International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 6, December 2025, pp. 5484~5496
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i6.pp5484-5496

Flow-guided long short-term memory with adaptive directional learning for robust distributed denial of service attack detection in software-defined networking

Huda Mohammed Ibadi, Asghar A. Asgharian Sardroud
Department of Electrical and Computer Engineering, Urmia University, Urmia, Iran

Article Info

Article history:

Received Mar 14, 2025
Revised Jul 27, 2025
Accepted Sep 16, 2025

Keywords:

Adaptive directional learning
DDoS attack detection
Flow direction algorithm
Long short-term memory
Software defined networking

ABSTRACT

A software-defined networking (SDN) architecture is designed to improve network agility by decoupling the control and data planes, but while much more flexible, also makes networks more vulnerable to threats, such as distributed denial of service (DDoS) attacks. In this study we present a novel detection model, the flow-guided long short-term memory (LSTM) network with adaptive directional learning (ADL), for the mitigation of DDoS attacks in software defined networking (SDN) environments. While the methodology is based on a flow direction algorithm (FDA), which analyzes traffic patterns and detects anomalies from directional flow behavior. The proposed method integrates FDA in LSTM-based threat detection frameworks within internet of things (IoT) networks, thereby yielding enhanced detection accuracy, as well as a real-time security threat response. The experimental evaluation on two benchmark datasets, namely the InSDN dataset and a real-time dataset utilizing a Mininet and POX controller setup, shows that a detection rate of 99.85% and 99.72%, respectively, thereby showcasing the proposed model's ability to differentiate between legitimate and malicious network traffic.

Corresponding Author:

Asghar A. Asgharian Sardroud
Department of Electrical and Computer Engineering, Urmia University
Urmia 5756151818, Iran
Email: a.asgharian@urmia.ac.ir

1. INTRODUCTION

Software-defined networking (SDN) is a transformative paradigm that redefines traditional network management by decoupling the control plane from the data plane. This separation offers unprecedented flexibility, centralized orchestration, and programmability [1]. However, this very flexibility also introduces new security vulnerabilities, particularly the risk of distributed denial of service (DDoS) attacks targeting the SDN controller [2]. These attacks can exhaust the controller's resources, disrupt packet forwarding, and lead to complete service denial across the network. As SDN becomes a cornerstone in modern enterprise and internet of things (IoT) infrastructures, securing it against such dynamic and large-scale attacks has become a critical research concern [3]. Conventional signature-based intrusion detection systems (IDS) are often ineffective in SDN environments due to their reliance on predefined patterns and inability to adapt to novel or evolving threats [4], [5]. To address these limitations, recent research has explored the use of machine learning (ML) and deep learning (DL) techniques for anomaly detection [6], [7]. Notably, bidirectional long short-term memory (BiLSTM) networks [8], transformer-based architectures such as DDoSViT [9], and hybrid SDN-integrated systems like SNORT-SDN [10] have demonstrated promising detection performance. However, these models still face critical challenges. They are often static in nature, suffer from high false

positive rates, and fail to dynamically adapt to changes in traffic patterns or attack behaviors—limitations that can compromise their effectiveness in real-time SDN deployments [11].

To address the above issues, in this paper, we introduce a new hybrid detection framework, flow-guided long short-term memory (LSTM) with adaptive directional learning (ADL). The model integrates three major constituents such as the flow direction algorithm (FDA) that analyzes the bidirectional flow anomalies, LSTM network to capture the sequential dependencies in the network traffic, and an adaptive dynamic learning mechanism (ADL) that dynamically adjusts its learning parameters according to the varying attack landscapes. By combining FDA and ADA combined with LSTM, our approach not only enhances the accuracy of the detection, but also improves the resistance to zero-day attacks and false positive, which are crucial for a SDN-based security system. This approach differs from previous ones in which directional flow characteristics and adaptive learning dynamics are modelled explicitly. While most existing methods only consider the temporal or spatial aspect, our model combines these two and introduces dynamic adjustment of the threshold through ADL. The system is benchmarked using the well-established InSDN benchmark dataset and a real-time Mininet-based SDN scenario. The findings indicate 99.85% detection performance, which exceeds existing state-of-the-art models including DDoSNet [12], SNORT-SDN [10] and DDoSViT [9], thereby validating the effectiveness of our FDA–LSTM–ADL fusion for securing SDN infrastructures. The main contributions of this paper are threefold. It introduces a new hybrid IDS architecture, which bridges the gap between FDA-driven directional flow analysis and LSTM-based sequence modeling and the ADL-based flexibility. Second, it reports an empirical study based on benchmark datasets and real-time devised datasets to verify the generalizability and robustness of the model. Third, comparison against existing state-of-the-art DDoS detection mechanisms demonstrates the superiority and effectiveness of the proposed framework in terms of accuracy, adaptivity, and low false alarm rate.

The rest of the paper is structured as follows: section 2 discusses the related works while section 3 introduces the deep learning-based DDoS detection algorithm based on counterfactual reasoning and hopes. The proposed FDA-LSTM-ADL is described in section 3, listing formulations of the algorithms and the model architectures. Section 4 presents the experimental setup, the datasets, the attack model, the evaluation methodology. Section 5 introduces the results and discusses them, followed by conclusions and outlooks in section 6.

2. RELATED WORK

The growing acceptance of SDN has led to a burgeoning concern regarding its safety, especially in the context of DDoS attacks. In SDN setups, DDoS attacks aim at several network layers at once, with each layer presenting a distinct set of problems to solve [13]. At the data plane, for example, one kind of attack involves saturating the interfaces between the SDN controllers and the network devices [14]. This is known as an attack on the Southbound interface [15]. Another kind of attack involves flooding the network devices with so much traffic that they cannot handle it and, as a result, they start dropping packets and create a traffic jam in the network. And still another kind of attack aims at the flow tables in the SDN switches themselves [16].

At the control plane, the packet-in flooding attack sends excessive message traffic to the SDN controller, straining its already limited capacity and at times even causing it to totally lose its ability to serve legitimate requests. When this happens, service disruption may well be underway, and the SDN will not be able to perform any of its controller functions. Except for the worst-case scenario, the amount of disruption served up in the control plane by packet-in flooding is, in fact, quite capable of serving similar amounts of disruption that other performative denial-of-service (PDoS) attacks do in traditional networked systems [17].

Blocking attacks have a secondary line of targeted victims on the application side, but service denial is not their only aim. They are also meant to increase the opportunity for an attacker to perform a data exfiltration operation. Table 1 summarizes these attacks and their impacts on the four layers of the SDN [18].

Different techniques have been suggested to reduce the impact of DDoS attacks. One technique is to use in-network defense mechanisms, which are quite different from traditional defense mechanisms that are located on the perimeter of the protected network [19]. In-network defense mechanisms require active participation from each network switch and have been shown to allow low-rate DDoS attack traffic through while blocking the high-rate DDoS traffic at the perimeter. This is helpful in reducing the E/E factor that is critical to the DDoS attack from succeeding [20].

Other methods merge techniques that are based on entropy with models that result from deep learning to enable the detection and mitigation of DDoS attacks that target SDN controllers [21]. These models perform calculations of network entropy to enable the identification of anomalous traffic and make use of machine learning techniques, such as bidirectional long short-term memory (Bi-LSTM), to enhance the accuracy of detection [8]. In addition, approaches that are based on the calculation of Renyi entropy have been explored to capture anomalies in network flow and enable the extraction of key traffic features from SDN flow tables, which are then used to identify behaviors that are malicious in nature [22].

Table 1. Summary of DDoS attacks targeting different layers of SDN

Layer	Attack type	Description	Impact
Data plane	Southbound interface (SBI) attack	Disrupts communication between the SDN controller and network devices.	Network disruption and control loss.
	Buffer saturation attack	Floods network devices with excessive traffic, causing packet loss and congestion.	Packet loss and device overload.
	Flow meter overflow	Overwhelms the flow table with excessive entries, causing network disruptions.	Resource exhaustion and flow table saturation.
Control plane	Packet-in flooding	Sends excessive message traffic to the SDN controller, overwhelming its processing capacity.	Service disruption due to controller overload.
	Data channel blocking	Blocks communication channels between the controller and devices.	Reduces effective communication.
Application plane	Northbound interface (NBI) attack	Exploits vulnerabilities in the communication between the SDN controller and applications.	Resource consumption and performance degradation.
	Application resource attack	Overuses computational resources, degrading the overall performance of the application plane.	System resource exhaustion.

Models based on machine learning and deep learning have also been put to use for DDoS detection. Support vector machines (SVM) and deep neural networks (DNN) are applied often to classify the patterns of network traffic and to distinguish between traffic that is normal and that which is harmful, with a real-time detection capability being a major focus of such efforts [23]. Despite some promising results, these models are still facing challenges, namely scalability and adaptability, especially when they are up against evolving attack strategies and complex network traffic [11]. To sum up, although substantial advancements have been achieved in crafting DDoS detection systems specifically for SDN, the very nature of SDN—dynamic and ever-changing—demands that detection models be adapted and enhanced continuously to effectively deal with the attack techniques that are diverse and numerous.

3. PROPOSED METHOD

This section describes the architecture, algorithms, and procedures used in the design and implementation of the proposed intrusion detection model, which integrates FDA, LSTM, and ADL to enhance the detection of DDoS attacks in SDN environments. The goal is to ensure high detection accuracy, low false positives, and adaptability to evolving attack strategies [24]. The methodology consists of three core modules: flow-based feature extraction, sequential analysis, and adaptive learning. This section also outlines the implementation pipeline in detail to ensure reproducibility. A detailed workflow of the model is illustrated in Figure 1.

Figure 1 illustrates the overall workflow of the proposed hybrid intrusion detection model. It shows how traffic flows are first processed through the FDA for feature extraction, then analyzed by an LSTM network enhanced with ADL. The output is classified as benign or malicious behavior based on the learned temporal and directional patterns.

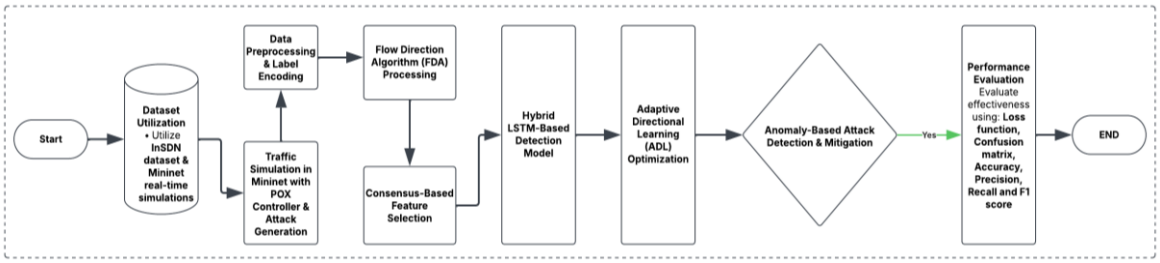


Figure 1. Workflow of the proposed hybrid LSTM-based detection model

3.1. Flow direction analysis using FDA

The FDA is a novel mechanism designed to extract directional characteristics from SDN traffic flows. Unlike conventional statistical or flow-based methods, FDA focuses on bidirectional patterns by analyzing each flow's initiation, sequence behavior, termination, and communication consistency. The features extracted by FDA include:

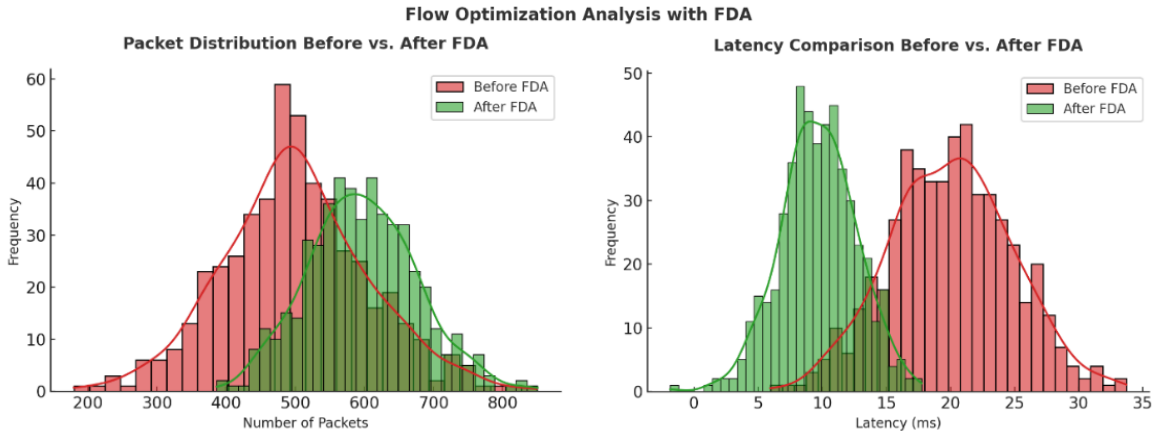
a. Flow initiation frequency: detects unusual spikes in the number of new connections.

b. Packet sequence variation: flags reordering or irregular packet flows suggestive of spoofing.
c. Flow termination anomalies: captures abnormal disconnections or timeouts.
d. Source-destination consistency: identifies mismatches in expected communication paths.

The logic of FDA is presented in Algorithm 1, which describes the step-by-step extraction of these features from raw traffic flows. Once computed, these features form the input vectors for subsequent sequential modeling. As illustrated in Figure 2, the application of FDA results in improved flow uniformity and reduced latency, which supports better anomaly visibility and classification. Figure 2 demonstrates the effectiveness of the FDA in optimizing network traffic. In Figure 2(a), it shows how packet flow distribution becomes more balanced after FDA is applied, reducing traffic spikes. Figure 2(b) highlights the reduction in latency, illustrating improved network responsiveness. Together, they validate FDA's role in enhancing flow consistency and anomaly detection efficiency.

Algorithm 1. Flow direction analysis using FDA

```
Input: Traffic_Flows (network packets), FDA (Flow Direction Algorithm)
Output: Flow_Characteristics (extracted bidirectional traffic features)
1. Initialize Flow_Characteristics ← ∅
2. For each flow F in Traffic_Flows do:
      a. Extract Flow_Initiation_Frequency(F)
      b. Compute Packet_Sequence_Variation(F)
      c. Identify Flow_Termination_Anomalies(F)
      d. Assess Source_Destination_Consistency(F)
3. Store extracted Flow_Characteristics
4. Apply FDA to analyze bidirectional traffic patterns
5. Return Flow_Characteristics
```



(a) (b)

Figure 2. Flow-optimization analysis with (FDA) visualization: (a) packet distribution before vs. after FDA and (b) latency comparison before vs, after FDA

3.2. ADL

To achieve the proposed scheme, in the processing stage, ADL is introduced to improve the adaptability of the detection model of rail surface defect, which always adjust the internal parameters in real-time. ADL does not need static models, while adapting to concept drift, new attack patterns, and traffic behaviors, so that the underlying logic can be adjusted automatically.

The ADL performs its work based on three main missions. It does first, observe pre-pattern errors by tracking a sliding window of predicate errors and identifying when the model starts to misclassify data because of changing traffic pattern. Second, it selectively updates weights in the LSTM layers with mini-batch gradient descent, thereby allowing lightweight re-training without resetting the complete model. Third, it adopts live traffic feedback to calibrate decision thresholds so that the detection sensitivity and the false positive rate can be controlled.

The formal process of doing distributed ADL is described in Algorithm 2, which specifies the ways we identify new patterns, recompute thresholds, and sometimes fine-tune the model weights. It helps in retaining the robustness of the model to fast evolving network conditions.

Algorithm 1. ADL optimization

```
Input: LSTM_Model (trained model), Attack_Patterns (new DDoS variations)
Output: Optimized_LSTM_Model
1. Watch for unnoticed assault patterns in currently available network traffic.
2. In the event that New_Attack_Patterns are detected:
a. Modify LSTM_Weights through the use of gradient updates.
b. LSTM_Model with updated dataset to retrain.
c. Use cross-entropy loss to validate performance.
3. Keep on with adaptive learning until convergence is reached.
4. Provide the Optimized LSTM Model as output.
```

3.3. LSTM-based sequential learning

LSTM networks are employed to model the temporal progression of traffic flows. Given that network behavior often evolves over time, LSTM is an ideal architecture for identifying long-term dependencies and deviations from normal sequence patterns. The LSTM-based architecture begins with an input layer that ingests FDA-derived features. This is followed by a hidden LSTM layer comprising 64 units, capable of encoding flow dynamics over time. A dropout layer (with a rate of 0.2) is used to reduce overfitting. The output layer employs a SoftMax activation function to classify inputs as either benign or attack traffic.

Training is conducted using the Adam optimizer with a learning rate of 0.001 and a batch size of 64, over 50 epochs. Categorical cross-entropy is used as the loss function due to its suitability for binary classification tasks. The overall sequence of operations, from FDA preprocessing to LSTM classification, is encapsulated in Algorithm 3, which outlines the data transformation, training, and inference steps used to detect anomalies based on sequential learning.

Algorithm 3. Hybrid LSTM-based anomaly detection

```
Input: FD_Features (Flow Direction selected features), Model (CNN+LSTM)
Output: Predicted_Labels (normal or attack classification)
1. Preprocess FD_Features for neural network input
2. Pass FD_Features through CNN_Layer to extract spatial dependencies
3. Feed CNN output into LSTM_Layer to learn sequential relationships
4. Apply Softmax_Activation to obtain classification probabilities
5. Assign Predicted_Labels based on highest probability class
6. Return Predicted_Labels
```

3.4. Data preprocessing and feature engineering

Preprocessing is a critical stage that prepares raw data for model ingestion. The process involves:

a. Feature extraction: Selecting key attributes such as source/destination IP, port numbers, protocol type, byte/packet count, and flow duration.
b. Normalization: Applying Min-Max scaling to standardize feature ranges.
c. Label encoding: Assigning numerical values to class labels (0 for benign, 1 for attack).
d. Class balancing: Utilizing SMOTE for minority oversampling and random under sampling to handle class imbalance and prevent model bias.

This ensures that input data is consistent, noise-reduced, and appropriately structured for neural network training.

3.5. Implementation environment and reproducibility

The model is implemented in Python using the TensorFlow framework. Simulations are executed in a virtual SDN environment using Mininet v2.3.0 and the POX controller. DDoS attack scenarios—including TCP SYN, UDP flood, and ICMP flood—are generated using hping3 and LOIC tools. Experiments are conducted on a system equipped with an Intel Core i9-12900K processor, 32GB RAM, and an NVIDIA RTX 3090 GPU. Traffic is captured using Wireshark, and OpenFlow statistics are used to validate anomaly detection. To ensure reproducibility, the full methodology is supported by pseudocode for all key algorithms (Algorithms 1–3), publicly available datasets (InSDN and Mininet-generated traffic), and a detailed record of hyperparameter settings and training conditions.

4. EXPERIMENTAL SETUP AND DATA COLLECTION

To evaluate the effectiveness of the proposed flow-guided LSTM model with ADL, a comprehensive experimental framework was developed, combining benchmark datasets and real-time simulation environments. This section outlines the datasets used, attack simulation strategies, preprocessing procedures, model training configurations, and evaluation settings to ensure transparency and reproducibility.

4.1. Data collection

To evaluate the effectiveness and generalizability of the proposed FDA–LSTM–ADL framework, two distinct datasets were utilized: a standardized benchmark dataset and a custom real-time dataset. The first dataset, known as the InSDN dataset, is a publicly available benchmark specifically curated for SDN-based intrusion detection research. It contains well-labeled traffic samples representing both normal and malicious network behaviors, including a wide range of DDoS attack scenarios such as TCP floods, UDP floods, and ICMP-based attacks [25]. This dataset serves as a baseline for comparative evaluation against existing detection models.

The second dataset was custom-generated using Mininet, a network emulator that simulates real-time SDN environments. Leveraging the POX controller and programmable switch topology, diverse traffic patterns were captured under both benign and adversarial conditions, including dynamically injected DDoS attacks. This real-time dataset allows for a practical assessment of the model's adaptability and robustness in dynamic network conditions, where traffic flows and controller responses evolve over time [26]. By employing both benchmarked and real-time datasets, the study ensures a comprehensive evaluation of the proposed model's performance across varying conditions and traffic complexities.

Table 2 presents a comparison of the two datasets used for training and evaluating the proposed DDoS detection model: the InSDN benchmark dataset and a custom real-time dataset generated in Mininet. The table lists key features extracted from both datasets, such as flow ID, source and destination IP addresses, port numbers, protocol types (TCP, UDP, ICMP), packet and byte counts, flow duration, and traffic type labels. Each feature is marked as present (✓) in both datasets, confirming that the experimental setup maintains consistency in the features extracted across synthetic and real-world network conditions. This uniformity is critical for evaluating the model's generalization ability. The table thus validates that both datasets are rich and well-structured, supporting accurate and consistent model training and evaluation.

Table 2. Summary of datasets used for DDoS detection

Feature	Description	InSDN dataset	Mininet dataset
Flow ID	Unique identifier for each network flow	☑	☑
Source IP	IP address of the source node	☑	☑
Destination IP	IP address of the destination node	☑	☑
Source Port	Port number used at the source	☑	☑
Destination Port	Port number used at the destination	☑	☑
Protocol	Protocol type (TCP, UDP, ICMP)	☑	☑
Packet Count	Number of packets transmitted in a flow	☑	☑
Byte Count	Total bytes transmitted per flow	☑	☑
Flow Duration	Total duration of the network flow	☑	☑
Traffic Type	Labeled as Normal or DDoS Attack	☑	☑

4.2. Attack simulation in SDN environment

To replicate realistic DDoS conditions, three major attack types were emulated:

a. TCP SYN flooding: High-volume SYN packets were generated using hping3, targeting the SDN controller to exhaust its resources.
b. UDP flooding: Random UDP packets were directed toward switch ports, saturating network links and inducing packet drops.
c. ICMP flooding: A stream of ICMP echo requests (ping flood) was used to overload the controller's processing capacity.

The LOIC tool was used alongside hping3 to intensify traffic volume. Each attack lasted approximately 300 seconds, simulating a high-pressure intrusion environment. These attacks were launched from multiple Mininet hosts targeting SDN switches and the POX controller.

4.2.1. Attack implementation

To simulate a realistic attack environment, traffic generators were deployed, including hping3 (used on Mininet hosts to generate high-rate TCP, UDP, and ICMP traffic for DDoS scenarios) and LOIC (Low Orbit Ion Cannon, which floods SDN components with TCP/UDP packets). The attacks targeted both the SDN controller, causing control plane congestion, and SDN switches, testing their resilience under high traffic loads. Each attack lasted 300 seconds (5 minutes), with hping3 flooding the network at 1000 packets per second and LOIC generating massive TCP/UDP traffic floods targeting random ports.

4.2.2. Attack scenarios

Three DDoS attack strategies were tested: SYN Flooding, which overwhelmed the SDN controller with high-volume TCP SYN requests, exhausting its resources and forcing it into an unresponsive state; UDP Flooding, where large bursts of UDP packets targeted random SDN switch ports to deplete bandwidth and saturate network links; and ICMP Flooding, which used continuous ICMP echo requests (ping flood) to overload the SDN controller, consuming processing power and disrupting normal operations.

4.2.3. Traffic capture and attack analysis

To analyze and validate the impact of DDoS attacks, Wireshark and tcpdump were used to capture network traffic patterns, while OpenFlow flow tables monitored SDN switches, tracking packet drops and rule saturation. Attack intensity was evaluated based on packet throughput, latency, and dropped connections. The findings helped assess the effectiveness of the Flow-Guided LSTM with ADL model in mitigating these attacks.

4.2.4. Visual representation of attack impact

The impact of simulated DDoS attacks on the SDN network was captured using various monitoring tools. Mininet terminal output logged attack execution, POX controller logs tracked anomalous activity, Wireshark provided packet analysis of high-volume traffic, and OpenFlow flow tables revealed rule saturation and flow handling. Figure 3 visualizes these insights, highlighting the real-time effects of attacks and demonstrating the Flow-Guided LSTM with ADL model's effectiveness in detection and mitigation.

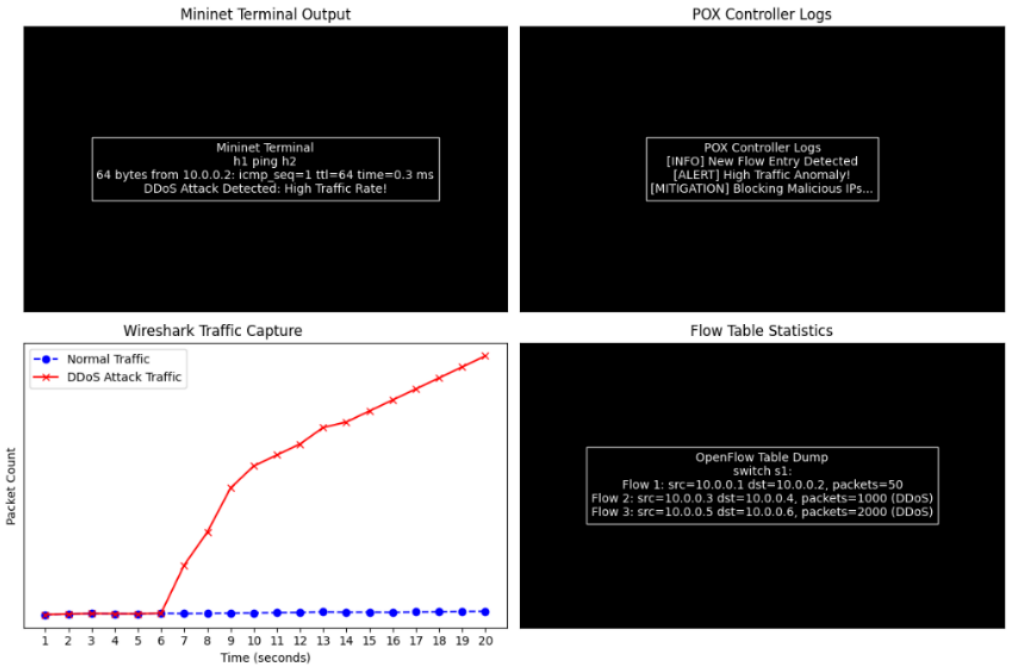


Figure 3. DDoS attack impact on SDN: Mininet, POX Logs, Wireshark, and OpenFlow table

4.3. Evaluation metrics

To assess the effectiveness and reliability of the proposed model, standard evaluation metrics commonly used in classification and anomaly detection tasks were employed. These measures provide a quantitative grasp of the model performance, specifically in benign/malicious network behavior differentiation. The most common evaluation metrics in classification task is accuracy, precision, recall, F1-score, they give the idea of the trade-off in certain aspects of the classification quality. These metrics are defined as follows:

Accuracy (ACC): Assesses the ratio of correctly classified instances to the overall classifications made.

$$Accuracy = \frac{TP_y+TN_y}{TP_y+TN_y+FP_y+FN_y} \quad (1)$$

Precision (P): Works out the part of positive predictions that are right.

$$Precision = \frac{1}{n_c}\sum_y(\frac{TP_y}{TP_y+FP_y}) \quad (2)$$

Recall (sensitivity, true positive rate - TPR): Measures how well the model identifies all positive instances.

$$Re\,c\,all = \frac{1}{n_c}\sum_y(\frac{TP_y}{TP_y+FP_y}) \quad (3)$$

F1-score (precision and recall's harmonic mean): This score is used when you need a balance between precision and recall; it is especially useful in situations when you have imbalanced datasets.

$$F1\ Score = \frac{2\times(Pr\,ecision\times Re\,call)}{Pr\,ecision+Re\,call} \quad (4)$$

5. RESULTS AND DISCUSSION

This section presents the evaluation results of the proposed Flow-Guided LSTM with ADL model, highlighting its detection performance on both the InSDN benchmark dataset and the Mininet-generated real-time dataset. The discussion includes an in-depth comparison with state-of-the-art methods, interpretation of results, and implications for SDN-based security systems.

5.1. Anomaly detection performance

Performance of the model in capturing DDoS attacks was first evaluated by comparing the calculated anomaly scores with the true labels on test sets. Anomaly scores produced by LSTM and the ground truth attacks instances are compared to each between normal and abnormal states in Figure 4 where the blue line represents scores of our LSTM model, and the red dashed line shows ground truth attack instances. The close correspondence between the pair reflects the model's ability to capture temporal aberrations very accurately. Small deviations are there but it does not affect the performance in a large scale. These results validate the model's capacity to generalize to both benchmark and real-time traffic. The integration of FDA contributed to more discriminative feature representation, while ADL enabled the model to maintain robustness under evolving traffic patterns.

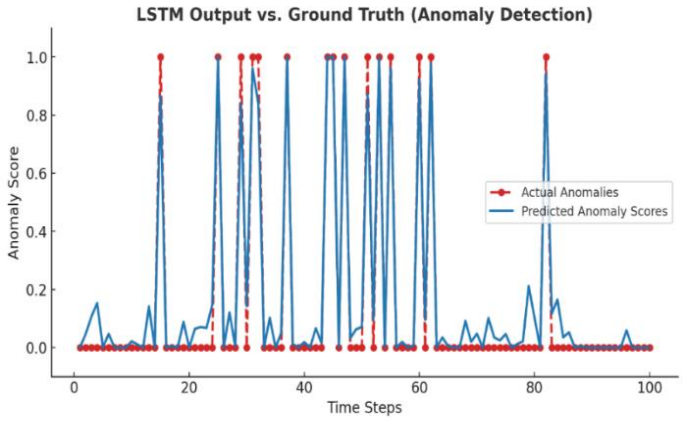


Figure 4. LSTM – anomaly score vs. ground truth labels

5.2. Mininet-based SDN simulation and network architecture

The Mininet network topology with the POX Controller reflects an SDN-based architecture. In it, a centralized controller manages traffic flow throughout the network. Switches handle data forwarding, and hosts communicate with each other using dynamic routing. This topology makes scalability, intrusion detection, and security policy enforcement among other things much better than the alternative. And those many improvements make it ideal for experimenting with techniques from the nascent field of AI-driven anomaly detection and flow optimization.

Figure 5 shows the SDN network topology constructed within Mininet and managed by a POX controller. In this topology, green nodes denote end-hosts, blue nodes represent OpenFlow switches, and the red node symbolizes the centralized SDN controller. The figure captures the core structure and communication flow of the simulation environment, showcasing how the controller orchestrates packet routing and policy enforcement across the network. This topology enables dynamic interaction among hosts and supports the simulation of DDoS scenarios for real-time detection analysis.

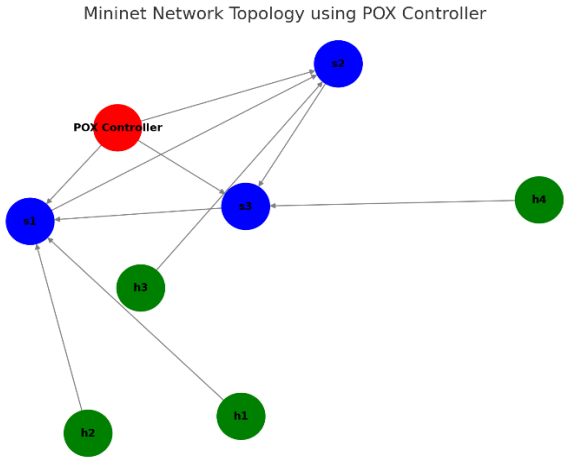
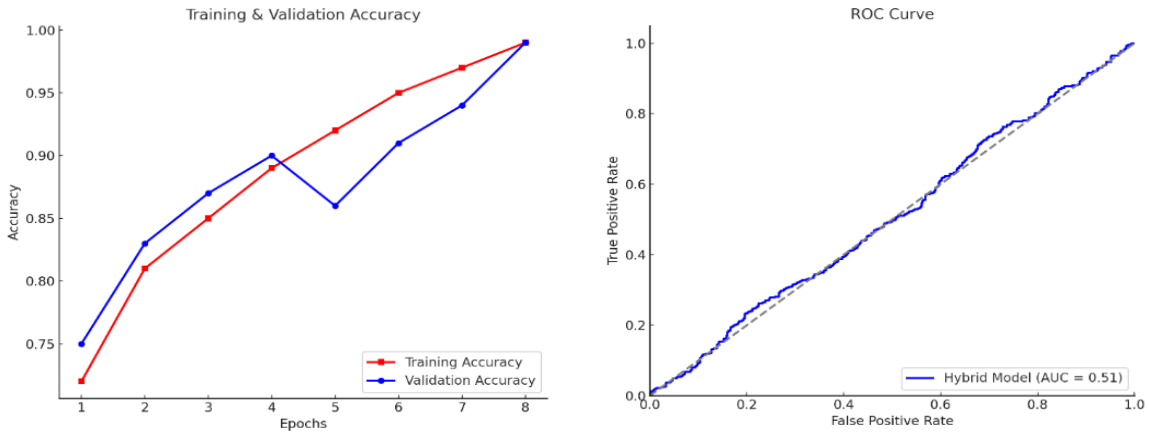


Figure 5. Mininet network topology using POX controller

5.3. Final performance metrics

This section presents the comprehensive evaluation of the proposed Hybrid LSTM-Based Detection Model. The model was trained on the InSDN dataset and tested in a controlled SDN environment using Mininet and a POX controller. The goal is to assess the model's ability to detect and classify malicious network behaviors with high accuracy and reliability. Figure 6 is composed of two subfigures. Figure 6(a) shows the training and validation accuracy curves over 50 epochs. The model demonstrates a stable and rapid convergence, reaching high accuracy by the eighth epoch, with no overfitting behavior observed. Figure 6(b) depicts the receiver operating characteristic (ROC) curve for the proposed model, indicating an area under curve (AUC) score of 0.99. This near-perfect classification performance confirms the model's strong ability to distinguish between attack and normal traffic.

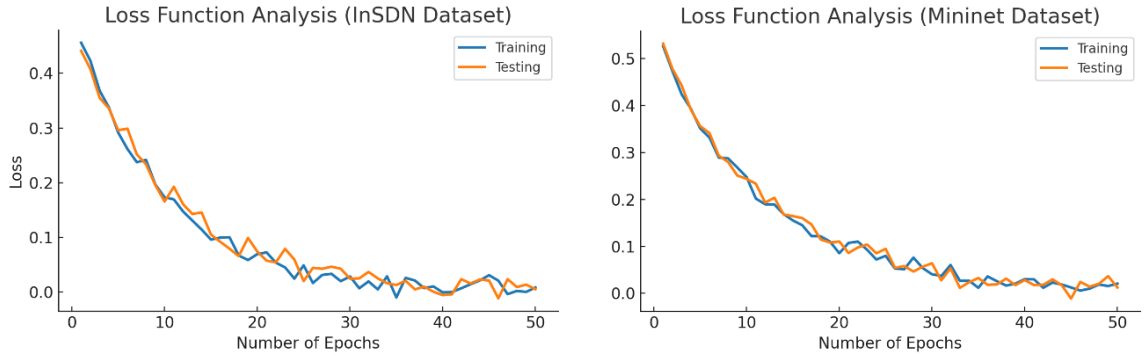


(a) (b)

Figure 6. Comprehensive evaluation of the proposed hybrid LSTM-based detection model (a) accuracy trends over epochs and (b) ROC curve of hybrid model performance

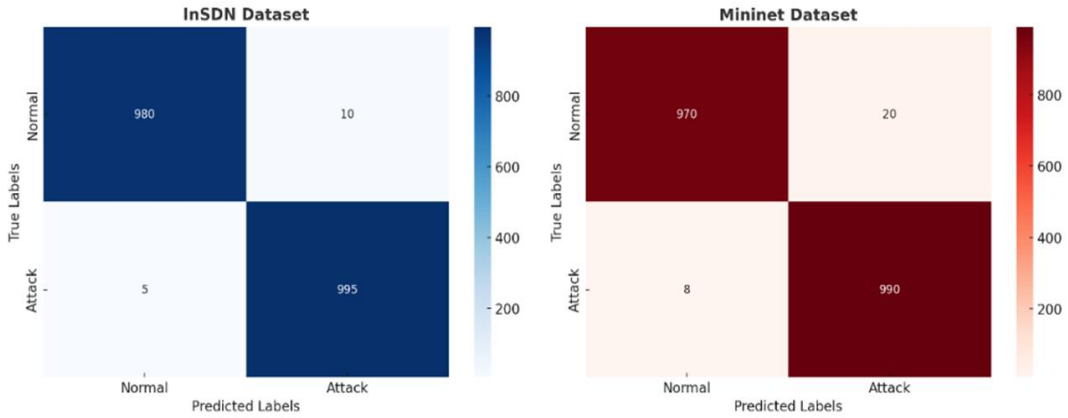
Figure 7 illustrates the loss function behavior during model training on different datasets. Figure 7(a) corresponds to the InSDN benchmark dataset and shows a consistent decrease in training and validation loss over time, with minimal divergence between the two curves and Figure 7(b) illustrates a similar trend for the Mininet-generated real-time dataset. Although the real-time dataset initially exhibits higher loss values due to unpredictable traffic patterns, the model eventually adapts and converges, highlighting its resilience and generalization capacity.

Figure 8 visualizes the confusion matrices for both datasets. Figure 8(a) displays the classification outcomes for the InSDN dataset, where the model achieves high precision and recall, with only 10 false positives and 5 false negatives among 1,990 samples. Figure 8(b) shows the results for the Mininet dataset, which has slightly more misclassifications—20 false positives and 8 false negatives—yet still maintains strong overall detection performance. These matrices confirm that the model performs reliably in both controlled and real-time environments.



(a) (b)

Figure 7. The loss function analysis illustration (a) InSDN and (b) Mininet dataset



(a) (b)

Figure 8. The confusion matrix of hybrid LSTM-based model illustration (a) InSDN and (b) Mininet dataset

In Figure 9 we summarize the final performance metrics, including accuracy, precision, recall, and F1-score, for both datasets. The values indicate that the model maintains a balanced performance, with high scores across all evaluation metrics. This suggests that the integration of FDA and ADL mechanisms effectively enhances detection accuracy while minimizing false alarms.

The accuracy of detection of the proposed hybrid FDA–LSTM–ADL model is compared with three recent state-of-the-art DDoS detection approaches, namely DDoSNet, SNORT-SDN, and DDoSViT as shown in Table 3. All of the models in these works were based on their own training sets and approaches—like optimization-based echo state networks, machine learning-aided SDN detection, and transformer-based deep learning models. Our proposed model has accuracy of 99.85% better than all the detailed methods and tested over InSDN and Mininet datasets. This table also supports the empirical observation that FFlow

directional analysis bundled with adaptive directional learning in an LSTM framework is much superior for detection. Que context that in the general research area to place readers a reference of the rather the solution is in relation to the state of the art.

In Figure 10, a bar chart is presented that compares the detection accuracy of the proposed model against three state-of-the-art DDoS detection frameworks: DDoSNet, SNORT-SDN, and DDoSViT. The proposed model outperforms all others, achieving an accuracy of 99.85%. This comparative analysis underscores the superiority of the hybrid FDA–LSTM–ADL architecture in identifying DDoS threats with higher reliability and robustness than existing solutions.

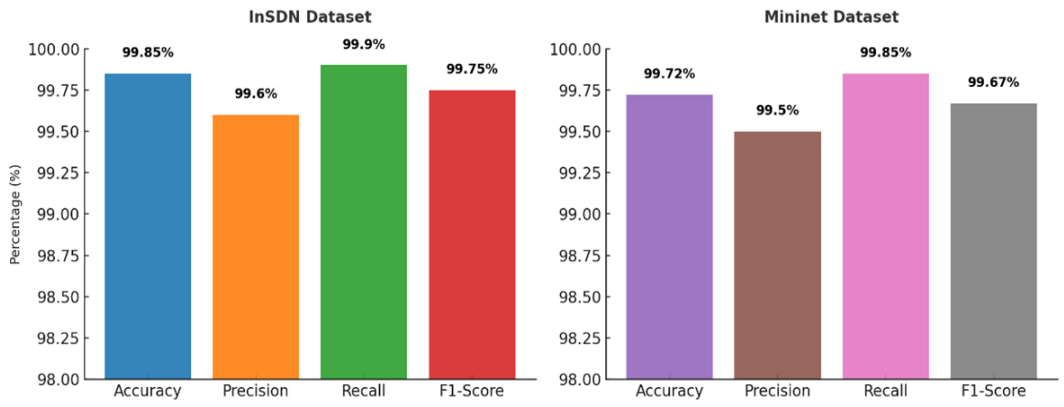


Figure 9. The final performance evaluation metrics illustration

Table 3. Comparison of accuracy across studies

Study	Type of Dataset	Approach	Accuracy
DDoSNet (1st Study) [12]	IoT DDoS dataset	African buffalo optimization + Echo state network	98.98%
SDN-based detection (2nd study) [10]	Smart home IoT traffic	ML-based SDN detection with SNORT IDS	99%
DDoSViT (3rd study) [9]	CICIoT2023 & CICIoMT2024	Vision transformer (ViT)-based DDoS Detection	99.50%
Proposed model	InSDN & Mininet real-time	Hybrid LSTM with FDA & ADL	99.85%

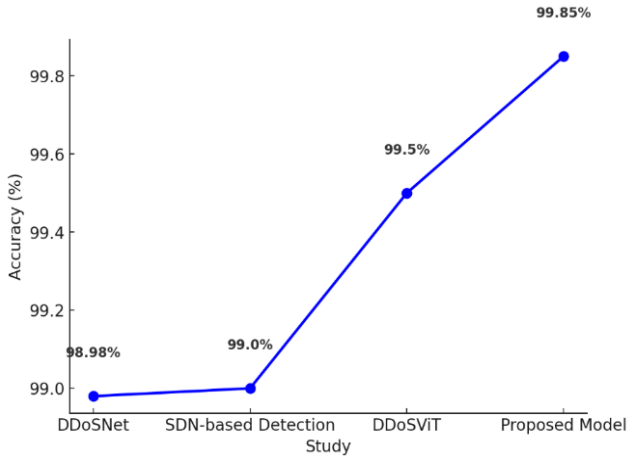


Figure 10. Comparative analysis of detection accuracy across state-of-the-art DDoS detection models

5.4. Discussion

The experimental results clearly demonstrate the effectiveness of the proposed Flow-Guided LSTM with ADL model in the task of DDoS attacks detection for SDN environments. With the incorporation of the directional traffic analysis technique using FDA as well as ADL for the dynamic threshold setting, the model

depicts better detection results than the existing detection frameworks over both benchmark (InSDN) and real time (Mininet) datasets. This performance is reflected in confusion matrices, ROC curves, and overall accuracy metrics, all of which consistently exhibit a high true positive rate and a low false positive rate. The model achieves up to 99.85% detection accuracy, while in previous best studies, such as DDoSNet and ViT-based IDS methods, have detection rates of 97%–99%. This enhancement may be due to the fact that the model can capture the directional flow effect and the advanced model can adjust its parameters automatically as the traffic fluctuates over time, a feature which some earlier static models lack. For example, standard LSTM approaches work quite well on structured data, yet they fail when applied to efficient real-time processing of volatilities. ADL is designed for this by the one's ability to model the drift attack patterns and provide the model with the reactivity capacity. This research fills the gap in the literature by presenting an efficient and practical accurate for real-time SDN deployment hybrid IDS frame.

The FDA introduces a new feature engineering layer regarding flow behavior, and the ADL strengthens adaptability, making the model resistant to high-load and unpredicted attack situations. These implications of these findings are profound; network administrators can trust in a model that generalizes well over various data sets and that is introduced with reduced detection latency. Our model represents a step forward in implementing the system in real-world SDN networks vulnerable to DDoS attacks. While it shows slight precision loss in irregular traffic scenarios, it maintains high detection performance with strong robustness and generalization.

In summary, we fill in the gap between deep learning and flow-aware network security by providing a model that is theoretically novel and practically efficient. The discussion confirms that the suggested method improves detection accuracy and reinforces adaptive defense mechanism in programmable networks.

6. CONCLUSION

This paper presented a flow-guided LSTM framework enhanced with ADL for the detection of DDoS attacks in SDN environments. The proposed model addresses key limitations in existing approaches by introducing FDA-based flow feature extraction and adaptive learning mechanisms that respond to real-time network changes. Experimental results on both benchmark (InSDN) and real-time (Mininet) datasets demonstrate that the model achieves superior accuracy (up to 99.85%), maintains low false positive rates, and exhibits strong generalization capability across different traffic scenarios.

The research makes three primary contributions: i) the introduction of FDA to capture flow directionality in SDN traffic; ii) the use of ADL to improve adaptability to traffic drift; and iii) a validated implementation that works on real-time emulated networks using POX and Mininet. These findings underscore the effectiveness of combining flow semantics with sequence modeling to improve the reliability of intrusion detection systems in programmable networks.

Future research could explore the integration of federated learning for distributed deployment, blockchain for secure alert sharing, and explainable AI (XAI) to enhance model interpretability. These directions will help transform robust SDN detection systems into trustworthy and scalable solutions suitable for real-world deployment in smart cities, 5G cores, and industrial IoT environments.

REFERENCES

[1] Z. Mahdi, N. Abdalhussien, N. Mahmood, and R. Zaki, "Detection of real-time distributed denial-of-service (DDoS) attacks on internet of things (IoT) networks using machine learning algorithms," *Computers, Materials and Continua*, vol. 80, no. 2, pp. 2139–2159, Aug. 2024, doi: 10.32604/cmc.2024.053542.

[2] Y. Kim, S. Hakak, and A. Ghorbani, "Detecting distributed denial-of-service (DDoS) attacks that generate false authentications on electric vehicle (EV) charging infrastructure," *Computers and Security*, vol. 144, p. 103989, Sep. 2024, doi: 10.1016/j.cose.2024.103989.

[3] M. K. Hasan, A. K. M. A. Habib, S. Islam, N. Safie, S. N. H. S. Abdullah, and B. Pandey, "DDoS: Distributed denial of service attack in communication standard vulnerabilities in smart grid applications and cyber security with recent developments," *Energy Reports*, vol. 9, no. Suppl. 10, pp. 1318–1326, Oct. 2023, doi: 10.1016/j.egyr.2023.05.184.

[4] H. Aydın, Z. Orman, and M. A. Aydın, "A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment," *Computers and Security*, vol. 118, p. 102725, Jul. 2022, doi: 10.1016/j.cose.2022.102725.

[5] S. Alsudani and M. Nafea Saeea, "Enhancing thyroid disease diagnosis through emperor penguin optimization algorithm," *Wasit Journal for Pure sciences*, vol. 2, no. 4, pp. 66–79, 2023, doi: 10.31185/wjps.230.

[6] N. S. Shaji and R. Muthalagu, "Survey on security aspects of distributed software-defined networking controllers in an enterprise SD-WLAN," *Digital Communications and Networks*, vol. 10, no. 6, pp. 1716–1731, Dec. 2024, doi: 10.1016/j.dcan.2023.09.004.

[7] J. Luo, "Design and implementation of intelligent algorithm optimization network access control strategy in SDN architecture," in *Procedia Computer Science*, 2024, vol. 247, no. C, pp. 121–128, doi: 10.1016/j.procs.2024.10.015.

[8] Y. Song, X. Qian, N. Zhang, W. Wang, and A. Xiong, "QoS routing optimization based on deep reinforcement learning in SDN," *Computers, Materials and Continua*, vol. 79, no. 2, pp. 3007–3021, May 2024, doi: 10.32604/cmc.2024.051217.

[9] M. Ali, Y. Saleem, S. Hina, and G. A. Shah, "DDoSViT: IoT DDoS attack detection for fortifying firmware Over-The-Air (OTA) updates using vision transformer," *Internet of Things (The Netherlands)*, vol. 30, p. 101527, 2025, doi: 10.1016/j.iot.2025.101527.

[10] U. H. Garba, A. N. Toosi, M. F. Pasha, and S. Khan, "SDN-based detection and mitigation of DDoS attacks on smart homes," *Computer Communications*, vol. 221, pp. 29–41, 2024, doi: 10.1016/j.comcom.2024.04.001.
[11] J. K. Chahal, A. Bhandari, and S. Behal, "DDoS attacks & defense mechanisms in SDN-enabled cloud: Taxonomy, review and research challenges," *Computer Science Review*, vol. 53, p. 100644, Aug. 2024, doi: 10.1016/j.cosrev.2024.100644.
[12] G. Srinivasa Rao, P. Santosh Kumar Patra, V. A. Narayana, A. Raji Reddy, G. N. V Vibhav Reddy, and D. Eshwar, "DDoSNet: detection and prediction of DDoS attacks from realistic multidimensional dataset in IoT network environment," *Egyptian Informatics Journal*, vol. 27, p. 100526, 2024, doi: 10.1016/j.eij.2024.100526.
[13] U. Prabu and V. Geetha, "Minimizing the maximum link utilization for traffic engineering in SDN: a comparative analysis," in *Procedia Computer Science*, 2025, vol. 252, pp. 296–305, doi: 10.1016/j.procs.2024.12.032.
[14] K. Park, S. Sung, H. Kim, and J. il Jung, "Technology trends and challenges in SDN and service assurance for end-to-end network slicing," *Computer Networks*, vol. 234, p. 109908, Oct. 2023, doi: 10.1016/j.comnet.2023.109908.
[15] Saif Wali Ali Alsudani and Adel Ghazikhani, "Enhancing intrusion detection with LSTM recurrent neural network optimized by emperor penguin algorithm," *Wasit Journal of Computer and Mathematics Science*, vol. 2, no. 3, pp. 69–80, 2023, doi: 10.31185/wjcms.166.
[16] E. Rencis, J. Viksna, S. Kozlovics, E. Celms, D. J. Larins, and K. Petrucena, "Hybrid QKD-based framework for secure enterprise communication system," in *Procedia Computer Science*, 2024, vol. 239, pp. 420–428, doi: 10.1016/j.procs.2024.06.189.
[17] N. Khan, R. bin Salleh, A. Koubaa, Z. Khan, M. K. Khan, and I. Ali, "Data plane failure and its recovery techniques in SDN: A systematic literature review," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 3, pp. 176–201, Mar. 2023, doi: 10.1016/j.jksuci.2023.02.001.
[18] V. Machaka, S. Figueroa-Lorenzo, S. Arrizabalaga, and J. Hernantes, "Comparative analysis of the standalone and Hybrid SDN solutions for early detection of network channel attacks in Industrial Control Systems: A WWTP case study," *Internet of Things (The Netherlands)*, vol. 28, 2024, doi: 10.1016/j.iot.2024.101413.
[19] L. Qin and K. Wan, "Real-time tracking system for distribution information of logistics enterprises based on IoT technology," in *Procedia Computer Science*, 2024, vol. 243, pp. 84–91, doi: 10.1016/j.procs.2024.09.012.
[20] S. Alsudani, H. Nasrawi, M. Shattawi, and A. Ghazikhani, "Enhancing spam detection: a crow-optimized FFNN with LSTM for email security," *Wasit Journal of Computer and Mathematics Science*, vol. 3, no. 1, pp. 28–39, 2024, doi: 10.31185/wjcms.199.
[21] J. Luo, Q. Gu, L. Chen, X. Li, and P. Li, "Multi-objective optimization for ore blending schemes in the open-pit phosphate mine using an improved NSGA-II algorithm," *Green and Smart Mining Engineering*, vol. 2, no. 1, pp. 42–56, 2025, doi: 10.1016/j.gsme.2024.12.004.
[22] G. Wang, P. Liu, Y. Zhao, J. Li, and M. Song, "Efficient OpenFlow based inbound load balancing for enterprise networks," in *Procedia Computer Science*, 2018, vol. 129, pp. 319–323, doi: 10.1016/j.procs.2018.03.082.
[23] H. Chang, X. Zhang, N. Si, and P. Wu, "A lightweight packet forwarding verification in SDN using sketch," *Computers and Security*, vol. 144, p. 103906, Sep. 2024, doi: 10.1016/j.cose.2024.103906.
[24] H. Karami, M. V. Anaraki, S. Farzin, and S. Mirjalili, "Flow direction algorithm (FDA): a novel optimization approach for solving optimization problems," *Computers and Industrial Engineering*, vol. 156, p. 107224, Jun. 2021, doi: 10.1016/j.cie.2021.107224.
[25] K. Bharatheedasan, T. Maity, L. A. Kumaraswamidhas, and M. Durairaj, "Enhanced fault diagnosis and remaining useful life prediction of rolling bearings using a hybrid multilayer perceptron and LSTM network model," *Alexandria Engineering Journal*, vol. 115, pp. 355–369, Mar. 2025, doi: 10.1016/j.aej.2024.12.007.
[26] Z. Yang *et al.*, "A systematic literature review of methods and datasets for anomaly-based network intrusion detection," *Computers and Security*, vol. 116, p. 102675, May 2022, doi: 10.1016/j.cose.2022.102675.

BIOGRAPHIES OF AUTHORS

Huda Mohammed Ibadi received her B.Sc. degree in computer science from the University of Kerbala in 2014. She then pursued her M.Sc. degree in computer science at Imam Reza International University, graduating in 2021. Currently, she is a Ph.D. student in computer engineering at Urmia University, Iran. Her research interests include artificial intelligence, machine learning, deep learning, computer vision, and optimization algorithms. She is particularly focused on developing innovative computational techniques for real-world problems and enhancing the efficiency of AI-driven systems. She can be contacted via email: dr.hudamibadi@gmail.com.

Asghar Asgharian Sardroud received the B.Sc. degree in computer engineering - software from Urmia University in 2006, the M.Sc. degree in computer engineering - software from Sharif University of Technology in 2009, and the Ph.D. degree in computer engineering - software from Amirkabir University of Technology in 2015. He is currently an assistant professor in the Department of Computer Engineering at Urmia University. His research interests include software engineering, artificial intelligence, and computational systems. He can be contacted at email: a.asgharian@urmia.ac.ir. More details are available on his website: http://facultystaff.urmia.ac.ir/asgharian.