**Equivalent: Jurnal Ilmiah Sosial Teknik**
**Volume 8, Issue 3, 1346-1361**
**e_ISSN: 2775-0833**
https://jurnalequivalent.id/index.php/jequi/index
DOI: doi.org/10.59261/jequi.v8i3.371

# Flow-Based DDoS Detection and Mitigation in Software-Defined Networking Using a Hybrid CNN-LSTM-Transformer Model

**Tirtadi Muchtar***
Institut Teknologi Bandung, Indonesia

**Kusprasapta Mutijarsa**
Institut Teknologi Bandung, Indonesia

***Corresponding author:**
Tirtadi Muchtar, Institut Teknologi Bandung, Indonesia.
✉tirtadi.muchtar@gmail.com

**Article Info:**

***Article history:***
Received: June 04, 2026
Revised: July 09, 2026
Accepted: July 13, 2026
Available Online: September 03, 2026

***Keywords:***
Software-Defined Networking; DDoS Detection; CNN-LSTM-Transformer; OpenFlow; Ryu Controller.



**Abstract**

**Background:** Software-Defined Networking (SDN) improves network programmability through centralized control, but this architecture also makes the controller vulnerable to Distributed Denial-of-Service (DDoS) attacks that may overload Packet-In processing, exhaust controller resources, and disrupt service availability.
**Objective:** This study aims to develop and evaluate a flow-based DDoS detection and mitigation approach in SDN using a hybrid CNN-LSTM-Transformer model.
**Methods:** The proposed model represents flow records as feature sequences, enabling CNN to capture local feature patterns, LSTM to learn sequential dependencies, and Transformer self-attention to model global inter-feature relationships. Offline evaluation was conducted on CIC-DDoS2019, inSDN, and CIC-IDS2017 using 71 features, stratified 70:20:10 training/validation/testing splits, and three independent seeds. A CNN-LSTM ablation model was used for comparison.
**Results:** The results show the proposed model achieved mean accuracies of 99.87%, 99.98%, and 99.93% on CIC-DDoS2019, inSDN, and CIC-IDS2017, respectively. Compared with CNN-LSTM, the proposed model produced lower mean validation losses and required fewer epochs across all datasets. In real-time validation, UDP, TCP, and ICMP flooding attacks were detected within 0.0693-0.2668s and mitigated using high-priority drop rules, reducing Ryu CPU usage from 52.6%-91.4% during attacks to 0.7%-1.7% after blocking.
**Conclusion:** The findings indicate CNN-LSTM-Transformer model provides an accurate, efficient, and deployment-oriented DDoS defense approach for emulated SDN environments. The proposed model not only achieved high accuracy in offline testing but also demonstrated feasibility in real-time SDN environments with relatively low computational overhead, achieving detection times of 0.0693–0.2668s and reducing Ryu CPU usage from up to 91.4% during attacks to below 1.7% after mitigation.

**To cite this article:** Muchtar, T., & Mutijarsa, K. (2026). Flow-Based DDoS Detection and Mitigation in Software-Defined Networking Using a Hybrid CNN-LSTM-Transformer Model. *Equivalent: Jurnal Ilmiah Sosial Teknik*, *8*(3), 1346-1361. https://doi.org/10.59261/jequi.v8i3.371

## INTRODUCTION

DDoS attacks continue to escalate in both scale and complexity, becoming one of the most critical cybersecurity challenges globally. Recent threat intelligence reports emphasize this growing trend. For instance, NETSCOUT identified nearly 7.9 million DDoS attacks globally in just the first half of 2023, reflecting a massive surge in attack volume (NetScout Systems, 2023). Against this backdrop of escalating global threats, SDN separates the control plane from the data plane, enabling centralized and programmable network management. This architecture allows the controller to determine forwarding policies, while switches forward packets based on

installed flow rules. SDN improves flexibility, traffic engineering, and security monitoring, but the centralized controller also becomes a critical target because disruption at the control plane can affect the entire network operation (Ahuja et al., 2021; Çil & Demirci, 2026). The centralized nature of SDN may also create a single point of failure when the controller is exposed to large-scale attacks (Hssayeni & Mahgoub, 2026; Wang & Li, 2021).

DDoS attacks are serious threats to SDN because they may exhaust controller resources, overload communication channels, and fill switch flow tables with excessive malicious requests. In flooding attacks, unmatched packets trigger Packet-In messages to the controller, forcing it to process many requests within a short time. This condition can increase controller CPU usage, delay legitimate flow installation, and reduce service availability. Therefore, SDN-based DDoS defense requires both accurate detection and fast mitigation at the data plane to prevent attack traffic from continuously burdening the controller (Musa et al., 2024). Similar concerns have also been reported in studies focusing on SDN controller protection, where rapid detection is required to maintain the availability and integrity of network services (Altamemi et al., 2022).

Figure 1 illustrates the three-layer SDN architecture and the DDoS attack surface considered in this study. The figure shows the application plane, control plane, and infrastructure plane, including the SDN controller, OpenFlow switch, legitimate host, attacker, victim, Packet-In flooding path, and drop-rule mitigation path. This representation highlights that DDoS defense in SDN is not only a traffic classification problem but also an operational problem related to controller workload and flow-rule-based mitigation. DDoS flooding experiments in emulated SDN environments have shown that attack detection needs to consider both traffic classification and network performance indicators such as response time, throughput, and controller availability (Benlloch et al., 2026; Sangodoyin et al., 2021).

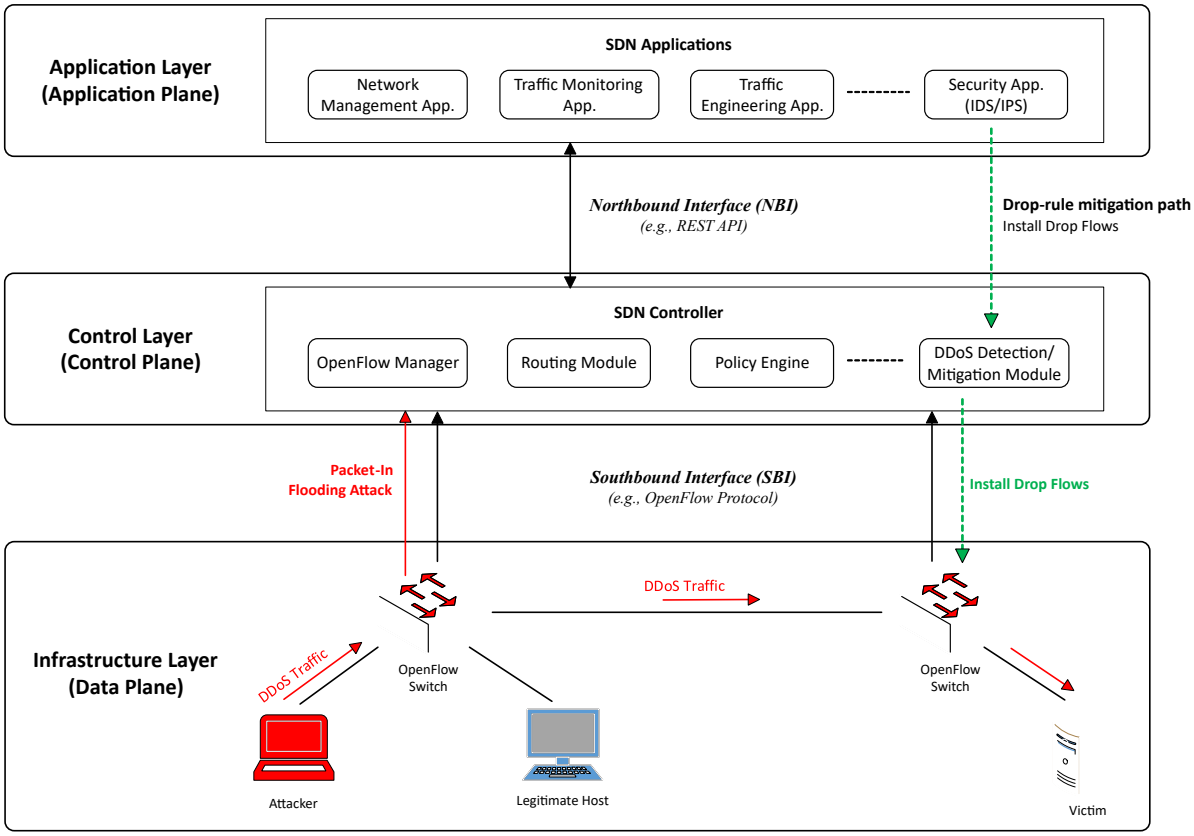


**Figure 1.** Three-layer SDN architecture and DDoS attack surface

Machine learning (ML) and deep learning (DL) have been widely used to detect DDoS attacks in SDN. Traditional ML models such as Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), K-Nearest Neighbors (KNN), Logistic Regression (LR), Naïve Bayes (NB), and XGBoost have shown strong performance using flow-level features. Feature engineering and feature selection are also important because irrelevant or redundant features may reduce

detection performance and increase computational cost (Liu et al., 2023). The importance of qualified features is also emphasized in SDN controller-based DDoS detection, because non-qualified features and unrealistic datasets may increase false positives and reduce detection reliability (Mansoor et al., 2023)

More recent studies have used DL models such as CNN, LSTM, BiLSTM, GRU, and Transformer to learn complex network traffic patterns. CNN can capture local feature patterns, LSTM can learn sequential dependencies, and Transformer self-attention can model broader relationships among features (Wang & Li, 2021; Wang & Wang, 2022). Hybrid CNN-BiLSTM models have also been explored for SDN intrusion detection because they can combine local feature extraction and sequential learning in one architecture (Ben et al., 2023). Recent SDN intrusion detection studies have further compared CNN-LSTM and Transformer-based architectures, showing that both approaches can achieve strong performance when supported by appropriate preprocessing and feature configurations (Ataa et al., 2024). Feature positional encoding further supports Transformer-based attack detection by preserving feature-position information in tabular data (Çavşi Zaim & Yolaçan, 2025).

Although many studies report high accuracy, several research gaps remain. Many ML/DL-based DDoS detection studies still depend on offline datasets, so their results may not fully reflect dynamic SDN traffic behavior in operational environments (Ali et al., 2023). Realistic or real-time validation is still needed to assess whether a detection system can operate effectively in practical SDN scenarios (Musa et al., 2024). Operational metrics such as detection time, response time, controller CPU usage, memory consumption, scalability, and processing overhead are also often underreported, although these metrics are essential for deployment-oriented SDN security systems (Wainwright et al., 2025). Another practical issue is the mismatch between rich offline dataset features and features that can be extracted from OpenFlow statistics in real-time, which may limit controller integration (Muchtar & Mutijarsa, 2025).

Previous studies have proposed several SDN-based DDoS detection and mitigation approaches. Ahuja et al. (2021) developed an automated Mininet-Ryu-based DDoS detection system using an SVC-RF model and achieved 98.8% accuracy (Çil & Demirci, 2026). Al-Dunainawi et al. (2023) combined Mininet, Ryu controller, and an optimized 1D-CNN model for DDoS detection and mitigation in SDN. Wang and Wang (2022) proposed SDN-Defend, an online detection and mitigation using CNN-ELM and IP traceback on inSDN and CIC-IDS2017. Yungaicela-Naula et al. (2021) presented a modular SDN-based architecture using Mininet and ONOS for real-time transport and application layer DDoS detection (Sheikh et al., 2025). Wang et al (2023) proposed CC-Guard as a coordinated SDN controller protection framework that integrates detection, switch migration, anomaly detection, and mitigation. However, a compact framework that combines multi-dataset offline evaluation, ablation analysis, OpenFlow-feasible real-time validation, mitigation, and controller resource measurement is still needed. Several recent DL-based studies have achieved high offline performance using public benchmark datasets. Hnamte (2023) evaluated DCNN and DNN models on inSDN, CICIDS2017, and CIC-DDoS2019 and reported very high detection accuracy. Abdusalomov et al. (2024) evaluated CNN, RNN, LSTM, and GRU models for DDoS detection in SDN using CIC-DDoS2019 and DDoS Attack SDN datasets. Chaymae et al. (2024) proposed a Transformer Encoder and CNN-BiLSTM-based intrusion detection model for SDN using NSL-KDD and CIC-IDS2017. These studies demonstrate the growing relevance of hybrid and attention-based DL architectures, but many of them still place stronger emphasis on classification performance than on controller-level deployment feasibility.

Therefore, this study proposes a flow-based DDoS detection and mitigation approach in SDN using a hybrid CNN-LSTM-Transformer model. The model is designed to learn local, sequential, and global inter-feature relationships from flow records represented as feature sequences. The novelty of this study lies in combining offline multi-dataset evaluation, architectural ablation, and real-time SDN validation within one experimental framework. Unlike previous studies using CNN-LSTM (Ataa et al., 2024; Li et al., 2022), Transformer (Hssayeni & Mahgoub, 2026; Wang & Li, 2021), or CNN-BiLSTM architectures (Ben et al., 2023; Chaymae et al., 2024), this study uniquely integrates all three complementary learning mechanisms within a single architecture, and provides: (1) multi-dataset offline evaluation across three public datasets; (2) architectural ablation to quantify the Transformer contribution; (3) real-time SDN validation

using only 11 OpenFlow-feasible features; (4) controller-based mitigation through high-priority flow rules; and (5) operational resource measurement including CPU usage, detection time, and steady-state recovery time. In addition to classification performance, this study evaluates deployment-oriented aspects through OpenFlow-feasible feature extraction, controller-based mitigation, and operational resource measurements. This approach is expected to provide a more practical perspective for developing accurate and feasible DDoS defense mechanisms in SDN environments.

## METHOD

This study used a quantitative experimental design to develop and evaluate a flow-based DDoS detection and mitigation approach in an SDN environment. The experiment consisted of three stages: (1) offline model evaluation; (2) ablation analysis; and (3) real-time SDN validation. Offline evaluation was conducted using three public datasets, namely CIC-DDoS2019, inSDN, and CIC-IDS2017, while real-time validation was performed in a Mininet-Ryu environment using OpenFlow 1.3. The overall research workflow is illustrated in Figure 2, which describes the process from dataset preparation, preprocessing, model training, performance evaluation, and deployment-oriented validation in the SDN controller.

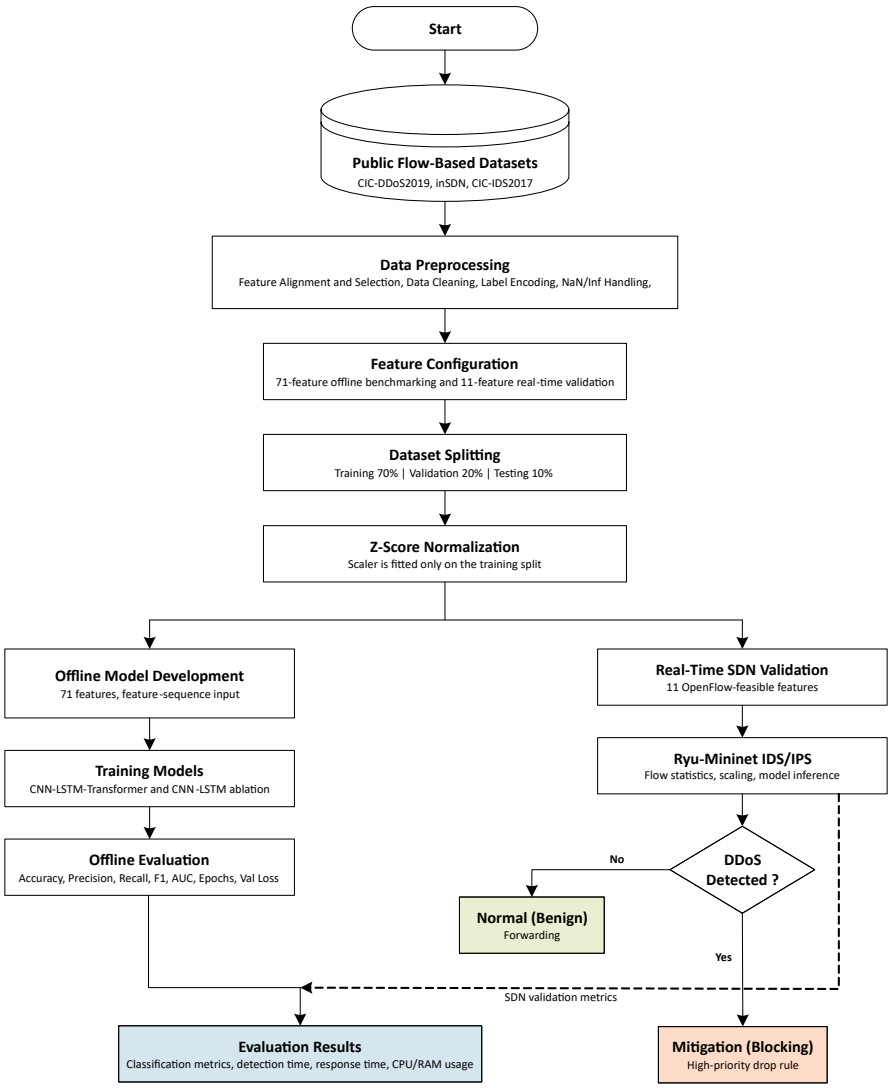


**Figure 2.** Proposed DDoS detection and mitigation workflow

All datasets were formulated as binary classification tasks, where benign traffic was labeled as “0” and DDoS traffic as “1”. Each processed datasets contained 71 input features and one binary label, while the real-time SDN validation used 11 OpenFlow-feasible features extracted from flow statistics in the Ryu controller, these 11 features include Protocol, Flow Duration, Total Forward Packets, Total Length of Forward Packets, Flow Bytes/s, Flow Packets/s, Forward

Packets/s, Packet Length Mean, Average Packet size, Sub-Flow Forward Packets, and Sub-Flow Forward Bytes. The datasets were divided into training, validation, and testing sets using a stratified 70:20:10 ratio to maintain the Benign/DDoS distribution in each subset. Three independent random seeds, namely 42, 43, and 44, were used to evaluate model stability. The number of records, Benign/DDoS ratio, feature configuration, split ratio, and classification task are summarized in Table 1.

**Table 1**
*Dataset overview and experimental configuration*

| Dataset | Records | Benign/ DDoS Ratio | Offline Features | Real-Time Features | Split | Task |
|---|---|---|---|---|---|---|
| CIC-DDoS2019 | 205.565 | 51:49 | 71 | 11 | 70:20:10 | Binary |
| inSDN | 141.945 | 48:52 | 71 | 11 | 70:20:10 | Binary |
| CIC-IDS2017 | 225.709 | 43:57 | 71 | 11 | 70:20:10 | Binary |

Data preprocessing was applied consistently to all datasets. Missing, NaN, and infinite values were replaced with "0" to prevent numerical errors during training. Feature scaling was performed using StandardScaler, where the scaler was fitted only on the training set and then applied to the validation and testing sets to avoid data leakage. The normalization process followed Equation (1), where $x$ is the original feature value, $\mu$ is the mean of the training set, and $\sigma$ is the standard deviation of the training set.

$$z = \frac{x-\mu}{\sigma} \quad (1)$$

After scaling, each flow record was reshaped into a feature-sequence representation with one channel, as expressed in Equation (2). In this representation, $N$ denotes the number of samples and $F$ denotes the number of features. This format allowed the model to process tabular flow features as an ordered feature sequence.

$$X' \in R^{N\times F\times 1} \quad (2)$$

The proposed CNN-LSTM-Transformer model was designed to combine local, sequential, and global feature learning. The input feature sequence was first projected using a dense embedding layer and combined with positional embedding to preserve feature-order information. A Conv1D layer was then used to extract local patterns among neighboring features, followed by an LSTM layer to learn sequential dependencies within the feature sequence. A Transformer encoder was added to capture global inter-feature relationships, and the output was processed through global average pooling and a sigmoid dense layer for binary classification. The processing flow of the proposed model is shown in Figure 3.

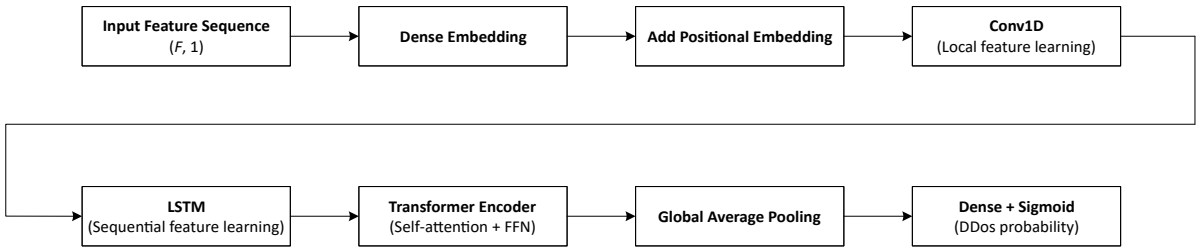


**Figure 3.** Processing flow of the proposed CNN-LSTM-Transformer model

The Transformer encoder used the scaled dot-product attention mechanism to measure the relationship between query, key, and value representations, as expressed in Equation (3).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

To evaluate the contribution of the Transformer component, a CNN-LSTM ablation model was developed using the same input representation, embedding strategy, Conv1D layer, LSTM layer, optimizer, batch size, split configuration, and random seeds but without the Transformer encoder. Both models were trained using the Adam optimizer with a learning rate of $1 \times 10^{-4}$, batch size of 256, maximum epoch of 100, and binary cross-entropy loss. Early stopping was applied by monitoring validation loss with patience of 5 and minimum delta of $1 \times 10^{-4}$. The binary cross-entropy loss is shown in Equation (4).

$$\mathcal{L}_{\mathcal{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (4)$$

Model performance was evaluated using accuracy, precision, recall, F1-score, and AUC. Accuracy measures the proportion of correctly classified samples, precision measures the correctness of positive predictions, recall measures the ability to detect actual DDoS samples, and F1-score balances precision and recall. These metrics are defined in Equations (5) - (8), where $TP$, $TN$, $FP$, and $FN$ denote true positive, true negative, false positive, and false negative, respectively.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (6)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (7)$$

$$\text{F1} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \quad (8)$$

For real-time SDN validation, the proposed model was retrained using 11 features feasible from OpenFlow flow statistics and integrated into a Ryu controller. The controller periodically collected flow statistics, extracted the required features, applied the saved scaler, performed model inference, and classified each flow source as benign or DDoS. When the predicted DDoS probability exceeded 0.5, the controller installed high-priority drop rules on the OpenFlow switch to block traffic from the detected attacker. The validation was conducted using a linear 2-switch and 2-host Mininet topology with Open vSwitch and OpenFlow 1.3. UDP, TCP SYN, and ICMP flooding attacks were generated using hping3, while detection time, response time, controller CPU usage, RAM usage, and steady-state recovery time were recorded. The real-time validation topology is shown in Figure 4.

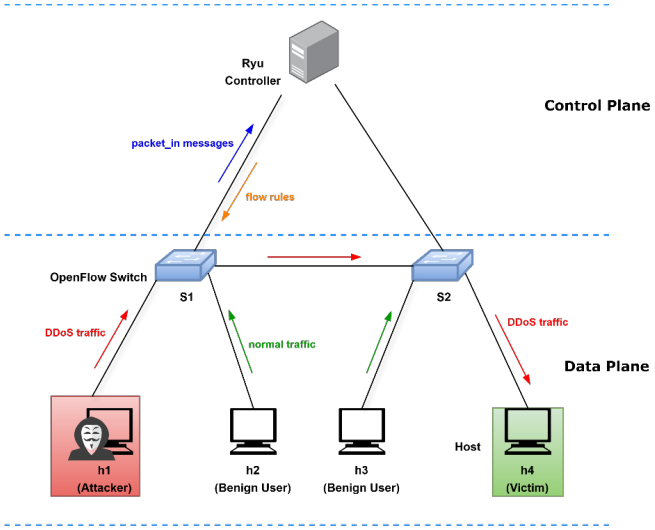


**Figure 4.** Real-time SDN validation topology

Detection time represents the interval between the initial collection of flow statistics and attack identification, whereas response time represents the interval between attack identification and the issuance of the corresponding OpenFlow drop rule by the controller. These metrics are defined in Equations (9) and (10), respectively.

$$T_{detection} = T_{classified} - T_{arrival} \quad (9)$$

$$T_{response} = T_{mitigated} - T_{classified} \quad (10)$$

The simulation was conducted in a controlled virtualized SDN environment to evaluate the feasibility of real-time DDoS detection and mitigation. The main software and hardware resources used in the simulation are summarized in Table 2.

**Table 2**
*Simulation environment specifications*

| Component | Specification |
|---|---|
| Virtual Machine (VM) | Oracle VirtualBox v7.1.4 r165100 |
| Operating System (OS) | Ubuntu LTS 24.04 Desktop 64-bit |
| VM CPU | 2 cores |
| VM RAM | 6 GB |
| Network Emulator | Mininet |
| Controller | Ryu |
| DL Library | TensorFlow |
| Python Version | Python 3.12.3 |
| TensorFlow Version | TensorFlow 2.20.0 |
| scikit-learn Version | scikit-learn 1.8.0 |
| Training Hardware | CPU-only (no GPU; 2-core VM) |

## RESULTS AND DISCUSSION

### Results

#### *Offline Performance Comparison between Proposed and Ablation Models*

The offline evaluation shows that both models achieved high classification performance across CIC-DDoS2019, inSDN, and CIC-IDS2017. However, the CNN-LSTM-Transformer model consistently produced slightly higher mean performance than the CNN-LSTM ablation model in most metrics. The comparison of offline performance is presented in Table 3.

**Table 3**
*Offline performance comparison*

| Dataset | Proposed | | | | | Ablation | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Acc. (%) | Prec. (%) | Recall (%) | F1-score (%) | AUC (%) | Acc. (%) | Prec. (%) | Recall (%) | F1-score (%) | AUC (%) |
| CIC-DDoS2019 | 99.87 | 99.93 | 99.80 | 99.86 | 99.98 | 99.85 | 99.91 | 99.79 | 99.85 | 99.96 |
| inSDN | 99.98 | 99.98 | 99.99 | 99.98 | 99.99 | 99.97 | 99.96 | 99.98 | 99.97 | 99.98 |
| CIC-IDS2017 | 99.93 | 99.95 | 99.91 | 99.94 | 99.99 | 99.87 | 99.88 | 99.89 | 99.89 | 99.98 |

Table 3 shows that both models produced strong offline detection performance. The proposed CNN-LSTM-Transformer model achieved the highest result on inSDN, with an accuracy of 0.9998, precision of 0.9998, recall of 0.9999, F1-score of 0.9998, and AUC of 0.9999. The model also obtained high performance on CIC-IDS2017, with an accuracy of 9993 and F1-score of 0.9994, while CIC-DDoS2019 achieved an accuracy of 0.9987 and F1-score of 0.9986.

Compared with the CNN-LSTM ablation model, the proposed model produced higher accuracy on all datasets. On CIC-DDoS2019, the accuracy increased from 0.9985 to 0.9987. On inSDN, the accuracy increased from 0.9997 to 0.9998. On CIC-IDS2017, the improvement was more visible, with accuracy increasing from 0.9987 to 0.9993. The F1-score and AUC values also increased consistently across the three datasets.

The precision and recall values indicate that both models were able to classify benign and DDoS traffic with high reliability. However, the proposed model showed slightly stronger balance between precision and recall. The AUC values, which ranged from 0.9998 to 0.9999 for the proposed model, also indicate strong discriminative capability in separating benign and DDoS traffic.

***Training Behavior of the Proposed and Ablation Models***

Training behavior was evaluated using accuracy and loss curves for both the proposed CNN-LSTM-Transformer model and the CNN-LSTM ablation model, supported by the observed average number of epochs and best validation loss. The proposed model required fewer epochs to reach optimal performance, with mean values of 16 on CIC-DDoS2019, 22 on inSDN, and 21 on CIC-IDS2017. The corresponding best validation loss values were 0.005775, 0.001685, and 0.003471, respectively. in contrast, the CNN-LSTM ablation model required more epochs, with mean values of 42, 33, and 25 on CIC-DDoS2019, inSDN, and CIC-IDS2017, respectively, and achieved higher validation loss values of 0.006629, 0.003011, and 0.004489. These results indicate that the proposed model converges faster and achieves better generalization performance across all datasets.

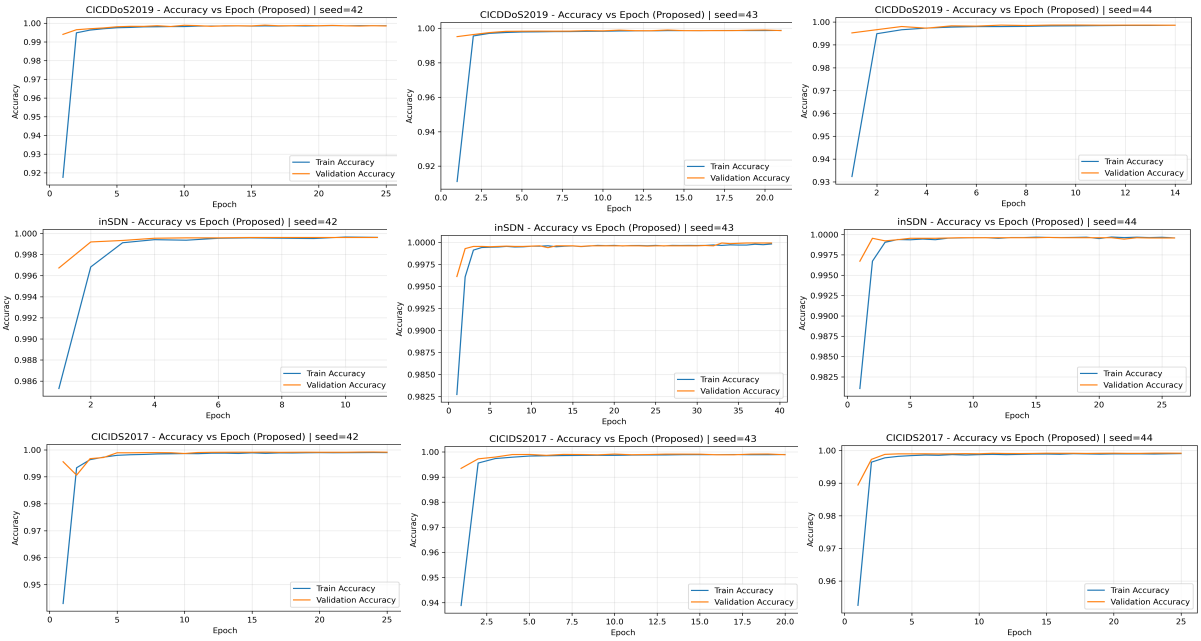


**Figure 5.** Accuracy vs number of Epoch (Proposed)

Figure 5 presents the training and validation accuracy curves of the proposed model on CIC-DDoS2019, inSDN, and CIC-IDS2017. The figure should be arranged into three subplots, with each subplot representing one dataset. Each subplot shows the progression of training accuracy and validation accuracy across epochs.

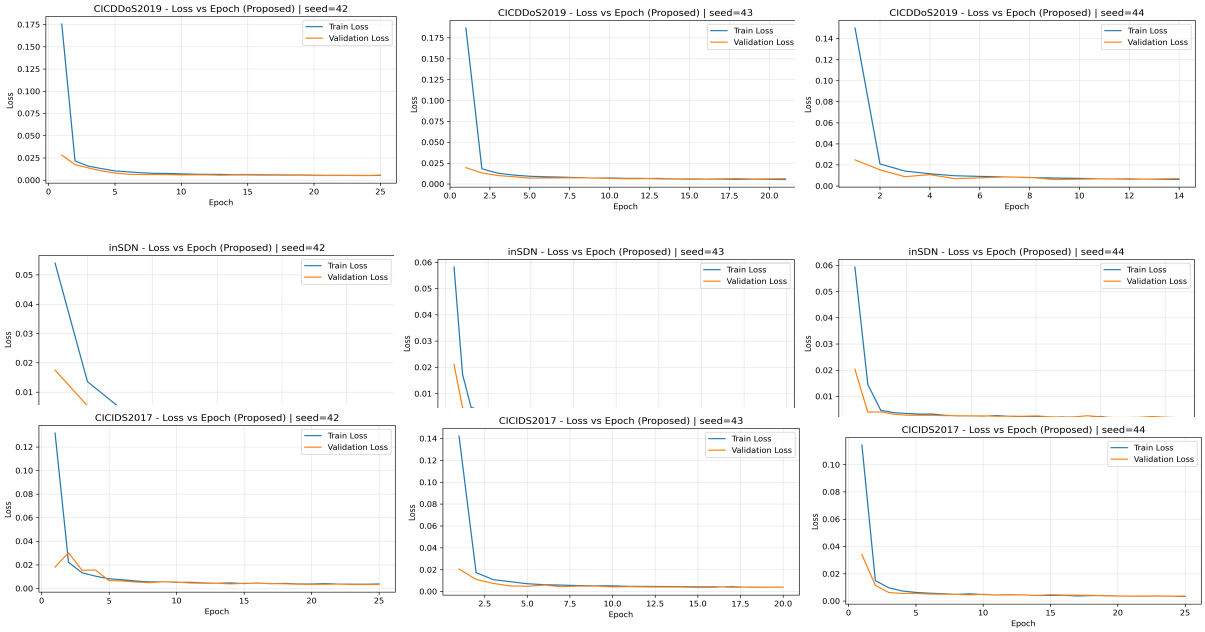


**Figure 6.** Loss vs number of Epoch (Proposed)

Figure 6 presents the training and validation loss curves of the proposed model on CIC-DDoS2019, inSDN, and CIC-IDS2017. The figure should also be arranged into three subplots. These curves illustrate the optimization process and validation behavior of the proposed architecture during training.

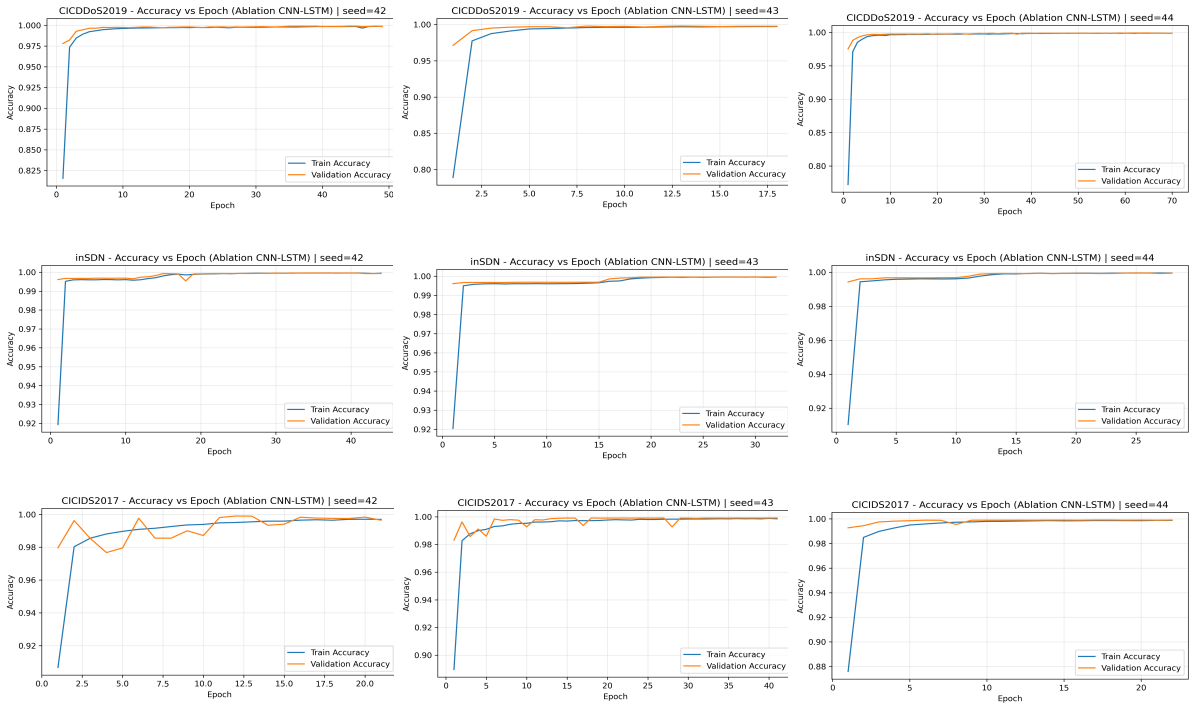


**Figure 7.** Accuracy vs number of Epoch (Ablation)

Figure 7 presents the training and validation accuracy curves of the CNN-LSTM ablation model on CIC-DDoS2019, inSDN, and CIC-IDS2017. Similar to Figure 5, the figure should be arranged into three subplots. This figure is used to show how the ablation model reached its accuracy performance across epochs without the Transformer encoder.

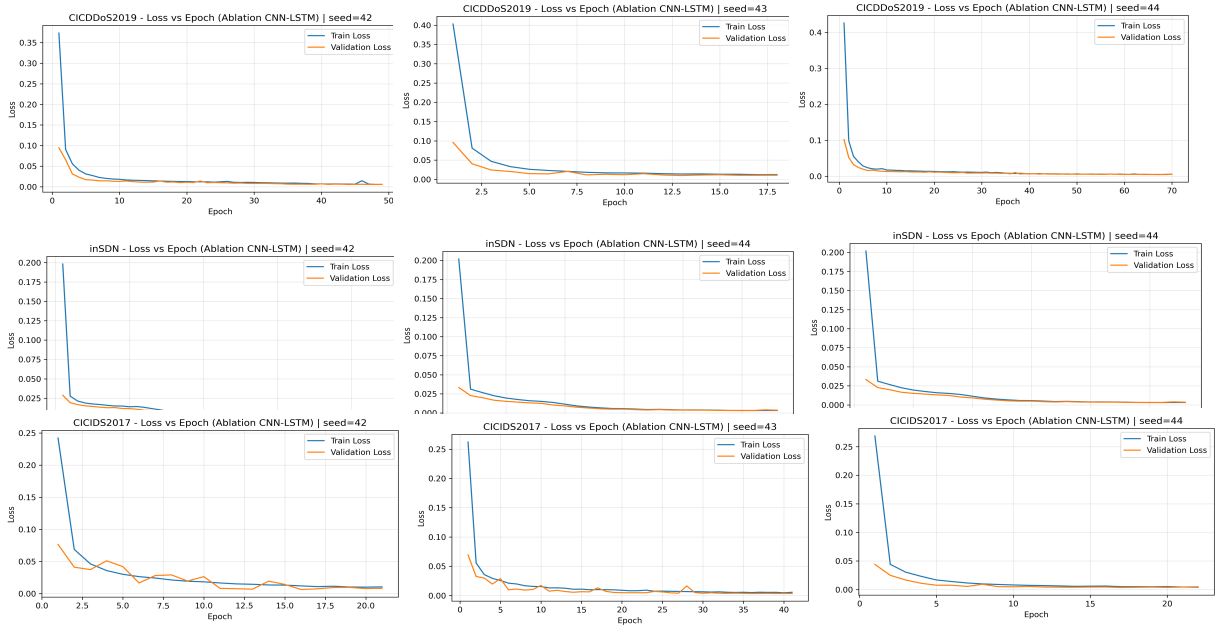


**Figure 8.** Loss vs number of Epoch (Ablation)

Figure 8 presents the training and validation loss curves of the ablation model on CIC-DDoS2019, inSDN, and CIC-IDS2017. This figure supports the convergence comparison by showing the loss behavior of the ablation model across epochs and enabling visual comparison with the proposed model in Figure 6.

The training curves are expected to show that both models converged successfully, but the proposed CNN-LSTM-Transformer model reached lower validation loss with fewer epochs. These values show that the proposed model reached its best validation condition with fewer epochs and lower validation loss than the CNN-LSTM ablation model across the three datasets.

**Real-Time SDN Detection and Mitigation Results**

The real-time SDN validation was conducted to evaluate whether the 11-feature models could detect and mitigate flooding attacks in the Mininet-Ryu environment. The validation included UDP, TCP SYN, and ICMP flooding scenarios. The detection and response results are summarized in Table 4.

**Table 4**

*Real-time SDN validation results*

| Model | Protocol | Detection Time (s) | Response Time (s) | Mitigation Status |
|---|---|---|---|---|
| CIC-DDoS2019 | UDP | 0.1784 | 0.0022 | Blocked |
| CIC-DDoS2019 | TCP | 0.2668 | 0.0034 | Blocked |
| CIC-DDoS2019 | ICMP | 0.1617 | 0.0029 | Blocked |
| inSDN | UDP | 0.0693 | 0.0040 | Blocked |
| inSDN | TCP | 0.0718 | 0.0037 | Blocked |
| inSDN | ICMP | 0.0785 | 0.0016 | Blocked |
| CIC-IDS2017 | UDP | 0.0822 | 0.0007 | Blocked |
| CIC-IDS2017 | TCP | 0.0736 | 0.0006 | Blocked |
| CIC-IDS2017 | ICMP | 0.0772 | 0.0008 | Blocked |

Table 4 shows that all tested flooding attacks were detected and blocked successfully. The detection time ranged from 0.0693s to 0.2668s. The fastest detection occurred in the inSDN-based model during UDP flooding, while the longest detection occurred in the CIC-DDoS2019-based model during TCP SYN flooding. All detection times remained below 0.3s.

The response time ranged from 0.0006s to 0.0040s. The lowest response time was produced by the CIC-IDS2017-based model during TCP SYN flooding, while the highest response

time was produced by the inSDN-based model during UDP flooding. These values indicate that after an attack source was detected, mitigation rules could be installed rapidly.

The mitigation column shows that all detected attacks were blocked. The controller applied high-priority drop rules to stop traffic from the detected attacker source. This confirms that the validation pipeline performed both detection and mitigation rather than only offline classification.

**Controller Resource Utilization during Real-Time Validation**

Controller resource utilization was observed to measure the operational effect of DDoS attacks and mitigation. The summarized CPU, RAM, and steady-state results are shown in Table 5.

**Table 5**

*Controller resource utilization during real-time validation*

| Model | Avg. CPU During Attack (%) | Avg. CPU After Blocking (%) | Avg. Steady-State Time (s) | Avg. RAM (%) |
|---|---|---|---|---|
| CIC-DDoS2019 | 88.4 | 0.9 | 20.0 | 11.2 |
| inSDN | 74.8 | 1.1 | 18.7 | 11.2 |
| CIC-IDS2017 | 88.2 | 1.0 | 12.0 | 11.2 |

Table 5 shows that the Ryu controller experienced high CPU usage during flooding attacks. The average CPU usage during attack reached 88.4% for the CIC-DDoS2019-based model, 74.8% for the inSDN-based model, and 88.2% for the CIC-IDS2017-based model. After mitigation, the average CPU usage decreased to approximately 1% for all models.

The steady-state recovery time varied across models. The CIC-DDoS2019-based model required an average of 20.0 s to reach stable CPU usage after attack initiation. The inSDN-based model required 18.7s, while the CIC-IDS2017-based model required 12.0s. RAM usage remained stable at 11.2% in all validation scenarios.

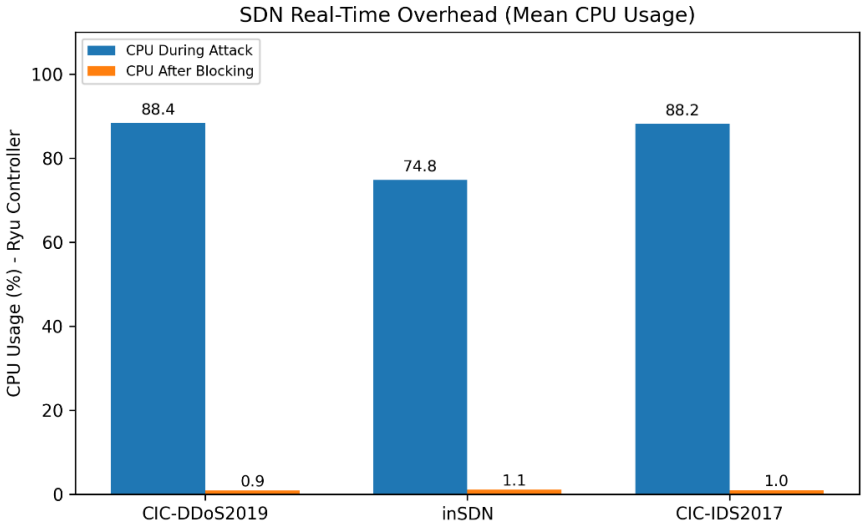


**Figure 9.** Ryu controller CPU usage before and after mitigation

Figure 9 presents a grouped bar chart comparing the average Ryu CPU usage during attacks and after blocking. The figure highlights the reduction in controller workload after high-priority drop rules were applied.

**Comparison with Previous Studies**

To provide a broader perspective on the effectiveness of the proposed model, a comparison with previous studies on SDN-based DDoS detection is presented in Table 6. The comparison includes various deep learning and hybrid models evaluated on commonly used benchmark datasets such as CIC-DDoS2019, inSDN, and CIC-IDS2017. These prior works employ

different architectural approaches, including CNN-based, recurrent, and hybrid models, and report high classification performance under offline evaluation settings. By positioning the proposed model alongside these studies, this comparison aims to highlight its relative performance and demonstrate its competitiveness within the current state of the art.

**Table 6**
*Comparison of DDoS detection performance in SDN with previous studies*

| No | Authors | Model | Dataset | Best Accuracy |
|---|---|---|---|---|
| 1. | (Alghazzawi et al., 2021; Bilal et al., 2026) | CNN-BiLSTM | CIC-DDoS2019 | 94.52% |
| 2. | (Wang & Wang, 2022) | CNN-ELM | inSDN<br>CIC-IDS2017 | 99.91%<br>98.92% |
| 3. | (Mbasuva & Zodi, 2022) | CNN-DNN-RNN | CIC-IDS2017 | 99.05% |
| 4. | (Li et al., 2022) | CNN-LSTM | CIC-IDS2017 | 96.13% |
| 5. | (Chen et al., 2023) | DBN-LSTM | CIC-DDoS2019 | 96.55% |
| 6. | (Said & Askerzade, 2023) | CNN-BiLSTM | inSDN | 98.03% |
| 7. | (Ataa et al., 2024) | CNN-LSTM<br>Transformer | inSDN | 99.08%<br>99.16% |
| 8. | (Abid et al., 2024) | CNN-LSTM | CIC-DDoS2019 | 99.4% |
| 9. | This study | Proposed model | CIC-DDoS2019<br>inSDN<br>CIC-IDS2017 | 99.87%<br>99.98%<br>99.93% |

As shown in Table 6, the proposed model achieves superior accuracy across all evaluated datasets, reaching 99.87% on CIC-DDoS2019, 99.98% on inSDN, and 99.93% on CIC-IDS2017, outperforming most existing approaches. While several previous studies have reported strong results, such as CNN-ELM (Wang & Wang, 2022) and Transformer-based models (Ataa et al., 2024), the proposed CNN-LSTM-Transformer model consistently provides higher or more stable performance across multiple datasets. This improvement can be attributed to the integration of complementary feature-learning mechanisms that capture local, sequential, and global feature relationships. In addition, unlike many prior works that focus primarily on offline accuracy, this study also incorporates real-time SDN validation and operational analysis, providing a more comprehensive evaluation of model effectiveness and deployment feasibility.

**Discussion**

The results demonstrate that the proposed CNN-LSTM-Transformer model effectively combines complementary feature-learning mechanisms within a flow-based representation. The consistently high performance across CIC-DDoS2019, inSDN, CIC-IDS2017 indicates that DDoS traffic patterns can be learned not only from individual features but also from inter-feature relationships when flows are represented as sequences. In this architecture, CNN extracts local feature patterns, LSTM captures sequential dependencies, and the Transformer encoder enhances representation through self-attention. This observation is consistent with prior studies emphasizing the importance of feature quality and preprocessing in SDN-based DDoS detection (Liu et al., 2023), as well as the role of positional encoding in improving Transformer-based learning on structured features (Çavşi Zaim & Yolaçan, 2025). The ablation results further show that the Transformer component improves convergence efficiency, as reflected in fewer training epochs and lower validation loss, even though metric improvements over the CNN-LSTM baseline are relatively small. This supports the view that self-attention can capture complex feature relationships beyond convolutional and recurrent layers (Hssayeni & Mahgoub, 2026; Wang & Li, 2021), and that hybrid architectures are most effective when each component contributes distinct representational capabilities (Ataa et al., 2024).

The real-time SDN validation highlights the gap between offline classification performance and controller-level feasibility. While many studies report high accuracy using benchmark datasets, such results do not guarantee practical deployment within an SDN controller. This study addresses this gap by separating offline evaluation using 71 features from

real-time validation using 11 OpenFlow-feasible features, which are directly obtainable during controller operation. This distinction is important because not all dataset features are available at runtime, and practical deployment depends on feature accessibility. Prior work has identified challenges in SDN-based DDoS detection, including overfitting, limited real-time validation, and integration constraints (Muchtar & Mutijarsa, 2025). The mitigation results further show that DDoS defense in SDN should be treated as an operational process rather than a standalone classification task. The integration of detection with mitigation, through high-priority drop rules for UDP, TCP SYN, and ICMP flooding, effectively reduced controller load, as indicated by decreased CPU usage. This finding aligns with the need for integrated detection-mitigation mechanisms and the evaluation of operational metrics such as response time and resource utilization (Wainwright et al., 2025).

Compared with previous studies, the contribution of this work lies in combining model performance, feature feasibility, and controller-level validation. Prior research has explored SDN-based DDoS detection using ML/DL models and testbed environments, but this study extends them by integrating multi-dataset evaluation, ablation analysis, real-time validation with OpenFlow-feasible features, and controller resource measurement. Despite these contributions, several limitations should be acknowledged. The evaluation was conducted on a simplified Mininet topology and a single-controller Ryu environment, which may not fully represent large-scale deployments (Sheikh et al., 2025; Yungaicela-Naula et al., 2021). Additionally, the attack scenarios were limited to flooding-based attacks and tested in an emulated environment. Future work should therefore explore more complex topologies, diverse attack types, multi-controller architectures, and semi-production testbeds to better assess scalability and robustness.

## CONCLUSION

This study presents an SDN-based DDoS defense framework that integrates detection and mitigation using a CNN-LSTM-Transformer model. The findings demonstrate that combining local, sequential, and global feature learning improves flow-based attack detection while maintaining feasibility for controller-level implementation. The study also emphasizes that performance evaluation should extend beyond accuracy to include convergence behavior, feature feasibility, and operational impact, such as mitigation effectiveness and controller resource usage. Quantitatively, the proposed model achieved mean test accuracies of 99.87%, 99.98%, and 99.93% on CIC-DDoS2019, inSDN, and CIC-IDS2017, respectively, and consistently outperformed the CNN-LSTM ablation model in both accuracy and convergence efficiency (lower validation loss; fewer training epochs). In real-time SDN validation, all flooding attack types (UDP, TCP SYN, and ICMP) were successfully detected within 0.0693–0.2668 seconds and blocked using high-priority drop rules, reducing Ryu controller CPU usage from a peak of 52.6%–91.4% during attacks to 0.7%–1.7% after mitigation.

By integrating hybrid deep learning, ablation analysis, real-time SDN validation, and resource measurement, this work bridges the gap between offline benchmark performance and practical deployment. The theoretical contribution of this study lies in demonstrating that feature-sequence representation with complementary CNN-LSTM-Transformer learning mechanisms is both effective and architecturally generalizable across different SDN benchmark datasets. The practical contribution is the demonstration that this approach can be operationally integrated into a Ryu controller using only OpenFlow-obtainable features, with measurable mitigation effectiveness in an emulated environment. Importantly, a key limitation of this study is that real-time validation was conducted exclusively in an emulated single-controller Mininet SDN topology with specific flooding attack scenarios, which constrains the generalizability of these findings to production-scale or heterogeneous deployments. Future research should therefore focus on: (1) multi-controller and distributed SDN architectures; (2) topology heterogeneity and diverse attack types including application-layer and slow-rate DDoS; (3) adversarial robustness testing against evasion techniques; (4) latency overhead analysis at larger traffic scales; and (5) lightweight model optimization for resource-constrained controller environments.

**ACKNOWLEDGEMENT**

The authors would like to express their sincere gratitude to the Ministry of Communication and Digital Affairs (Komdigi) of the Republic of Indonesia for supporting this study through a scholarship program. The authors also acknowledge the academic and institutional support provided by the School of Electrical Engineering and Informatics, Institut Teknologi Bandung, which contributed to the completion of this research.

**AUTHOR CONTRIBUTION STATEMENT**

Tirtadi Muchtar contributed to the conceptualization of the study, formulation of the research objectives, methodological design, dataset preprocessing, model development, experimental implementation, data analysis, visualization, and preparation of the original manuscript. He also conducted the offline evaluation, ablation analysis, and real-time SDN validation, including the interpretation and presentation of the research findings.

Kusprasapta Mutijarsa contributed to research supervision, methodological validation, critical review, and manuscript refinement. He provided academic guidance on the research design, interpretation of results, and positioning of the study within SDN-based DDoS detection and mitigation research, and substantially revised the manuscript to enhance its scientific rigor, coherence, and scholarly contribution.

**REFERENCES**

Abdusalomov, A., Meliboev, A., Xamdamov, U., Melih Gul, O., Yazici, A., Elghirani, A., Nasimov, R., Cho, Y. I., & Gul, O. M. (2024). *Enhanced Detection of DDoS Attacks in SDN Using Deep Learning Algorithms*. 1–10.

Abid, Y. A., Wu, J., Xu, G., Fu, S., & Waqas, M. (2024). Multilevel Deep Neural Network Approach for Enhanced Distributed Denial-of-Service Attack Detection and Classification in Software-Defined Internet of Things Networks. *IEEE Internet of Things Journal*, *11*(14), 24715–24725. https://doi.org/10.1109/JIOT.2024.3376578

Ahuja, N., Singal, G., Mukhopadhyay, D., & Kumar, N. (2021). Automated DDoS attack detection in software defined networking. *Journal of Network and Computer Applications*, *187*, 103108. https://doi.org/10.1016/j.jnca.2021.103108

Al-Dunainawi, Y., Al-Kaseem, B. R., & Al-Raweshidy, H. S. (2023). Optimized Artificial Intelligence Model for DDoS Detection in SDN Environment. *IEEE Access*, *11*, 106733–106748. https://doi.org/10.1109/ACCESS.2023.3319214

Alghazzawi, D., Bamasaq, O., Ullah, H., & Asghar, M. Z. (2021). Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection. *Applied Sciences (Switzerland)*, *11*(24). https://doi.org/10.3390/app112411634

Ali, T. E., Chong, Y. W., & Manickam, S. (2023). Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review. *Applied Sciences (Switzerland)*, *13*(5). https://doi.org/10.3390/app13053183

Altamemi, A. J., Abdulhassan, A., & Obeis, N. T. (2022). DDoS attack detection in software defined networking controller using machine learning techniques. *Bulletin of Electrical Engineering and Informatics*, *11*(5), 2836–2844. https://doi.org/10.11591/eei.v11i5.4155

Ataa, M. S., Sanad, E. E., & El-khoribi, R. A. (2024). Intrusion detection in software defined network using deep learning approaches. *Scientific Reports*, *14*(1), 1–15. https://doi.org/10.1038/s41598-024-79001-1

Ben, S. R., Sabir, Z., & Askerzade, I. (2023). CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software-Defined Networking With Hybrid Feature Selection. *IEEE Access*, *11*(December), 138732–138747. https://doi.org/10.1109/ACCESS.2023.3340142

Benlloch, P., Romero, O., Leon, A., & Lloret, J. (2026). MLDAS: Machine Learning Dynamic Algorithm Selection for Software-Defined Networking Security. *Concurrency and Computation: Practice and Experience*, *38*(5), e70637.

Bilal, A., Sharif, K., Zhu, L., Li, F., Xu, C., & Karim, M. M. (2026). Evaluation to integration: Hybrid feature selection framework with ensemble machine learning for intrusion detection. *IEEE Transactions on Dependable and Secure Computing*.

Çavşi Zaim, H., & Yolaçan, E. N. (2025). FPE–Transformer: A Feature Positional Encoding-Based Transformer Model for Attack Detection. *Applied Sciences (Switzerland)*, *15*(3). https://doi.org/10.3390/app15031252

Chaymae, E. Y., Abdellatif, K., Khalid, C., & Jalel, B. O. (2024). Intrusion Detection System using Transformer Encoder and CNN-BiLSTM in Software-Defined Networks. *Proceedings - IEEE Global Communications Conference, GLOBECOM*, *Ml*, 2214–2219. https://doi.org/10.1109/GLOBECOM52923.2024.10901492

Chen, L., Wang, Z., Huo, R., & Huang, T. (2023). An Adversarial DBN-LSTM Method for Detecting and Defending against DDoS Attacks in SDN Environments. *Algorithms*, *16*(4). https://doi.org/10.3390/a16040197

Çil, A., & Demirci, M. (2026). SARE: Secure, Automated, and Regulatory-Compliant Platform for Improving Forensic Readiness in Software-Defined Networks. *International Journal of Communication Systems*, *39*(1), e70299.

Hnamte, V., & Hussain, J. (2023). An efficient DDoS attack detection mechanism in SDN environment. *International Journal of Information Technology (Singapore)*, *15*(5), 2623–2636. https://doi.org/10.1007/s41870-023-01332-5

Hssayeni, M. D., & Mahgoub, I. (2026). A Transformer-Based Intrusion Detection System for Zero-Day Attack Detection in IoT Networks. *Future Internet*, *18*(6), 282.

Li, M., Zhang, B., Wang, G., Zhuge, B., Jiang, X., & Dong, L. (2022). A DDoS attack detection method based on deep learning two-level model CNN-LSTM in SDN network. *Proceedings - 2022 International Conference on Cloud Computing, Big Data Applications and Software Engineering, CBASE 2022*, 282–287. https://doi.org/10.1109/CBASE57816.2022.00062

Liu, Z., Wang, Y., Feng, F., Liu, Y., Li, Z., & Shan, Y. (2023). A DDoS Detection Method Based on Feature Engineering and Machine Learning in Software-Defined Networks. *Sensors*, *23*(13). https://doi.org/10.3390/s23136176

Mansoor, A., Anbar, M., Bahashwan, A. A., Alabsi, B. A., & Rihan, S. D. A. (2023). Deep Learning-Based Approach for Detecting DDoS Attack on Software-Defined Networking Controller. *Systems*, *11*(6), 1–21. https://doi.org/10.3390/systems11060296

Mbasuva, U., & Zodi, G. A. L. (2022). Designing Ensemble Deep Learning Intrusion Detection System for DDoS attacks in Software Defined Networks. *Proceedings of the 2022 16th International Conference on Ubiquitous Information Management and Communication, IMCOM 2022*. https://doi.org/10.1109/IMCOM53663.2022.9721785

Muchtar, T., & Mutijarsa, K. (2025). Machine Learning for DDoS Attack Detection in SDN: A Systematic Literature Review. In *2025 IEEE 2nd International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs)* (pp. 224–229). IEEE. https://doi.org/10.1109/ICoCICs68032.2025.11384079

Musa**,** N. S., Mirza, N. M., Rafique, S. H., Abdallah, A. M., & Murugan, T. (2024). Machine Learning and Deep Learning Techniques for Distributed Denial of Service Anomaly Detection in Software Defined Networks - Current Research Solutions. *IEEE Access*, *12*(January), 17982–18011. https://doi.org/10.1109/ACCESS.2024.3360868

NetScout Systems, I. (2023). NETSCOUT Identified Nearly 7.9 Million DDoS Attacks in 1H2023 According to Its Latest DDoS Threat Intelligence Report.

Said, R. Ben, & Askerzade, I. (2023). Attention-Based CNN-BiLSTM Deep Learning Approach for Network Intrusion Detection System in Software Defined Networks. *2023 5th International Conference on Problems of Cybernetics and Informatics, PCI 2023*, 1–5. https://doi.org/10.1109/PCI60110.2023.10325985

Sangodoyin, A. O., Akinsolu, M. O., Pillai, P., & Grout, V. (2021). Detection and Classification of DDoS Flooding Attacks on Software-Defined Networks: A Case Study for the Application of Machine Learning. *IEEE Access*, *9*, 122495–122508. https://doi.org/10.1109/ACCESS.2021.3109490

Sheikh, M. N. A., Raza, M. S., Hwang, I.-S., Hossain, M. A., Ullah, I., Hasan, T., & Ab-Rahman, M. S. (2025). SDN-Enabled IoT based transport layer DDoS attacks detection using RNNs. *Computers, Materials, & Continua*, *85*(2), 4043.

Wainwright, R., Bagheri, M., Salama, A., & Saatchi, R. (2025). Software-Defined Networking Security Detection Strategies and Their Limitations with a Focus on Distributed Denial-of-

Service for Small to Medium-Sized Enterprises. *Applied Sciences (Switzerland)*, *15*(23), 1–19. https://doi.org/10.3390/app152312389

Wang, H., & Li, W. (2021). DDosTC: A transformer-based network attack detection hybrid mechanism in SDN. *Sensors*, *21*(15). https://doi.org/10.3390/s21155047

Wang, J., & Wang, L. (2022). SDN-Defend: A Lightweight Online Attack Detection and Mitigation System for DDoS Attacks in SDN. *Sensors*, *22*(21). https://doi.org/10.3390/s22218287

Wang, J., Wang, L., & Wang, R. (2023). A Method of DDoS Attack Detection and Mitigation for the Comprehensive Coordinated Protection of SDN Controllers. *Entropy*, *25*(8). https://doi.org/10.3390/e25081210

Yungaicela-Naula, N. M., Vargas-Rosales, C., & Perez-Diaz, J. A. (2021). SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning. *IEEE Access*, *9*, 108495–108512. https://doi.org/10.1109/ACCESS.2021.3101650