensemble.

The ensemble model exhibits remarkable precision, registering an outstanding value of 0.9910, surpassing individual classifiers. This underscores the ensemble's capability to effectively pinpoint anomalies and yield accurate positive predictions. The graphical representation in Figure 5,6 succinctly showcases precision values for each classifier, allowing a lucid comparison of their performances. Evidently, the ensemble consistently maintains high precision, further validating its proficiency in anomaly detection. These precision results provide insightful evidence regarding the models' capacity to minimize false positives and ensure reliable predictions. Precision plays a pivotal role in domains such as fraud detection, medical diagnosis, and quality control, ensuring precise decision-making and risk mitigation. Overall, the ensemble model's impressive precision underscores its potential as a dependable and robust tool across various domains demanding precision-centric tasks.

3) RECALL

Recall serves as a pivotal metric that gauges the accuracy of DDoS attack detection within the model, evaluating its capacity to accurately identify anomalies among positive examples. This section presents the recall results for both individual classifiers and the ensemble model. Figure 12 provides a visual representation of Recall performance for individual classifiers and the ensemble.

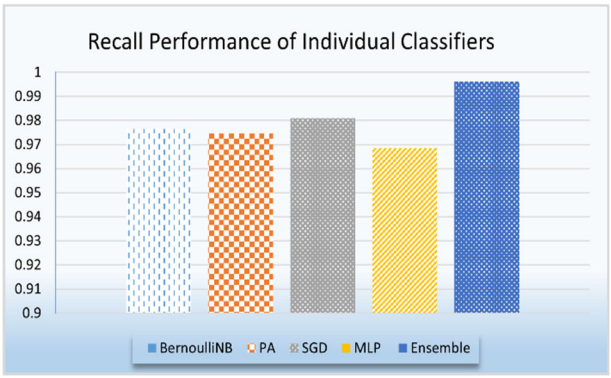

FIGURE 12. F1 Score performance of individual classifiers and the ensemble.

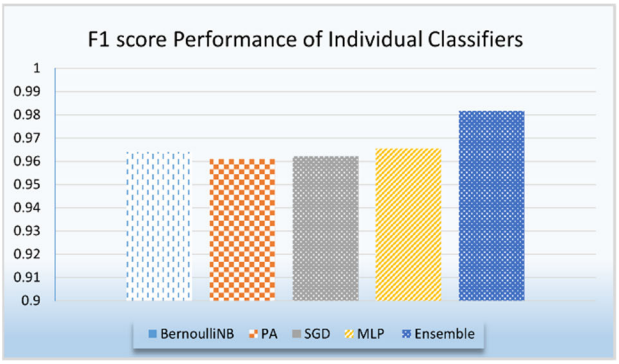

FIGURE 13. Recall performance of individual classifiers and the ensemble.

The ensemble model showcases exceptional recall, achieving an impressive value of 0.9962, surpassing the individual classifiers. This underscores the ensemble's remarkable proficiency in accurately detecting DDoS attacks and reducing false negatives. These recall results offer valuable insights into the models' effectiveness in capturing true positive examples while minimizing the likelihood of missing genuine anomalies. In domains such as intrusion detection and network security, recall stands as a critical measure of the model's capability to accurately identify significant events. Overall, the ensemble model's outstanding recall underscores its potential as a robust and reliable tool in scenarios where precise anomaly detection is pivotal for upholding system integrity and security.

4) F1 SCORE

The F1 score stands as a pivotal metric that consolidates precision and recall, offering a singular value that signifies the overall effectiveness of the model in anomaly detection. It assesses the model's capability to strike a balance between minimizing false positives (precision) and minimizing false negatives (recall). This section presents the F1 score results for both individual classifiers and the ensemble model. Figure 13 visually illustrates the F1 Score performance of individual classifiers and the consolidated ensemble.

The ensemble model achieves an exceptional F1 score of 0.9817, indicating its proficiency in striking a balance between precision and recall, effectively detecting anomalies. Figure 13 provides a graphical representation of the F1 score values for each classifier, offering a clear comparison of their performance. The consistently high F1 scores exhibited by the ensemble further substantiate its prowess in anomaly detection and classification. The F1 score is an indispensable metric as it comprehensively evaluates the overall model performance by considering both false positives and false negatives.

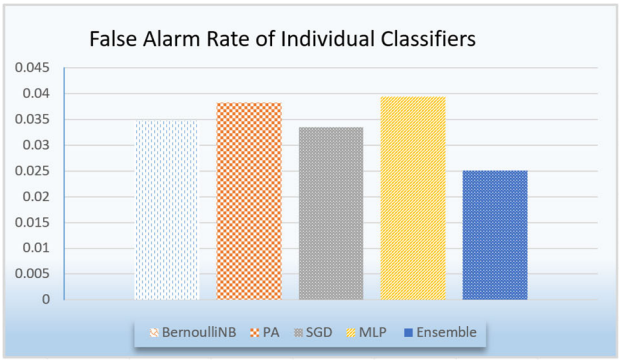

FIGURE 14. False alarm rate of individual classifiers and the ensemble.

5) FALSE POSITIVE

False positives play a crucial role in assessing the effectiveness of a DDoS detection model, serving as a key metric to gauge its performance. This metric evaluates the model's ability to maintain a balance between accurately identifying legitimate traffic and erroneously flagging it as malicious. In this section, we present the false positive results for both individual classifiers and the ensemble model, shedding light on their respective performances.

The ensemble model demonstrates exceptional proficiency in false positive management, achieving a remarkably low value of 0.025. This indicates the model's effectiveness in minimizing false alarms while accurately detecting DDoS attacks. To provide a visual comparison of the false positive rates across different classifiers, Figure 14 depicts the F1 score values for each classifier, offering insights into their relative performances and highlighting the ensemble model's superiority in false positive mitigation.

6) EVALUATION ON EXISTING DATASETS

In addition to evaluating the proposed model's performance on the custom dataset, rigorous assessments were conducted across three well-established benchmark datasets: CICIDS2019, InSDN, and slow-read-DDoS-attack-in-SDN datasets. These evaluations serve as robust validations, affirming the versatility and effectiveness of the proposed model across a spectrum of diverse datasets.

a: CICDDoS2019 DATASET ANALYSIS

The CICDDoS2019 dataset, a cornerstone in cybersecurity research facilitated by the Canadian Institute for Cybersecurity, offers a comprehensive repository encompassing both

DDoS attack instances and benign network traffic samples. Noteworthy for its extensive compilation of network traffic data, it provides an array of rich features surpassing the scale and breadth of existing datasets. With 88 distinctive characteristics derived from flow-based features and covering 12 diverse types of DDoS attacks, this dataset significantly enhances the value for robust DDoS attack detection and comprehensive analysis.

To evaluate the proposed model's performance on this dataset, several metrics, including accuracy, precision, recall, and F1-score, were computed. The meticulous evaluation produced exceptional performance metrics: accuracy at 98.70%, precision at 98.78%, recall at 98.81%, and an F1-score of 98.78%. These high-performance results underscore the model's proficiency in effectively identifying and categorizing DDoS attacks within this specific dataset.

b: InSDN DATASET ANALYSIS

The InSDN dataset serves as a critical asset tailored for the analysis of intrusion detection mechanisms within SDN environments. Specifically curated to study network intrusions, it encompasses a wide spectrum of network traffic data originating from an SDN-based network. Similar methodologies as employed with the CICDDoS2019 dataset were implemented to attain evaluation results for the InSDN dataset.

The method applied to the InSDN dataset yielded significant performance metrics: an accuracy of 98.20%, precision at 97.51%, recall of 97.93%, and an F1-score of 98.27%. These statistics are a testament to the model's proficiency in effectively analyzing the InSDN dataset.

c: SLOW-READ-DDoS-ATTACK-IN-SDN DATASET ANALYSIS

The slow-read-DDoS-attack-in-SDN dataset is purposefully crafted to delve into the intricate realm of slow-read DDoS attacks within SDN environments. It encompasses a variety of slow-read DDoS attack types, such as Slowloris, Slowhttptest, and Hulk. As the pioneering publicly available dataset devoted explicitly to investigating slow-read DDoS attacks in SDN landscapes, it stands as an invaluable asset for researchers devoted to scrutinizing detection mechanisms within SDN environments.

Similar methodologies, akin to those employed in obtaining the CICDDoS2019 and InSDN datasets, were adopted to acquire results from this dataset. The dataset was utilized to both train and evaluate the model, culminating in a notable performance: an accuracy score of 98.88%, precision measured at 96.80%, recall reaching 95.90%, and an F1-score of 96.27%.

d: SUMMARY OF THE EVALUATION RESULTS ON THE EXISTING DATASETS

The evaluation outcomes garnered from a thorough assessment across multiple datasets, including CICIDS2019, InSDN, slow-read-DDoS-attack-in-SDN, and the custom dataset, unequivocally depict outstanding performance metrics, notably high accuracy, precision, recall, and F1-scores.

TABLE 7. Performance of proposed method on different datasets.

Dataset	Accuracy	Precision	Recall	F1-Score	False positive
CICIDS2019	98.70%	99.78%	98.81%	98.78%	18.5%
InSDN	98.20%	97.51%	97.93%	98.27%	18%
Slow-read-DDoS-attack	98.88%	96.80%	95.90%	96.27%	3.65%
Custom dataset	99.26%	99.10%	99.60%	98.17%	2.25%

These results vividly underscore the method's resilience and adaptability across varied scenarios.

A summary of the comprehensive evaluation results for these datasets is presented in Table 7. The results demonstrate the reliability and robustness of the proposed model across diverse datasets, reinforcing its potential for effectively detecting and mitigating DDoS attacks in SDN environments.

B. EVALUATION METRICS OF THE MITIGATION PHASE

Effective evaluation metrics are pivotal in assessing the performance of a mitigation model and understanding the impact of DDoS attacks. This study meticulously selected commonly used evaluation metrics to comprehensively evaluate the proposed model's effectiveness and analyze the effects of DDoS attacks:

Response Time: Quantifying the duration taken to respond to a request, this metric provides insight into the system's responsiveness.

Resource Consumption Analysis: This metric measures the workload executed by the processor and indicates the amount of memory consumed by the system. Understanding resource utilization is vital for optimizing performance.

Effect of a DDoS attack on Legitimate Users: This refers to how the attack impacts regular users or clients trying to access network resources or services.

Through the careful selection of these evaluation metrics, valuable insights are gained into the performance of the mitigation system, allowing for a comprehensive understanding of the defense model's efficacy and aiding in informed decisions to enhance network resilience against DDoS attacks.

1) RESPONSE TIME

This section explores the latency aspects of the OML-based IPS by examining its response time during attack handling. Response time serves as a measure of latency, indicating the duration between the initiation of an attack and the activation of the ''Drop'' action by the OML-based IPS. The results, outlined in Table 8, highlight the system's efficient performance, with an average response time of approximately five seconds across multiple tests. This swift mitigation response ensures heightened protection against potential DDoS attacks, underscoring the solution's effectiveness.

TABLE 8. Average response time result.

Tests	Response Time (s)	Average RTT (ms)	Packet Loss (%)
1	**3**	**0.619**	**0**
2	**4**	**0.547**	**0**
3	**5**	**0.561**	**0**
4	**5**	**0.484**	**0**
5	**7**	**0.767**	**0**
Average	**4.8**	**0.596**	**0**

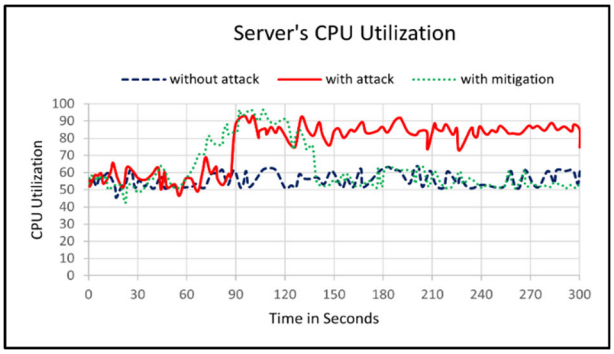

FIGURE 15. Server CPU utilization before, during and after the attack.

The consistently efficient performance observed, with response times consistently below ten seconds, underscores the model's proficiency in swiftly mitigating DDoS attacks, thereby minimizing network disruption during mitigation procedures.

Our experimental findings shed light on both the latency and throughput of our model in processing network traffic and responding to DDoS attacks. By analyzing metrics like response time and packet processing rates, we gain insights into the model's real-time detection and mitigation capabilities. Furthermore, we can explore optimization strategies to reduce latency, such as deploying efficient data structures, implementing parallel processing techniques, and optimizing algorithmic complexity. Similarly, throughput enhancements may involve optimizing network communication protocols, improving parallelism, or harnessing hardware acceleration techniques.

2) RESOURCE CONSUMPTION ANALYSIS

Assessing resource consumption involved the use of Docker containers within MININET to emulate CPU utilization. Figures 15 and 16 present the temporal evolution of resource consumption for both the Ryu controller and the target server. Noteworthy observations include:

Baseline CPU utilization equilibrium between the controller and the server before the attack.

Swift escalation in CPU utilization during the attack, peaking at 68% for the controller and 85% for the server.

Post-mitigation, the controller stabilizes at approximately 28%, and the server maintains an average utilization of around 60%.

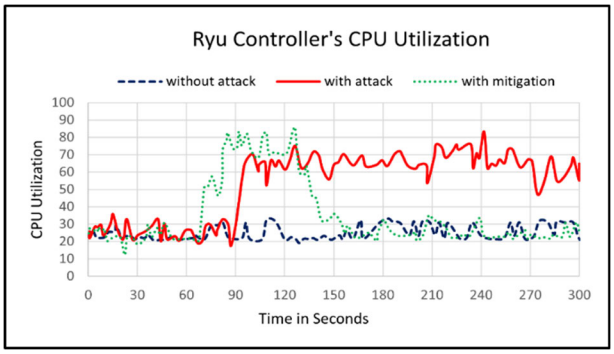

FIGURE 16. RYU controller CPU utilization before, during and after the attack.

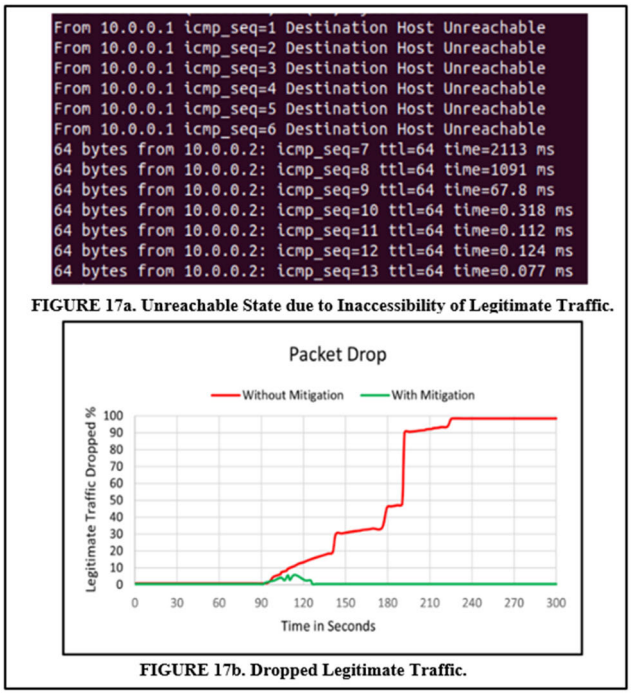

FIGURE 17. Orchestrated responses for legitimate connections.

The discernible correlation between baseline utilization levels and those during mitigation underscores minimal performance overhead during the proposed model's flow rule execution.

This analysis delves into the intricate interplay between DDoS attacks and resource consumption, highlighting the mitigation strategy's prowess in efficiently managing and stabilizing resource utilization.

3) MITIGATING PACKET DROP (IMPACT OF DDOS ATTACK ON LEGITIMATE USER)

The impact of a DDoS attack on legitimate users is a critical aspect addressed by the proposed model. Figure 16 illustrates the orchestrated responses for legitimate connections, emphasizing the IPS's capability in ensuring minimal disruption to legitimate users.

TABLE 9. Comparison of the proposed solution with commercial alternatives.

Feature	Proposed Model	Arbor Peakflow	Radware DefensePro	F5 Silverline
Detection Techniques	Online ensemble learning, dynamic adaptation	Signature-based, anomaly-based	Signature-based, behavioral analysis	Cloud-based, global scrubbing centers
Adaptability	Real-time adaptation to emerging threats	Limited adaptability to zero-day attacks	Real-time policy updates	Scalable cloud-based solution
Performance	Real-time response to streaming data	Effective for known attack signatures	Real-time mitigation	Global scrubbing for high scalability
Scalability	Handles large volumes of streaming data	Scalable for large networks	May require hardware investments	Scalable cloud-based solution
Latency	Minimal latency for rapid response	Low latency for detection	Fast mitigation response	Potential latency issues for geographically dispersed networks
Cost-effectiveness	Efficient resource utilization	Costly for large-scale deployments	Costly for hardware investments	Cost-effective cloud-based solution

A marginal impact on legitimate connection drops, accounting for 0.02, reaffirming the IPS's success in preserving the integrity of legitimate user interactions. Swift reinstatement of inadvertently impeded legitimate connections, typically within an 8-second timeframe, ensuring minimal disruptions for legitimate users. This swift and efficient response time, coupled with minimal disruption to legitimate users, highlights the model's success in maintaining a high-quality user experience even in the face of potential threats.

4) INDUSTRY RECOMMENDATIONS AND COMPARISONS WITH COMMERCIAL SOLUTIONS

In the dominion of DDoS detection and mitigation, our solution stands as a contender alongside established commercial systems. Drawing from our empirical findings and industry benchmarks, we meticulously evaluate our approach against prevalent commercial solutions, aiming to elucidate its strengths in adaptability, real-time responsiveness, and cost-effectiveness. Through this comparative analysis, we endeavor to shed light on the competitive edge and potential market adoption of our model.

Arbor Networks Peakflow [44], exemplifies robust capabilities in DDoS detection and mitigation, leveraging a combination of signature-based and anomaly-based techniques. Widely recognized for its comprehensive protection against diverse attack vectors, Peakflow is a stalwart presence across expansive networks. However, its effectiveness may falter in dynamic attack scenarios, where rapid evolution and sophistication pose challenges to signature-based detection methods [45].

In contrast, Radware DefensePro [46], offers real-time DDoS protection through a blend of signature-based and behavioral analysis techniques [47]. Renowned for its ability to counter both volumetric and application-layer attacks, DefensePro provides granular control over mitigation policies. Nonetheless, concerns linger regarding its scalability for high-traffic networks and the potential need for substantial hardware investments in large-scale deployments [48].

F5 Silverline DDoS Protection [49], presents a cloud-based mitigation service equipped with global scrubbing centers, delivering scalable and efficient defense mechanisms against DDoS assaults. Leveraging F5's expertise in application delivery, Silverline seamlessly integrates with existing solutions. Nevertheless, challenges may arise concerning latency in geographically dispersed networks and potential cost barriers for smaller organizations [50].

In addition to qualitative evaluations, Table 9 offers a comprehensive comparison that concisely outlines the strengths and weaknesses of each solution across diverse features and capabilities. This comparison serves as a valuable resource for stakeholders, enabling them to make informed decisions that align with their specific requirements and constraints.

VII. CONCLUSION AND FUTURE WORK

The proposed model, consisting of the traffic collector, OML-based IDS, and OML-based IPS, offers a modular and scalable framework for effective DDoS attack detection and mitigation in SDN networks. This architecture excels in addressing both low-rate and high-rate incidents, showcasing remarkable adaptability. The OML-based IDS demonstrates exceptional performance in detecting various DDoS attack types, achieving detection and legitimate rates exceeding 99%. Integration with the OML-based IPS further enhances capabilities in addressing the entire spectrum of DDoS attacks within SDN networks. In contrast to prior approaches, this solution exhibits superior adaptability to unfamiliar traffic patterns, covering diverse DDoS attack variations.

While recognizing the need for multiple layers of protection, the modular design allows independent improvements in various components, ensuring adaptability. Compatibility with various SDN controllers reinforces the model's versatility. The proposed approach presents a real-time solution for detecting and mitigating DDoS attacks in SDN-based networks. Future research directions include validating the model with real-world network topologies, exploring deep learning models, conducting live environment testing, and extending capabilities to cloud-based SDN environments for comprehensive protection in hybrid network infrastructures.

REFERENCES

[1] P. Goransson, C. Black, and T. Culver, *Software Defined Networks: A Comprehensive Approach*. San Mateo, CA, USA: Morgan Kaufmann, 2016.

[2] J. Singh and S. Behal, "Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions," *Comput. Sci. Rev.*, vol. 37, Aug. 2020, Art. no. 100279.
[3] A. A. Alashhab, M. S. M. Zahid, A. A. Barka, and A. M. Albaboh, "Experimenting and evaluating the impact of DoS attacks on different SDN controllers," in *Proc. IEEE 1st Int. Maghreb Meeting Conf. Sci. Techn. Autom. Control Comput. Eng. (MI-STA)*, May 2021, pp. 722–727.
[4] M. P. Singh and A. Bhandari, "New-flow based DDoS attacks in SDN: Taxonomy, rationales, and research challenges," *Comput. Commun.*, vol. 154, pp. 509–527, Mar. 2020.
[5] M. Chhabra and B. B. Gupta, "An efficient scheme to prevent DDoS flooding attacks in mobile ad-hoc network (MANET)," *Res. J. Appl. Sci., Eng. Technol.*, vol. 7, no. 10, pp. 2033–2039, Mar. 2014.
[6] A. Mishra, N. Gupta, and B. B. Gupta, "Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller," *Telecommun. Syst.*, vol. 77, no. 1, pp. 47–62, May 2021.
[7] Y. Al-Dunainawi, B. R. Al-Kaseem, and H. S. Al-Raweshidy, "Optimized artificial intelligence model for DDoS detection in SDN environment," *IEEE Access*, vol. 11, pp. 106733–106748, 2023.
[8] Q. Li, H. Huang, R. Li, J. Lv, Z. Yuan, L. Ma, Y. Han, and Y. Jiang, "A comprehensive survey on DDoS defense systems: New trends and challenges," *Comput. Netw.*, vol. 233, Sep. 2023, Art. no. 109895.
[9] A. Bhardwaj, V. Mangat, R. Vig, S. Halder, and M. Conti, "Distributed denial of service attacks in cloud: State-of-the-art of scientific and commercial solutions," *Comput. Sci. Rev.*, vol. 39, Feb. 2021, Art. no. 100332.
[10] A. A. Alashhab, M. S. M. Zahid, M. Abdullahi, and M. S. Rahman, "Real-time detection of low-rate DDoS attacks in SDN-based networks using online machine learning model," in *Proc. 7th Cyber Secur. Netw. Conf. (CSNet)*, Oct. 2023, pp. 95–101.
[11] I. D. Mienye and Y. Sun, "A survey of ensemble learning: Concepts, algorithms, applications, and prospects," *IEEE Access*, vol. 10, pp. 99129–99149, 2022.
[12] O. Sagi and L. Rokach, "Ensemble learning: A survey," *WIREs Data Mining Knowl. Discovery*, vol. 8, no. 4, p. e1249, Jul. 2018.
[13] M. Pirizadeh, N. Alemohammad, M. Manthouri, and M. Pirizadeh, "A new machine learning ensemble model for class imbalance problem of screening enhanced oil recovery methods," *J. Petroleum Sci. Eng.*, vol. 198, Mar. 2021, Art. no. 108214.
[14] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with Naive Bayes—Which Naive Bayes?" in *Proc. CEAS*, Mountain View, CA, USA, 2006, pp. 28–69.
[15] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, Y. Singer, and M. K. Warmuth, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.*, vol. 7, no. 3, pp. 551–585, 2006.
[16] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Statist.* Paris, France: Springer, 2010, pp. 177–186.
[17] T. Windeatt, "Accuracy/diversity and ensemble MLP classifier design," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1194–1211, Sep. 2006.
[18] S. Xu, X.-W. Wang, and M. Huang, "Software-defined next-generation satellite networks: Architecture, challenges, and solutions," *IEEE Access*, vol. 6, pp. 4027–4041, 2018.
[19] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.
[20] N. Bizanis and F. A. Kuipers, "SDN and virtualization solutions for the Internet of Things: A survey," *IEEE Access*, vol. 4, pp. 5591–5606, 2016.
[21] A. A. Alashhab, M. S. M. Zahid, A. Muneer, and M. Abdukkahi, "Low-rate DDoS attack detection using deep learning for SDN-enabled IoT networks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 11, pp. 371–377, 2022.
[22] V. D. M. Rios, P. R. M. Inácio, D. Magoni, and M. M. Freire, "Detection and mitigation of low-rate denial-of-service attacks: A survey," *IEEE Access*, vol. 10, pp. 76648–76668, 2022.
[23] T. Lukaseder, L. Maile, B. Erb, and F. Kargl, "SDN-assisted network-based mitigation of slow DDoS attacks," in *Proc. 14th Int. Conf. Secur. Privacy Commun. Netw.* Singapore: Springer, Aug. 2018, pp. 102–121.
[24] N. Muraleedharan and B. Janet, "Behaviour analysis of HTTP based slow denial of service attack," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2017, pp. 1851–1856.
[25] A. A. Alashhab, M. S. M. Zahid, M. A. Azim, M. Y. Daha, B. Isyaku, and S. Ali, "A survey of low rate DDoS detection techniques based on machine learning in software-defined networks," *Symmetry*, vol. 14, no. 8, p. 1563, Jul. 2022.
[26] M. Roopak, S. Parkinson, G. Y. Tian, Y. Ran, S. Khan, and B. Chandrasekaran, Jan. 14, 2024, "An unsupervised approach for the detection of zero-day DDoS attacks in IoT networks," *Authorea*, doi: 10.22541/au.170526630.07302484/v1.
[27] K. Sornalakshmi, "Detection of DoS attack and zero day threat with SIEM," in *Proc. Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Jun. 2017, pp. 1–7.
[28] A. A. Alashhab, M. S. M. Zahid, M. Alashhab, and S. Alashhab, "Online machine learning approach to detect and mitigate low-rate DDoS attacks in SDN-based networks," in *Proc. IEEE Int. Conf. Artif. Intell. Eng. Technol. (IICAIET)*, Sep. 2023, pp. 152–157.
[29] M. A. Ribeiro, M. S. P. Fonseca, and J. de Santi, "Detecting and mitigating DDoS attacks with moving target defense approach based on automated flow classification in SDN networks," *Comput. Secur.*, vol. 134, Nov. 2023, Art. no. 103462.
[30] Ö. Tonkal, H. Polat, E. Başaran, Z. Cömert, and R. Kocaoğlu, "Machine learning approach equipped with neighbourhood component analysis for DDoS attack detection in software-defined networking," *Electronics*, vol. 10, no. 11, p. 1227, May 2021.
[31] V. Deepa, K. M. Sudar, and P. Deepalakshmi, "Design of ensemble learning methods for DDoS detection in SDN environment," in *Proc. Int. Conf. Vis. Towards Emerg. Trends Commun. Netw. (ViTECoN)*, Mar. 2019, pp. 1–6.
[32] J. A. Pérez-Díaz, I. A. Valdovinos, K. R. Choo, and D. Zhu, "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning," *IEEE Access*, vol. 8, pp. 155859–155872, 2020.
[33] K. T. R. Khamkar, A. Kotkar, P. Jadhav, and R. Hanchate, "Low rate DDoS attack identification and defense using SDN based on machine learning method," *Int. Res. J. Eng. Technol.*, vol. 8, no. 3, pp. 174–178, 2021.
[34] K. M. Sudar and P. Deepalakshmi, "Flow-based detection and mitigation of low-rate DDOS attack in SDN environment using machine learning techniques," in *IoT and Analytics for Sensor Networks*. Hyderabad, India: Springer, 2022, pp. 193–205.
[35] R. Polikar, "Ensemble learning," in *Ensemble Machine Learning: Methods and Applications*. New York, NY, USA: Springer, 2012, pp. 1–34.
[36] M. A. Ganaie, M. Hu, A. Malik, M. Tanveer, and P. Suganthan, "Ensemble deep learning: A review," *Eng. Appl. Artif. Intell.*, vol. 115, Sep. 2022, Art. no. 105151.
[37] Y. Ren, L. Zhang, and P. N. Suganthan, "Ensemble classification and regression-recent developments, applications and future directions," *IEEE Comput. Intell. Mag.*, vol. 11, no. 1, pp. 41–53, Feb. 2016.
[38] A. Ray, P. K. Bala, and N. P. Rana, "Exploring the drivers of customers' brand attitudes of online travel agency services: A text-mining based approach," *J. Bus. Res.*, vol. 128, pp. 391–404, May 2021.
[39] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, nos. 1–4, pp. 131–156, 1997.
[40] O. Yevsieieva and S. M. Helalat, "Analysis of the impact of the slow HTTP DOS and DDOS attacks on the cloud environment," in *Proc. 4th Int. Scientific-Practical Conf. Problems Infocommunications. Sci. Technol. (PIC S&T)*, Oct. 2017, pp. 519–523.
[41] 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, Mar. 2013.
[42] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, Dec. 1997.
[43] K. K. Kamarajugadda and T. R. Polipalli, "Stride towards aging problem in face recognition by applying hybrid local feature descriptors," *Evolving Syst.*, vol. 10, no. 4, pp. 689–705, Dec. 2019.
[44] S. Y. Lim and A. Jones, "Network anomaly detection system: The state of art of network behaviour analysis," in *Proc. Int. Conf. Converg. Hybrid Inf. Technol.*, Aug. 2008, pp. 459–465.
[45] C. Rodriquez, "The expanding role of service providers in DDoS mitigation," in *Stratecast Perspectives and Insight for Executives (SPIE)*, vol. 15. CA, USA: Frost & Sullivan, 2015, pp. 1–10.
[46] N. Dayal, P. Maity, S. Srivastava, and R. Khondoker, "Research trends in security and DDoS in SDN," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 6386–6411, Dec. 2016.

[47] L. Malina, P. Dzurenda, and J. Hajny, "Testing of DDoS protection solutions," in *Security and Protection of Information*. CA, USA: Internet Archive, 2015, pp. 113–128.
[48] J. Smith-Perrone and J. Sims, "Securing cloud, SDN and large data network environments from emerging DDoS attacks," in *Proc. 7th Int. Conf. Cloud Comput., Data Sci. Eng.*, Jan. 2017, pp. 466–469.
[49] J. D'Hoinne, A. Hils, and C. Neiva, "Magic quadrant for web application firewalls," Gartner, Stamford, CT, USA, Tech. Rep. 1, 2014.
[50] M. S. Merkow, *Practical Security for Agile and DevOps*. London, U.K.: Auerbach Publications, 2022.

ABDUSSALAM AHMED ALASHHAB received the B.Sc. degree in computer networks from the Faculty of Information Technology, University of Tripoli, Libya, in 2014, and the M.Sc. degree (Hons.) in computer networks from the Limkokwing University of Creative Technology, Malaysia, in 2017. He is currently pursuing the Ph.D. degree with the Computer and Information Sciences Department, Universiti Teknologi PETRONAS (UTP), Malaysia. He is also a Lecturer with Alasmarya Islamic University, Libya. His research interests include cyber security, software defined networking, and machine learning techniques.

MOHD SOPERI ZAHID (Member, IEEE) received the Ph.D. degree in computer science from the University of Wisconsin–Milwaukee, in 2009. He is currently an Associate Professor with the Computer and Information Sciences Department, Universiti Teknologi PETRONAS, Malaysia. Prior to that, he was a Faculty Member of the Faculty of Computing, Universiti Teknologi Malaysia. He has published papers in numerous journals and conference proceedings. His main research interests include computer network failure recovery and machine learning for software defined network security and healthcare.

BABANGIDA ISYAKU received the B.Sc. degree in computer science and information system from Oxford Brookes University, in 2012, and the M.Sc. and Ph.D. degrees in computer science from Universiti Teknologi Malaysia (UTM), in 2017 and 2022, respectively. He is currently with Sule Lamido University, Kafin Hausa, Jigawa, Nigeria. He is also a Researcher with Universiti Teknologi Malaysia, under the Postdoctoral Fellowship Scheme. His research interests include software defined networks, routing, failure recovery, and flowtable management. He was a recipient of the Best Paper Award at the IEEE Symposium on Computer Applications and Industrial Electronics, in 2020, and the Best Postgraduate Student Award from the Faculty of Computing, UTM.

ASMA ABBAS ELNOUR received the B.Sc. degree in computer science from Omdurman Ahlia University, Sudan, the M.Sc. degree in computer science and technology from Gezira University, Sudan, and the Ph.D. degree in computer science (artificial intelligence) from the University for Science and Technology. She is currently an Assistant Professor with the Department of Information Systems, Applied College, King Khalid University, Muhayil, Saudi Arabia. She has been the Head of the Information System Department, since 2018. Her research interests include information security emerging technologies and the Internet of Things (IoT).

WAMDA NAGMELDIN received the B.Sc. degree in computer science from the Faculty of Computer Studies, International University of Africa, Sudan, in 2004, the M.Sc. degree in computer science from the Faculty of Mathematical Science, University of Khartoum, Sudan, in 2007, and the Ph.D. degree in computer science from the University of Technology Malaysia, Malaysia, in 2020. She is currently a Lecturer with the Department of Information Systems, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Saudi Arabia. Her research interests include cryptography, cyber security, and cloud computing.

ABDELZAHIR ABDELMABOUD received the M.Sc. degree in computer science and information from Gezira University, Sudan, and the Ph.D. degree in software engineering from Universiti Teknologi Malaysia. He is currently an Associate Researcher with the Humanities Research Centre, Sultan Qaboos University. Previously, he is an Associate Professor with the Department of Information Systems, College of Science and Arts at Mohayil, King Khalid University, Saudi Arabia. He has held many positions, including the IT Manager, the Quality Manager, and a Database Administrator. He is also a member of the University of Technology Malaysia's Software Engineering Research Group. His research interests include cyber security and blockchain technologies built on the Internet of Things and cloud computing.

TALAL ALI AHMED ABDULLAH received the B.Sc. degree in computer science from the Faculty of Science, Taiz University, Yemen, in 2014, and the M.Sc. degree in information technology from the Kulliyyah of Information and Communication Technology, International Islamic University Malaysia (IIUM), Malaysia, in 2019. He is currently pursuing the Ph.D. degree with the Computer and Information Sciences Department, Universiti Teknologi PETRONAS (UTP), Malaysia. His research interest includes machine learning techniques and their applications.

UMAR DANJUMA MAIWADA received the B.Sc. degree in computer science from Bayero University, Kano, and the M.Sc. degree in computer science from Jodhpur National University, India. He is currently pursuing the Ph.D. degree with the CIS Department, Universiti Teknologi PETRONAS (UTP). He is also with Umaru Musa Yar'adua University, Katsina, as a Lecturer. He has more than 20 publications, including conferences and journal articles. His research interests include computer networking, programming with C++, data science, communication, digital twin's networks, and the IoT. He is a reviewer in some journals.