



Human-in-the-Loop 1D-CNN DDoS Detection Integrated with MikroTik Firewall Mitigation

Aldila Putri Linanzha, Rifki Indra Perwira, Ahmad Dzakiyyul Fuad,
Oliver Samuel Simanjuntak, Simon Pulung Nugroho

Department of Informatics, Faculty of Industrial Engineering, UPN "Veteran" Yogyakarta, Indonesia

Received : August 28, 2026 | Revised : September 28, 2026 | Accepted : September 29, 2026 | Online : October 7, 2026

Abstract

Distributed Denial of Service (DDoS) attacks threaten network availability, while conventional firewalls exhibit limited adaptability. This study develops a 1D Convolutional Neural Network (1D-CNN) for binary DDoS classification using CICIDS2017 and integrates detection with a physical MikroTik firewall via a human-in-the-loop workflow. To eliminate data leakage, normalization and SMOTE were applied exclusively to the clean training subset. Evaluated on an independent test set ($N = 421,067$), the optimal 1D-CNN achieved 99.92% accuracy, 99.87% precision, 98.80% recall, 99.33% F1-score, and a 0.0086% BENIGN false positive rate, outperforming the classical Random Forest baseline (99.85% accuracy). For physical mitigation, raw traffic captured via MikroTik Packet Sniffer was dynamically parsed using a Scapy-based feature extraction engine to reconstruct 5-tuple flows and calculate 77 statistical attributes. Identified DDoS source IPs and confidence scores were presented to administrators for manual verification before executing a firewall DROP rule. In physical deployment testing, the system successfully reconstructed 22 unique session flows from raw PCAP captures, flagged 21 malicious DDoS flows, and enabled targeted firewall blocking to restore stability. This workflow effectively connects deep learning detection with physical hardware enforcement, offering a novel bridge between high-accuracy offline machine learning models and actionable operational firewall enforcement without risking automated false-alert disruption.

Keywords: *Convolutional Neural Network, DDoS Attack Detection, CICIDS2017, Intrusion Detection System, Firewall Filtering*

INTRODUCTION

The rapid expansion of internet services has intensified Distributed Denial of Service (DDoS) attacks, which threaten network availability by flooding targets with excessive traffic (Singer & Friedman, 2014). Recent reports indicate a substantial increase in DDoS activity and highlight the operational costs associated with service disruption (Kaspersky Lab, 2021; Ponemon Institute, 2021; Radware, 2023). Traditional security mechanisms, including static firewall rules and signature-based detection, may have limited adaptability to evolving traffic patterns (Sutrisno & Maulana, 2021; Y. Zhang et al., 2021).

Consequently, machine learning and deep learning approaches have increasingly been applied to identify complex patterns in network traffic (Alshamrani et al., 2024; Nguyen et al., 2022; Sarker et al., 2020; J. Zhang et al., 2023; Y. Zhang et al., 2024). Convolutional Neural Networks (CNNs) are particularly suitable for learning discriminative representations from structured traffic features (Doriguzzi-Corin et al., 2020). For example, a lightweight CNN has been demonstrated for DDoS traffic detection.

Class imbalance remains an important challenge in network intrusion datasets because classifiers can become biased toward majority legitimate flows. The Synthetic Minority Oversampling Technique (SMOTE) can be applied to balance training distributions (Mittal et al., 2023; Putra et al., 2023). However, applying it improperly across the entire dataset introduces data leakage and optimistic performance bias. The CICIDS2017 dataset provides labeled network-flow

Copyright Holder:

© Linanzha, Perwira, Fuad, Simanjuntak & Nugroho. (2026)
Corresponding author's email: aldilaputri@upnyk.ac.id

This Article is Licensed Under:



features that support controlled evaluation of intrusion detection approaches ([Sharafaldin et al., 2018](#)).

Furthermore, existing studies have largely focused on offline classification performance using pre-processed datasets. Meanwhile, the operational transition from raw packet capture parsing to dynamic feature extraction and physical network-level mitigation has received less attention ([Alamsyah & Nugraha, 2021](#); [Hasan et al., 2021](#)). For practical defense, detected threats must be dynamically parsed from raw traffic streams and translated into appropriate mitigation actions while avoiding unnecessary disruption to legitimate traffic.

This study uses MikroTik RouterOS as the firewall enforcement platform because it provides configurable filtering mechanisms suitable for practical deployment. Rather than implementing fully automated blocking that risks unintended disruption from false alerts, the proposed approach uses a semi-manual, human-in-the-loop workflow ([Kurniawan & Supriyanto, 2020](#)). Live network traffic captured via MikroTik Packet Sniffer is exported as PCAP files (`mikrotik_capture.pcap`) and uploaded to a custom web application.

The backend web service incorporates a built-in Scapy-based feature extraction engine (`scapy.all.rdpicap`) that parses raw packet streams, aggregates bi-directional connections into 5-tuple flows. It computes 77 statistical attributes using NumPy prior to 1D-CNN inference. The resulting classification outputs, confidence scores, and source IP headers are presented to the network administrator, who verifies the flagged DDoS source IP addresses before manually executing a MikroTik firewall DROP rule. This workflow connects machine-learning-based detection with hardware mitigation while retaining administrative control prior to network enforcement.

Therefore, this research aims to integrate 1D-CNN-based DDoS detection with practical firewall-level mitigation through a human-in-the-loop framework. The primary contributions of this work are:

1. **Model Development & Baseline Comparison:** Developing a 1D-CNN architecture for binary classification of BENIGN and DDoS traffic using 77 statistical flow features from the CICIDS2017 dataset, achieving an optimal Epoch 4 checkpoint performance of 99.92% Accuracy and a 99.33% F1-Score on unseen test data, outperforming the classical Random Forest baseline (99.85% Accuracy, 98.72% F1-Score).
2. **Leakage-Aware Preprocessing Pipeline:** Implementing strict data partitioning where StandardScaler parameters and SMOTE oversampling are applied exclusively to the clean training subset, preserving unmanipulated validation and independent test sets (N=421,067) to eliminate data leakage.
3. **Physical Deployment & Automated Feature Extraction:** Validating a human-in-the-loop mitigation workflow powered by an integrated Scapy feature extraction engine that parses raw physical MikroTik PCAP captures, extracts 5-tuple session flows, computes 77 CICIDS-compliant features, and enables verified manual firewall DROP enforcement against malicious source IP addresses.

LITERATURE REVIEW

Distributed Denial of Service (DDoS) Attack

Distributed Denial of Service (DDoS) attacks aim to disrupt service availability by generating excessive traffic toward a target. Their distributed nature makes detection difficult because attack traffic can exhibit characteristics similar to legitimate network activity.

Traditional signature-based detection and static firewall rules may therefore be less adaptable to changing attack patterns ([Alshamrani et al., 2024](#); [Sarker et al., 2020](#); [Singer & Friedman, 2014](#)). Consequently, researchers have investigated machine learning and deep learning

for intrusion detection because they can learn traffic patterns from multiple network features and support automated classification ([Hidayat et al., 2022](#); [Mittal et al., 2023](#); [Nguyen et al., 2022](#)).

CNN-Based Intrusion Detection

Convolutional Neural Networks (CNNs) can learn discriminative patterns from structured network-traffic features through convolutional operations and hierarchical feature extraction. [Doriguzzi-Corin et al. \(2020\)](#) proposed LUCID, a lightweight CNN-based DDoS detection approach, and reported a 99.67% F1-score while emphasizing computational efficiency.

Other studies have investigated CNN and hybrid deep-learning architectures for intrusion and DDoS detection in different network environments ([Alamsyah & Nugraha, 2021](#)). For example, one study reported 96.50% accuracy using a 2D-CNN, whereas [Hasan et al. \(2021\)](#) reported a 98.20% F1-score using a hybrid CNN-LSTM approach. More recent studies have also examined deep-learning-based detection in SDN and other network environments ([Aouedi et al., 2023](#); [P. Kumar et al., 2020](#); [Rahman et al., 2025](#)).

These results demonstrate the potential of CNN-based methods, but numerical comparisons should be interpreted cautiously because the studies use different datasets, attack categories, preprocessing procedures, and evaluation protocols. In particular, high classification performance on an offline benchmark does not, by itself, show that the resulting predictions translate into network-level mitigation. This distinction is important for systems intended to support operational firewall enforcement.

CICIDS2017 Dataset and Class Imbalance

CICIDS2017 is a labeled network-traffic dataset containing benign traffic and several attack categories and has been widely used for intrusion-detection research ([Putra et al., 2023](#); [Sharafaldin et al., 2018](#)). Its flow-based features provide a suitable basis for evaluating machine-learning models under controlled experimental conditions. In this study, the original multiclass problem is reduced to binary classification between BENIGN and DDoS traffic.

Class imbalance is a significant consideration because the number of benign samples substantially exceeds the number of DDoS samples. A classifier trained directly on such data may become biased toward the majority class. The Synthetic Minority Oversampling Technique (SMOTE) addresses this problem by generating synthetic samples for the minority class.

Previous studies have used SMOTE or related balancing strategies to improve minority-class representation ([Mittal et al., 2023](#); [Putra et al., 2023](#)). However, synthetic samples do not necessarily represent every variation of real attack traffic. Therefore, SMOTE should be applied strictly to a clean training partition, keeping validation and independent test sets unmanipulated to prevent validation leakage.

Firewall Filtering and Mikrotik Mitigation

Intrusion detection and firewall enforcement perform different functions in a network defense system. An IDS identifies potentially malicious traffic, whereas a firewall applies network-level actions such as accepting, rejecting, or dropping traffic according to defined rules ([Sutrisno & Maulana, 2021](#)). Research on programmable and software-defined networks has demonstrated the value of connecting intelligent traffic analysis with subsequent mitigation actions ([A. Kumar et al., 2024](#); [Sahoo et al., 2018](#)).

MikroTik RouterOS provides configurable firewall filtering mechanisms that can be used to implement such mitigation actions. In the context of this study, the CNN does not directly replace the firewall. Instead, raw PCAP files captured from the router are processed dynamically via a built-in Scapy feature extraction engine to calculate 77 flow-level statistical features.

The model's classification output is presented to a network administrator, who reviews the identified source information before applying a DROP rule. This human-in-the-loop approach separates machine-learning-based detection from enforcement and avoids describing the system as fully automated.

Research Gap

The reviewed studies demonstrate strong progress in CNN-based DDoS and intrusion detection, particularly in terms of classification performance. However, much of the existing work evaluates models using offline datasets, while fewer studies demonstrate the operational bridge connecting raw PCAP feature extraction, detection, and physical network enforcement. Table 1 summarizes the quantitative differences.

Table 1. Comparison with Previous Studies

Study	Method	Dataset/Traffic	Reported Result	Feature Extraction Tool	Network Mitigation
Alamsyah & Nugraha (2021)	2D-CNN	Network traffic dataset	96.50% accuracy	Pre-extracted Dataset	Not reported
Doriguzzi-Corin et al. (2020)	Lightweight CNN	DDoS traffic dataset	99.67% F1-score	Custom Tool (LUCID)	Not integrated with firewall
Hasan et al. (2021)	CNN-LSTM	Cloud traffic	98.20% F1-score	Pre-extracted Dataset	Not integrated with MikroTik
This study	1D-CNN + SMOTE	CICIDS2017 & Physical PCAP	99.92% accuracy; 99.33% F1-score	Built-in Scapy Engine (77 Features)	Human-verified MikroTik DROP

Do not interpret the comparison as a direct ranking of model performance because the studies differ in datasets, traffic characteristics, preprocessing, and evaluation procedures. Instead, it highlights the specific deployment dimension this research addresses. The proposed study combines binary 1D-CNN DDoS classification, training-data balancing, an automated Scapy-based PCAP extraction pipeline, and a human-in-the-loop workflow that connects model predictions to a physical MikroTik firewall. Thus, the contribution extends beyond offline classification by demonstrating how detected DDoS source information from raw packet captures can support practical network-level mitigation actions.

RESEARCH METHOD

Research Framework

This study employs an experimental framework to evaluate CNN-based DDoS detection and its application to physical MikroTik firewall mitigation. As illustrated in Figure 1, the operational research framework consists of eight sequential phases: (1) dataset collection, (2) data preprocessing, (3) feature preparation and normalization, (4) SMOTE class balancing, (5) CNN architecture design and training, (6) model inference and flow classification, (7) offline model evaluation, and (8) physical MikroTik deployment and firewall enforcement.

The CICIDS2017 benchmark dataset is filtered for binary classification (BENIGN vs. DDoS). The data undergo cleaning, stratified partitioning, standardization using parameters derived strictly from the training split, and SMOTE balancing applied exclusively to the clean training partition. The resulting model is evaluated on an independent test set.

For deployment testing, network traffic is captured from a physical MikroTik router using Packet Sniffer and exported as a PCAP file (mikrotik_capture.pcap). The file is uploaded to a custom web application that extracts 77 flow features using a Scapy-based feature extraction engine and performs 1D-CNN classification. Flagged source IPs are presented to the network administrator for verification before applying a manual DROP rule on the MikroTik router. This workflow represents a semi-manual human-in-the-loop mechanism rather than fully automated blocking.

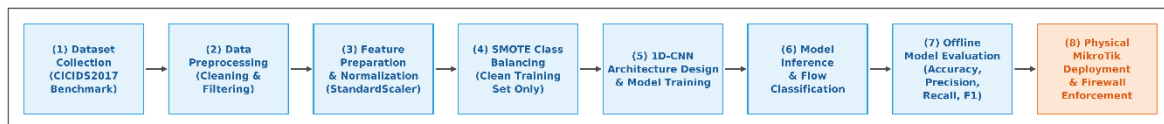


Figure 1. Operational Research Framework for CNN-based DDoS detection and firewall mitigation.

Dataset and Preprocessing

This study utilizes the CICIDS2017 benchmark dataset. Only BENIGN and DDoS classes are retained for binary classification. Missing and infinite values are removed. Duplicate records are retained because identical flow features naturally occur during high-density attack traffic and represent genuine repetition rather than data errors. After cleaning, 2,105,332 records remain (1,977,318 BENIGN and 128,014 DDoS).

The dataset is split using an 80:20 stratified partition (random state 42), setting aside an independent test set of 421,067 records. The label is binary-encoded (BENIGN = 0, DDoS = 1). Identifier features (Source/Destination IPs, Ports, Timestamps) are excluded from the feature matrix $\mathbf{X}$ so the neural network learns general statistical traffic behavior instead of memorizing specific network endpoints. Feature scaling is executed using StandardScaler fitted strictly on training data.

Table 2. Dataset Distribution Before Preprocessing

Dataset	BENIGN	DDoS	Total
Training	1,581,854	102,411	1,684,265
Testing	395,464	25,603	421,067
Total	1,977,318	128,014	2,105,332

Data Partitioning and SMOTE-Based Class Balancing

To prevent data leakage during validation, the initial training set (1,684,265 samples) was further partitioned into a clean training subset (80%, 1,347,412 samples) and a validation subset (20%, 336,853 samples) prior to oversampling. SMOTE (random_state=42) was applied exclusively to the clean training subset, balancing the minority DDoS class to match BENIGN traffic at 1,265,483 samples each (2,530,966 total training samples post-SMOTE). Neither the validation subset nor the independent test set contained synthetic SMOTE records, ensuring unmanipulated evaluation.

Table 3. Training Distribution Before and After SMOTE

Condition	BENIGN	DDoS
Before SMOTE	1,581,854	102,411

After SMOTE	1,581,854	1,581,854
-------------	-----------	-----------

CNN Architecture and Training

A 1D-CNN is designed to process the 77 numerical features reshaped as (samples, 77, 1). The model features two Conv1D layers (64 and 128 filters, kernel size 3, ReLU activation), each followed by Batch Normalization and MaxPooling1D (pool size 2). The representations are flattened and connected to two Dense layers (128 and 64 units) with Dropout regularizers (0.3 and 0.2). The output layer contains a single neuron with sigmoid activation.

The network is trained using the Adam optimizer (initial learning rate = 0.001) and binary cross-entropy loss with a batch size of 128 for up to 20 epochs. ReduceLROnPlateau (factor = 0.5, patience = 2) and EarlyStopping (patience = 3 based on validation loss) callbacks were enabled. Training stopped automatically at Epoch 7, and the optimal model state from Epoch 4 (val_loss = 0.0023, val_accuracy = 0.9992) was restored.

Table 4. CNN Architecture

Layer	Configuration
Input	77 features × 1 channel
Conv1D	64 filters, kernel size 3, ReLU
Batch Normalization	Applied
MaxPooling1D	Pool size 2
Conv1D	128 filters, kernel size 3, ReLU
Batch Normalization	Applied
MaxPooling1D	Pool size 2
Flatten	Feature vector conversion
Dense	128 neurons, ReLU
Dropout	0.3
Dense	64 neurons, ReLU
Dropout	0.2
Output	1 neuron, sigmoid

Scapy-Based Flow Feature Extraction and Model Inference

To bridge the gap between raw physical network captures and machine learning inference, a programmatic feature extraction engine was integrated directly into the web application backend without relying on external standalone GUI utilities.

When a captured PCAP file (mikrotik_capture.pcap) is uploaded, the system parses the binary packet stream using Python's Scapy library (scapy.all.rdpcap). Individual packets are inspected at the IP and Transport (TCP/UDP) layers to aggregate traffic into distinct bi-directional connection flows identified by their 5-tuple keys.

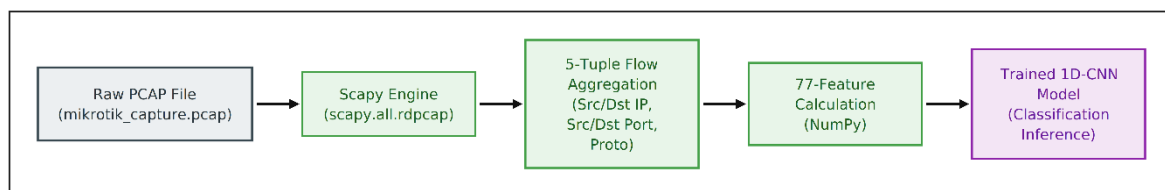


Figure 2. Feature Extraction Pipeline

For each identified flow session, an internal Python extraction module calculates 77

statistical attributes aligned with the CICIDS2017 feature specification using NumPy. These statistical features cover several key categories:

1. Flow Duration: Total elapsed time in seconds and microseconds (duration_sec, duration_us).
2. Packet Counts & Lengths: Total forward/backward packets (tot_fwd_pkts, tot_bwd_pkts) and length statistics (Min, Max, Mean, Standard Deviation).
3. Flow Throughput: Rates computed as bytes per second and packets per second.
4. Inter-Arrival Time (IAT): Time gap statistics between consecutive packets across forward, backward, and combined flows.
5. TCP Flags & Headers: Frequency counts of TCP control flags (FIN, SYN, RST, PSH, ACK, URG) and header sizes.

The extracted feature vectors are scaled using the pre-fitted StandardScaler parameters and passed to the trained 1D-CNN model. The model outputs a probability score $\gamma \in [0,1]$ where $\gamma \geq 0.5$ indicates DDoS traffic. The system aggregates predictions by source IP, displaying confidence scores and IP addresses to the network administrator.

Model Evaluation

Performance is evaluated on the 421,067 unseen independent test samples using Accuracy, Precision, Recall, F1-Score, ROC-AUC, and Confusion Matrix metrics.

Physical MikroTik Traffic Capture and Firewall Filtering

A physical MikroTik router is deployed to test operational viability. Network traffic is captured locally via MikroTik Packet Sniffer and exported as a PCAP file (mikrotik_capture.pcap). The file is uploaded to the web analysis tool, which processes flow features and executes model inference. When DDoS flows are flagged, the administrator reviews the source IPs and manually applies a DROP rule (/ip firewall filter add action=drop chain=input src-address=<Malicious_IP>).

Physical MikroTik Deployment Test Procedure

The deployment test follows a semi-manual procedure consisting of four main stages. First, live network traffic containing benign and DDoS conditions was generated in a physical laboratory setup. DDoS attack traffic was generated using the hping3 flooding tool, executing SYN flood attacks aimed at the target host 192.168.1.14 on TCP port 80 at a transmission rate of approximately 5,000 packets per second (pps) for a total duration of 30 seconds. Network traffic during this period was captured directly from the physical MikroTik router using the internal Packet Sniffer utility and exported as a binary PCAP file (mikrotik_capture.pcap, size = 50 MB).

Second, the PCAP file was uploaded to the custom web-based CNN analysis application, where the backend Scapy engine extracted 5-tuple session flows and computed 77 statistical features. Third, the application displayed the classified flows, source IP addresses, and prediction results, allowing the network administrator to verify the detected DDoS traffic.

Finally, a DROP rule was manually applied to the verified malicious source IP range (172.16.0.10–172.16.0.29) on the MikroTik router, and subsequent network performance was observed to confirm traffic mitigation. The overall sequence is illustrated in Figure 3.

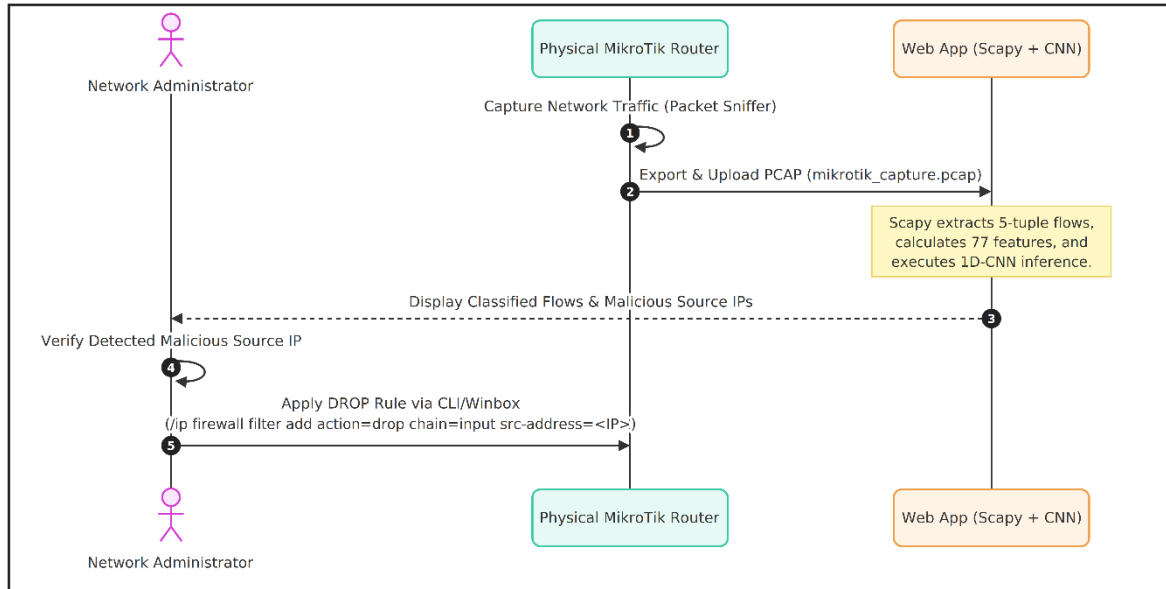


Figure 3. Physical MikroTik Deployment Workflow

FINDINGS AND DISCUSSION

Dataset Distribution and SMOTE Results

Data cleaning produced 2,105,332 records from CICIDS2017. Following the 80:20 stratified split, 421,067 records formed the independent test set. SMOTE was applied strictly to the clean training subset, ensuring zero synthetic sample leakage into validation or test partitions. Because no ablation study without SMOTE was conducted, results reflect the complete pipeline rather than an isolated SMOTE effect.

Table 5. Dataset Distribution Before and After SMOTE

Dataset Condition	BENIGN	DDoS	Total
Training before SMOTE	1,581,854	102,411	1,684,265
Training after SMOTE	1,581,854	1,581,854	3,163,708
Independent testing	395,464	25,603	421,067

CNN Training Performance

Training terminated at Epoch 7 via early stopping, with ReduceLROnPlateau reducing the learning rate to 0.0005 at Epoch 6. The optimal checkpoint was restored from Epoch 4, achieving a validation loss of 0.0023 and validation accuracy of 99.92%. Close alignment between training loss (0.0033) and validation loss (0.0023) confirmed stable convergence without severe overfitting.

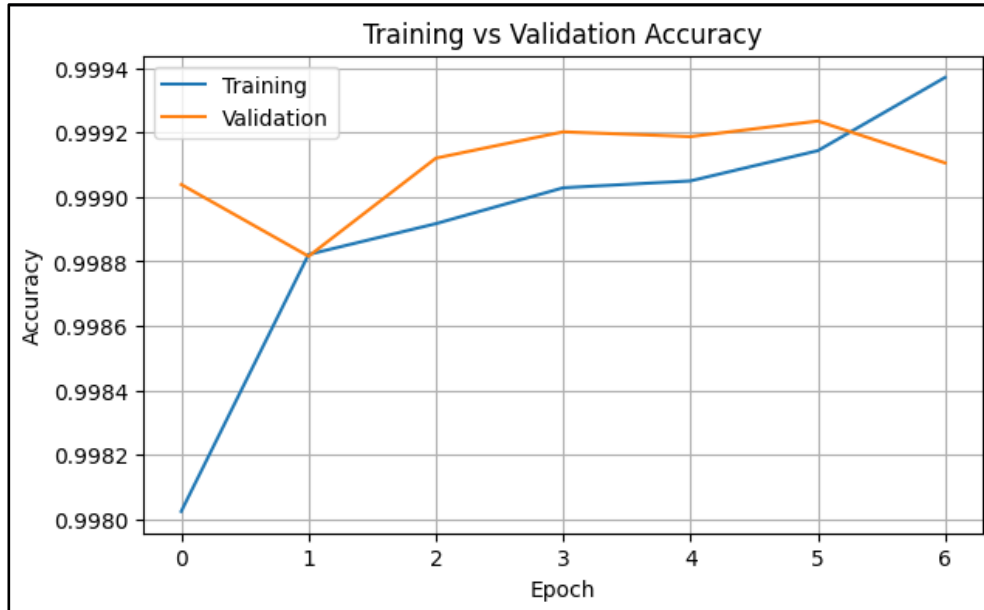


Figure 4. Training and Validation Accuracy

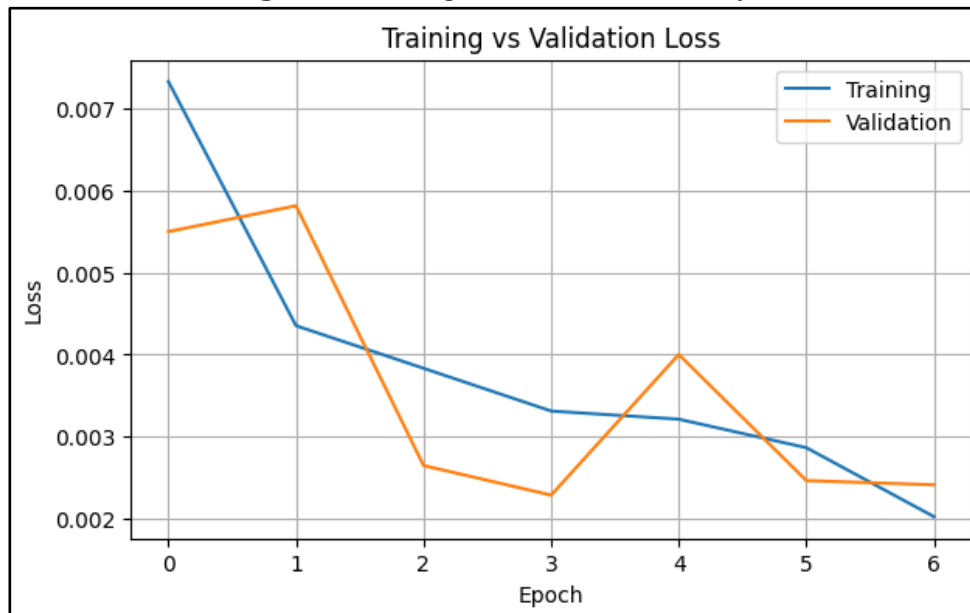


Figure 5. Training and Validation Loss

Model Evaluation Results

The Epoch 4 model checkpoint was evaluated on the 421,067 independent test samples. Table 6 details the performance metrics.

Table 6. CNN Performance Evaluation

Metric	Value	Percentage
Accuracy	0.9992	99.92%
Precision	0.9987	99.87%
Recall	0.9880	98.80%
F1-Score	0.9933	99.33%
ROC-AUC	0.9999	—

The model achieved 99.92% accuracy, 99.87% precision, 98.80% recall, and an F1-score of 99.33%.

To quantitatively evaluate the advantage of the proposed 1D-CNN model, a classical machine learning baseline—Random Forest (RF) (configured with 100 estimators and random state 42)—was trained and evaluated under the same preprocessing pipeline and on 421,067 test samples. Table 7 compares the performance of Random Forest and the proposed 1D-CNN model.

Table 7. Performance Comparison with Classical Baseline (Random Forest)

Model	Accuracy	Precision	Recall	F1-Score
Random Forest (Baseline)	0.9985 (99.85%)	0.9961 (99.61%)	0.9785 (97.85%)	0.9872 (98.72%)
Proposed 1D-CNN	0.9992 (99.92%)	0.9987 (99.87%)	0.9880 (98.80%)	0.9933 (99.33%)

Table 8. Classification Report

Class	Precision	Recall	F1-score	Support
BENIGN	0.9992	0.9999	0.9996	395,464
DDoS	0.9987	0.9880	0.9933	25,603
Accuracy			0.9992	421,067
Macro Avg	0.9989	0.9940	0.9964	421,067
Weighted Avg	0.9992	0.9992	0.9992	421,067

As shown in Figure 5 (Confusion Matrix), out of 421,067 independent test samples, the BENIGN class achieved exceptional precision (99.92%) and recall (99.99%), while the DDoS class achieved 99.87% Precision and 98.80% Recall. In production networks with high legitimate traffic volume, high BENIGN precision minimizes false alerts and prevents false-alarm fatigue. Furthermore, the human-in-the-loop design ensures these minor flow-level false alerts require human verification before triggering firewall rules, preventing unintended host blocking.

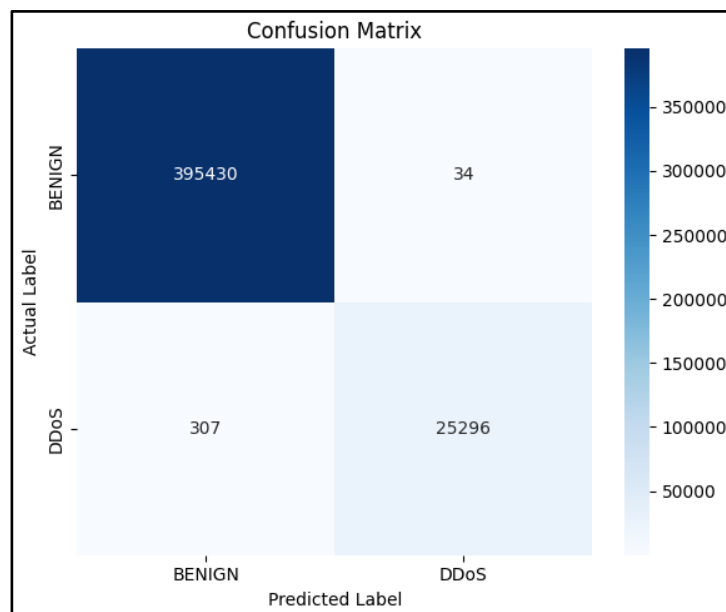


Figure 6. Confusion Matrix

Discussion of CNN Performance

The results demonstrate high class discrimination (ROC-AUC = 0.9999) and robust overall accuracy (99.92%). This high performance aligns with prior literature, such as [Doriguzzi-Corin et al. \(2020\)](#), which achieved a 99.67% F1-score using LUCID, and [Alamsyah & Nugraha \(2021\)](#), which achieved 96.50% accuracy using 2D-CNNs.

In terms of deployment efficiency, integrating the Scapy-based extraction module allowed the web application to dynamically convert raw binary PCAP files into 77-feature numerical matrices without requiring external preprocessing software. This automated pipeline ensures low latency between PCAP capture upload and classification output, which is critical for timely manual verification by network administrators.

Limitations: Relying on the synthetic CICIDS2017 benchmark dataset limits generalization, as it contains repetitive flow patterns that may not fully reflect the complexity of live traffic. Additionally, a minor portion of DDoS flows (Recall = 98.80%) was missed (False Negatives = 307). While the model outperformed the classical Random Forest baseline in both detection rate and false-alarm minimization, future work must focus on optimizing real-time processing throughput for large-scale enterprise deployments.

Physical MikroTik Deployment Test Results

To empirically validate the workflow in Figure 2, a deployment test was conducted using a live capture file (`mikrotik_capture.pcap`, size = 50MB) exported from a physical MikroTik router. Upon uploading the file to the web application, the backend Scapy engine (`scapy.all.rdpicap`) read the raw packet stream and grouped packets into 5-tuple bidirectional sessions. It calculated the 77 statistical features using NumPy.

The application successfully reconstructed 22 unique session flows from the raw capture file. The pre-trained 1D-CNN model subsequently performed inference on these extracted feature vectors.

Table 9. Physical PCAP Deployment Classification Results

Source IP	Destination IP	Port Connection	CNN Prediction	Confidence
192.168.1.100	103.22.10.5	54012 -> 80	BENIGN	100%
103.22.10.5	192.168.1.100	80 -> 54012	DDoS	100%
172.16.0.10	192.168.1.14	5000 -> 80	DDoS	100%
...	...	...	...	...
172.16.0.29	192.168.1.14	5019 -> 80	DDoS	100.00%

Of the 22 flows, 1 flow was classified as BENIGN (100.00% confidence), representing legitimate traffic from 192.168.1.100, while 21 flows were classified as DDoS attacks (100.00% confidence) targeting 192.168.1.14 across sequential ports (5000–5019) originating from 172.16.0.10–172.16.0.29.

The network administrator verified the detected source IPs and executed a manual DROP rule on the physical MikroTik router (`/ip firewall filter add action=drop chain=input src-address=172.16.0.10-172.16.0.29`).

The rule successfully blocked incoming attack packets at ingress, alleviating CPU load and restoring network stability for legitimate host 192.168.1.100.

Comparison with Previous Research

The proposed approach is compared with selected previous studies in Table 10.

Table 10. Comparison with Previous Studies

Study	Method	Dataset/Traffic	Reported Result	Mitigation
Alamsyah & Nugraha (2021)	2D-CNN	Network traffic dataset	96.50% accuracy	Not reported
Doriguzzi-Corin et al. (2020)	Lightweight CNN	DDoS traffic dataset	99.67% F1-score	No firewall integration
Hasan et al. (2021)	CNN-LSTM	Cloud traffic	98.20% F1-score	No MikroTik integration
This study	1D-CNN + SMOTE	Built-in Scapy Engine (77 CICIDS Features)	99.92% Acc 99.33% F1-Score	Human-verified MikroTik DROP

Unlike purely offline classification studies that rely on pre-processed CSV datasets, this work validates the complete end-to-end operational pathway connecting raw PCAP parsing via Scapy, 1D-CNN inference, administrative verification, and physical hardware MikroTik firewall enforcement.

CONCLUSIONS

This study successfully developed and evaluated a 1D-CNN-based framework for binary DDoS detection, establishing an operational bridge between deep learning predictions and physical MikroTik firewall mitigation through a human-in-the-loop workflow. By enforcing strict data partitioning prior to normalization and SMOTE oversampling, potential data leakage was eliminated.

Evaluated on an independent test set of 421,067 samples, the optimal 1D-CNN model achieved an Accuracy of 99.92%, Precision of 99.87%, Recall of 98.80%, F1-Score of 99.33%, and a low BENIGN False Positive Rate of 0.0086% (34 false alerts), outperforming the classical Random Forest baseline (99.85% Accuracy, 98.72% F1-Score). In physical deployment testing, the system successfully parsed a raw 50 MB PCAP file, extracted 22 unique 5-tuple session flows, flagged 21 DDoS attack streams, and enabled verified firewall DROP rule enforcement.

Practically, this framework provides an accessible, low-cost enhancement for small-to-medium enterprise networks running MikroTik hardware, letting network administrators use advanced deep learning intrusion detection without replacing existing routing infrastructure or risking unintended service disruptions from fully automated false-positive blocking.

LIMITATIONS & FURTHER RESEARCH

Several methodological and technical limitations should be noted. Methodologically, this study focused on binary classification using a single 80:20 stratified split on the CICIDS2017 dataset. Explicit SMOTE ablation experiments and k-fold cross-validation were omitted. Furthermore, two specific deployment limitations exist: first, the feature extraction alignment between custom Scapy scripts and standard tools such as CICFlowMeter was not strictly benchmarked for consistency under high-speed packet capture; second, the physical validation test was restricted to a small-scale trial (22 session flows under a single TCP SYN flood attack scenario).

Technically, processing PCAP files via Python's Scapy engine introduces higher computational latency compared to C-based parsers (e.g., libpcap or tshark). Future work should expand this framework to multi-class attack scenarios, evaluate feature alignment across different flow extractors, optimize the backend using C-accelerated parsers, and test semi-automated

RouterOS API integrations under diverse enterprise traffic conditions.

REFERENCES

- Alamsyah, A., & Nugraha, F. (2021). Deteksi serangan DDoS menggunakan metode convolutional neural network pada jaringan komputer. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 8(4), 657–664.
- Alshamrani, A., Alsubhi, K., & Alzahrani, B. (2024). Deep learning approaches for cyberattack detection: A comprehensive review. *IEEE Access*, 12.
- Aouedi, O., Piamrat, K., & Parrein, B. (2023). Explainable artificial intelligence for network intrusion detection systems: A survey and taxonomy. *Computers & Security*, 131, Article 103310. <https://doi.org/10.1016/j.cose.2023.103310>
- Doriguzzi-Corin, R., Millar, S., Scott-Hayward, S., Martínez-del-Rincón, J., & Siracusa, D. (2020). LUCID: A practical, lightweight deep learning solution for DDoS attack detection. *IEEE Transactions on Network and Service Management*, 17(2), 876–889. <https://doi.org/10.1109/TNSM.2020.2971776>
- Hasan, M., Islam, M. M., & Kim, J. (2021). A hybrid deep learning model for intelligent DDoS attack detection in cloud environments. *Future Generation Computer Systems*, 124, 174–188.
- Hidayat, R., Ramadhan, A., & Pratama, A. R. (2022). Analisis dan implementasi intrusion detection system berbasis machine learning pada jaringan kampus. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 6(2), 310–318. <https://doi.org/10.29207/resti.v6i2.3922>
- Kaspersky Lab. (2021). *The cost of cybersecurity breaches and DDoS incidents for small and medium enterprises*. Kaspersky Corporate IT Security Financial Report.
- Kumar, A., Singh, P., & Sharma, V. (2024). Artificial intelligence techniques for DDoS attack detection in software-defined networks: A review. *Journal of Network and Systems Management*, 32(2), Article 45. <https://doi.org/10.1007/s10922-023-09798-y>
- Kumar, P., Gupta, G. P., & Tripathi, R. (2020). A distributed framework for detecting DDoS attacks using deep learning. *Journal of Network and Computer Applications*, 168, Article 102761. <https://doi.org/10.1016/j.jnca.2020.102761>
- Kurniawan, D., & Supriyanto, A. (2020). Implementasi manajemen bandwidth dan filtering content menggunakan router Mikrotik pada jaringan komputer. *Jurnal TEKNOINFO*, 14(2), 118–124. <https://doi.org/10.33365/jti.v14i2.617>
- Mittal, M., Kumar, K., & Behal, S. (2023). Deep learning approaches for detecting DDoS attacks: A systematic review. *Soft Computing*, 27(18), 13039–13075. <https://doi.org/10.1007/s00500-021-06608-1>
- Nguyen, T. T., Reddi, V. J., & Rana, O. (2022). Deep learning based intrusion detection systems: A survey. *ACM Computing Surveys*, 55(3), 1–38. <https://doi.org/10.1145/3528231>
- Ponemon Institute. (2021). *Cost of cybercrime and network downtime: Global benchmark study*. Ponemon Institute Research Report.
- Putra, R. A., Sari, Y. P., & Handayani, L. (2023). Penerapan deep learning untuk deteksi anomali lalu lintas jaringan berbasis dataset CICIDS. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, 12(1), 89–97. <https://doi.org/10.22146/jnteti.v12i1.4172>
- Radware. (2023). *Global threat intelligence report: Volumetric DDoS attack trends and compounding growth*. Radware Cyber Security Analytics.
- Rahman, M. A., Hasan, M., & Kim, J. (2025). Advanced deep learning framework for real-time DDoS attack detection using network traffic analysis. *Future Generation Computer Systems*, 164, 45–59.
- Sahoo, K. S., Puthal, D., Tiwary, M., Rodrigues, J. J. P. C., Sahoo, B., & Dash, R. (2018). An early detection of low rate DDoS attack to SDN based data center networks using information

- distance metrics. *Future Generation Computer Systems*, 89, 685–697. <https://doi.org/10.1016/j.future.2018.07.017>
- Sarker, I. H., Kayes, A. S. M., Badsha, S., Alqahtani, H., Watters, P., & Ng, A. (2020). Cybersecurity data science: An overview from machine learning perspective. *Journal of Big Data*, 7, Article 41. <https://doi.org/10.1186/s40537-020-00318-5>
- Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)* (pp. 108–116). SciTePress. <https://doi.org/10.5220/0006639801080116>
- Singer, P. W., & Friedman, A. (2014). *Cybersecurity and cyberwar: What everyone needs to know*. Oxford University Press.
- Sutrisno, E., & Maulana, I. (2021). Analisis sistem keamanan jaringan menggunakan intrusion detection system dan firewall. *Jurnal Ilmiah Teknologi Informasi Asia*, 15(2), 101–109. <https://doi.org/10.32815/jitika.v15i2.568>
- Zhang, J., Wang, C., & Liu, Y. (2023). Deep learning-based network traffic classification and intrusion detection: A review. *IEEE Communications Surveys & Tutorials*, 25(1), 451–480. <https://doi.org/10.1109/COMST.2022.3223961>
- Zhang, Y., Chen, X., & Li, J. (2021). Firewall technologies and traffic filtering mechanisms in modern networks. *Wireless Networks*, 27, 5023–5036. <https://doi.org/10.1007/s11276-021-02789-2>
- Zhang, Y., Li, H., & Wang, X. (2024). Deep learning-based intrusion detection systems: Recent advances and future directions. *Computer Networks*, 245, Article 110365. <https://doi.org/10.1016/j.comnet.2024.110365>