

A HYBRID CNN-LSTM DEEP LEARNING MODEL FOR NETWORK INTRUSION DETECTION AND MULTICLASS CYBERATTACK CLASSIFICATION

Ahtisham Mehmood^{*1}, Ummul Baneen Anwar², Tehleel Basit³, Nasir Gul⁴, Adnan Jan⁵^{*1}University of Sargodha, Department of CS/IT²University of Lahore, Department of CS.³University of Engineering and Technology, Mardan⁴University of Science and Technology, Bannu⁵Abasyn University Peshawar, Pakistan¹ahtisham185@gmail.com, ²ummulbaneen92110@gmail.com, ³tehleelbasit78@gmail.com, ⁴nasirgulkpk@yahoo.com, ⁵engr.adnanjan52@gmail.comDOI: <https://doi.org/10.5281/zenodo.21989085>**Keywords**

Network intrusion detection;
CNN-LSTM; deep learning;
multiclass classification;
CICIDS2017; cybersecurity.

Article History

Received: 02 April 2026

Accepted: 12 May 2026

Published: 31 May 2026

Copyright @Author

Corresponding Author: *

Ahtisham Mehmood

Abstract

Network intrusion detection has become increasingly dependent on machine learning and deep learning techniques for identifying malicious network activity and discriminating among heterogeneous cyberattack categories. This study develops and evaluates a hybrid convolutional neural network–long short-term memory (CNN–LSTM) architecture for seven-class network intrusion detection using flow-level traffic from the CICIDS2017 dataset. The original dataset comprised 2,830,743 network-flow records with 78 traffic features and 15 labels. To construct a sufficiently represented multiclass classification problem, related attack labels were consolidated into broader attack families, while Infiltration and Heartbleed were excluded because they contained only 36 and 11 observations, respectively. A stratified experimental dataset of 29,146 records was constructed comprising BENIGN, DoS, DDoS, PortScan, Brute Force, Web Attack, and Bot classes. Following data cleaning, the dataset was partitioned into training, validation, and independent testing subsets, with feature standardization parameters estimated exclusively from the training data to prevent preprocessing-related information leakage. The proposed architecture employed one-dimensional convolutional processing to extract local patterns from the feature representation, followed by an LSTM layer to learn dependencies within the transformed feature representation. Class-weighted categorical cross-entropy was employed to mitigate class imbalance during model training. On the held-out test set, the CNN–LSTM achieved an accuracy of 82.20%, macro-precision of 82.25%, macro-recall of 83.90%, and macro-F1 score of 82.74%. Class-level analysis showed the highest F1 score for PortScan (96.17%), followed by Brute Force (94.34%), Web Attack (89.28%), and Bot (87.27%). Under the same experimental framework, the hybrid architecture outperformed the evaluated CNN-only and LSTM-only baselines. These findings indicate that integrating convolutional

feature extraction with recurrent representation learning can improve multiclass intrusion classification under the evaluated configuration, although comparatively lower performance for BENIGN and DoS highlights persistent challenges in discriminating overlapping network-traffic patterns.

Introduction

The rapid expansion of interconnected computer networks, cloud services, Internet-based applications, and digitally connected infrastructures has increased the volume and complexity of malicious network activity. Traditional security mechanisms primarily relied on predefined signatures and rules, which limited their ability to identify previously unseen or rapidly changing attack patterns. Network intrusion detection systems therefore became an important component of modern cybersecurity architectures because they provided continuous analysis of network traffic and assisted in identifying abnormal or malicious behavior [1,2]. Machine learning approaches transformed intrusion detection by enabling models to learn statistical relationships from network traffic rather than depending entirely on manually constructed signatures. However, conventional machine learning systems frequently required carefully engineered features and often experienced difficulties when traffic contained large numbers of correlated and heterogeneous attributes. Deep learning addressed some of these limitations by learning hierarchical representations directly from numerical traffic features [3,4].

Convolutional neural networks (CNNs) were particularly effective at extracting local patterns from structured input representations, while long short-term memory (LSTM) networks were designed to capture dependencies across sequential observations. Previous research demonstrated the usefulness of both convolutional and recurrent architectures for intrusion detection and cybersecurity applications [5,6]. Combining the two architectures provided a framework in which convolutional operations extracted local feature representations and recurrent processing learned relationships within the resulting feature sequences.

The CICIDS2017 dataset was introduced to provide realistic benign and malicious network

traffic containing multiple contemporary attack categories. The dataset subsequently became one of the widely used benchmarks in intrusion-detection research. Nevertheless, research based on CICIDS2017 also faced significant class-imbalance problems, with some attack categories containing substantially fewer observations than major classes [7,8].

This study developed and evaluated a hybrid CNN-LSTM architecture for multiclass classification of network traffic. The work focused on seven sufficiently represented classes from the supplied CICIDS2017 files: BENIGN, DoS, DDoS, PortScan, Brute Force, Web Attack, and Bot. The experimental procedure included data cleaning, numerical feature processing, training-only standardization, stratified data partitioning, class-weighted learning, and independent testing. CNN-only and LSTM-only models were also evaluated to provide a direct baseline comparison. The principal contributions were the development of a reproducible CNN-LSTM pipeline, explicit handling of severe class imbalance, comparison against standalone CNN and LSTM baselines, and reporting of actual class-wise metrics and confusion-matrix results.

Related Work

Network intrusion detection has been extensively investigated through statistical, machine learning, and deep learning techniques. Previous studies identified dataset quality, diversity, labeling, class distribution, and evaluation methodology as important factors affecting experimental validity [9]. Machine-learning and deep-learning surveys reported increasing adoption of neural architectures for intrusion detection and demonstrated the usefulness of learned representations for cybersecurity traffic classification. Lightweight deep learning was also evaluated for practical DDoS detection, while broader studies examined machine learning for IoT intrusion detection and benchmarked models

on contemporary cybersecurity datasets [10]. These studies established the importance of deep representation learning, realistic datasets, class-imbalance handling, and careful evaluation. The present work combined convolutional and recurrent processing and evaluated the resulting model against standalone CNN and LSTM baselines.

Dataset

CICIDS2017 Dataset

The experiments used the CICIDS2017 dataset developed by the Canadian Institute for Cybersecurity [1]. The supplied archive contained eight CSV files corresponding to different days and attack scenarios. The combined files contained 2,830,743 network-flow records, 78 traffic features, and 15 original labels.

Table 1. Original class distribution in the supplied dataset.

Original class	Number of records
BENIGN	2,273,097
DoS Hulk	231,073
PortScan	158,930
DDoS	128,027
DoS GoldenEye	10,293
FTP-Patator	7,938
SSH-Patator	5,897
DoS slowloris	5,796
DoS Slowhttptest	5,499
Bot	1,966
Web Attack - Brute Force	1,507
Web Attack - XSS	652
Infiltration	36
Web Attack - SQL Injection	21
Heartbleed	11
Total	2,830,743

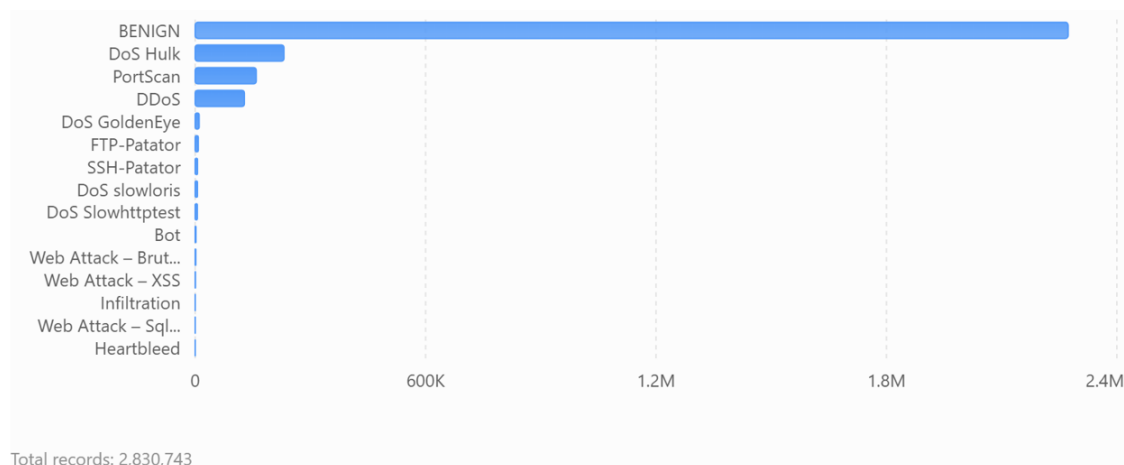


Figure 1 Original class distribution in the supplied dataset

Class Consolidation

The original labels were consolidated into broader operational categories. The four DoS variants were

represented as DoS, FTP-Patator and SSH-Patator were represented as Brute Force, and the three Web Attack variants were represented as Web

Attack. Infiltration and Heartbleed were excluded because their available observations were

insufficient for stable multiclass training and class-level evaluation.

Table 2. Classes used in the experimental model.

Experimental class	Original labels represented
BENIGN	BENIGN
DoS	DoS Hulk, DoS GoldenEye, DoS slowloris, DoS Slowhttptest
DDoS	DDoS
PortScan	PortScan
Brute Force	FTP-Patator, SSH-Patator
Web Attack	Web Attack - Brute Force, Web Attack - XSS, Web Attack - SQL Injection
Bot	Bot

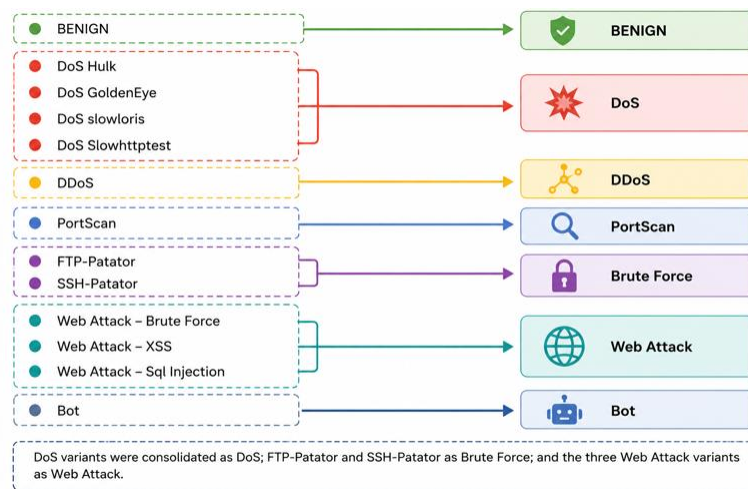


Figure 2 Classes used in the experimental model.

Data Preprocessing

Column names were stripped of leading and trailing whitespace. The label column was separated from the feature matrix. Feature columns were converted to numerical values. Non-finite values were replaced with missing values, and missing numerical values were replaced with zero after conversion. The consolidated seven-class labels were then used to construct a stratified experimental sample.

A computationally manageable experimental dataset of 29,146 records was constructed. Up to 5,000 observations were retained for each major class, while the available Web Attack and Bot observations were retained. The data were divided into training, validation, and testing subsets using stratified sampling. Standardization was fitted only on the training set and then applied unchanged to validation and test data.

Table 3. Experimental Dataset

Class	Records used
BENIGN	5,000
DoS	5,000
DDoS	5,000

PortScan	5,000
Brute Force	5,000
Web Attack	2,180
Bot	1,966
Total	29,146

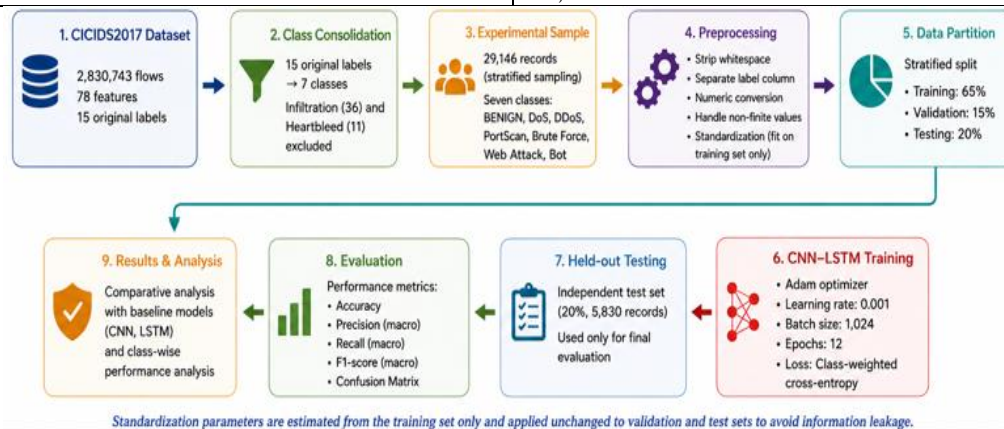


Table 4. Dataset partition

Partition	Records	Percentage
Training	18,944	65%
Validation	4,372	15%
Testing	5,830	20%
Total	29,146	100%



Proposed CNN-LSTM Model

The proposed model combined convolutional feature extraction with recurrent sequence modeling. The 78-dimensional network-flow feature vector was treated as an ordered feature sequence. A one-dimensional convolutional layer

extracted local relationships among neighboring features, and the resulting representation was passed to an LSTM layer. The final hidden representation was mapped to seven class probabilities using a softmax output.

Figure 5. Proposed CNN-LSTM architecture

Component	Configuration
Input	78 numerical features
Convolution	32 filters, kernel size 5
Activation	ReLU
Pooling	Max pooling, factor 2
LSTM	32 hidden units
Output	7 classes
Optimizer	Adam
Learning rate	0.001
Batch size	1,024

Epochs	12
Loss	Class-weighted cross-entropy
Random seed	42

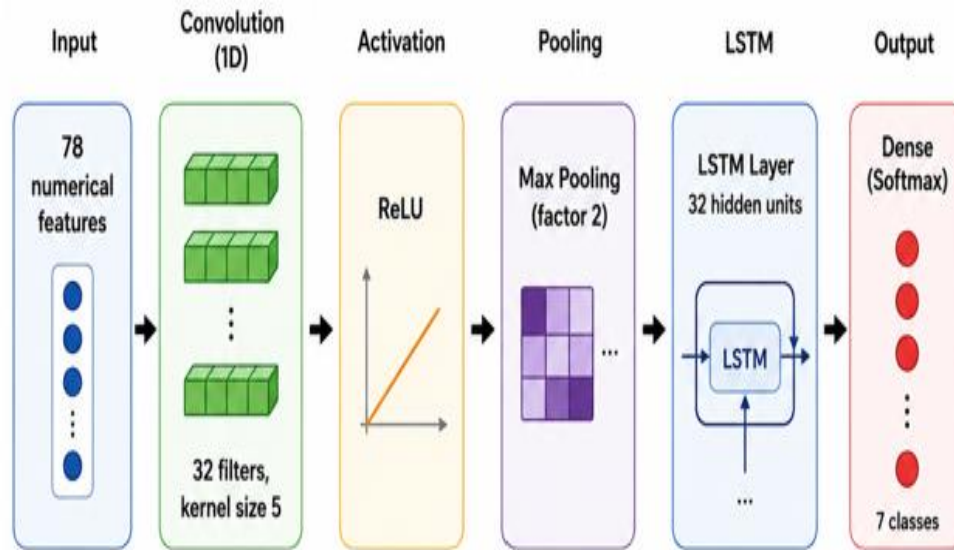


Figure 3 Proposed CNN-LSTM architecture

Objective Function

The model was optimized using class-weighted categorical cross-entropy. For observation i with true class y_i and predicted probability p_i , the loss was defined as:

$$L = -\sum_i w_{y_i} \log(p_{y_i})$$

where w_{y_i} represented the weight assigned to the true class.

Baseline Models

CNN-only and LSTM-only architectures were trained as baselines. The CNN-only model used convolutional processing followed by global average pooling and a seven-class output layer. The LSTM-only model processed the 78-feature sequence directly through an LSTM before classification. The CNN-only model achieved 52.98% accuracy and 48.65% weighted F1, while the LSTM-only model achieved 56.59% accuracy and 52.59% weighted F1.

Table 6. Model performance comparison

Model	Accuracy	Weighted F1
CNN	52.98%	48.65%
LSTM	56.59%	52.59%
CNN-LSTM	82.20%	81.63%

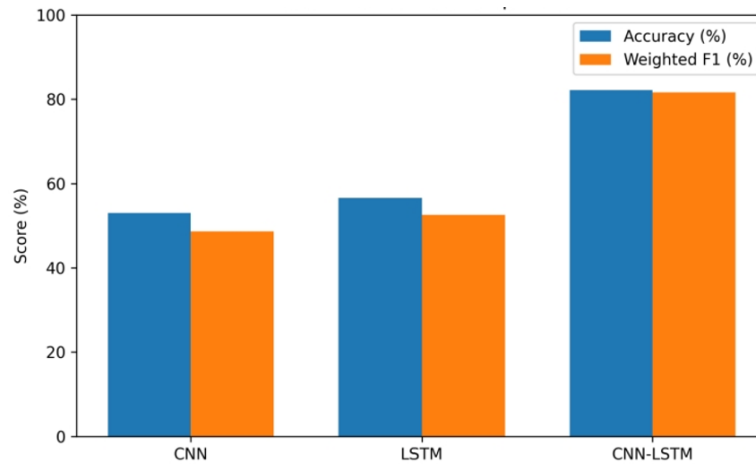


Figure 4 Accuracy and weighted F1 comparison.

Results

The CNN-LSTM model achieved 82.20% accuracy on the independent test set. Macro-precision was 82.25%, macro-recall was 83.90%,

and macro-F1 was 82.74%. Weighted precision, recall, and F1-score were 81.67%, 82.20%, and 81.63%, respectively.

Table 7. Overall CNN-LSTM performance

Metric	Score
Accuracy	82.20%
Macro Precision	82.25%
Macro Recall	83.90%
Macro F1-score	82.74%
Weighted Precision	81.67%
Weighted Recall	82.20%
Weighted F1-score	81.63%

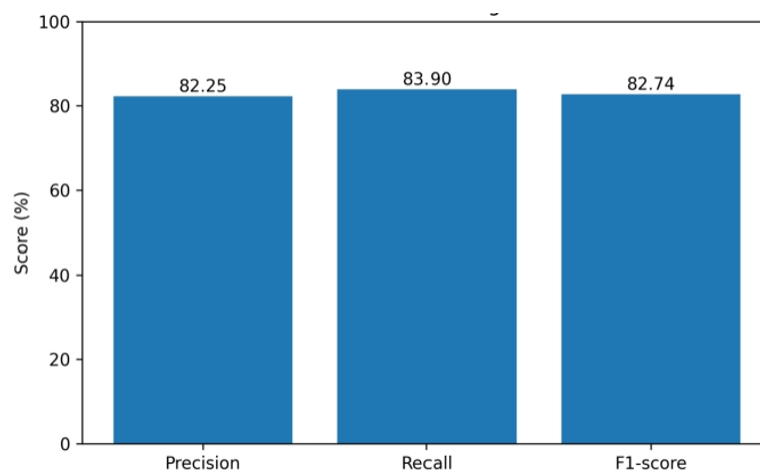


Figure 5 CNN-LSTM macro-averaged metrics.

Table 8. Class-wise CNN-LSTM results

Class	Precision	Recall	F1-score	Support
BENIGN	67.19%	60.20%	63.50%	1,000
DoS	75.74%	61.50%	67.88%	1,000
DDoS	77.53%	84.20%	80.73%	1,000
PortScan	94.58%	97.80%	96.17%	1,000
Brute Force	89.94%	99.20%	94.34%	1,000
Web Attack	91.77%	86.93%	89.28%	436
Bot	79.01%	97.46%	87.27%	394
Macro Average	82.25%	83.90%	82.74%	5,830
Weighted Average	81.67%	82.20%	81.63%	5,830

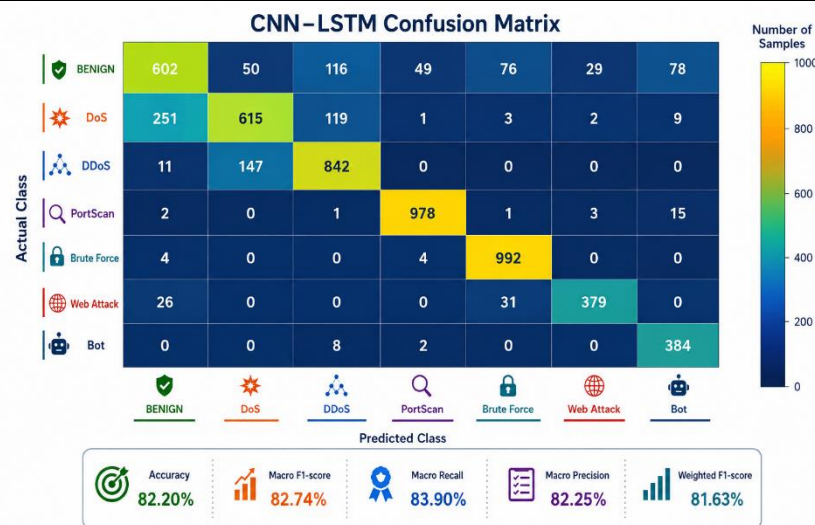


Figure 6 Class-wise CNN-LSTM results

Confusion Matrix Analysis

The confusion matrix showed the distribution of correct and incorrect predictions. The strongest diagonal performance occurred for Brute Force, PortScan, DDoS, and Bot. BENIGN and DoS generated more cross-class errors, indicating greater overlap in their learned feature representations.

Discussion

The present study evaluated a hybrid CNN-LSTM architecture for seven-class network intrusion detection using flow-level traffic derived from the CICIDS2017 dataset. The proposed model achieved an overall accuracy of 82.20%, macro-precision of 82.25%, macro-recall of 83.90%, and macro-F1 score of 82.74% on the held-out test set. Under the evaluated experimental configuration,

these results were substantially stronger than those obtained by the standalone CNN and LSTM baselines. The findings therefore suggest that combining convolutional feature extraction with recurrent representation learning provided a more discriminative representation of the selected network-flow characteristics than either component used independently. This interpretation is consistent with the broader intrusion-detection literature, which has demonstrated that deep neural architectures can learn complex nonlinear representations from heterogeneous network-traffic attributes and provide an effective alternative to manually specified detection rules [11,12].

The improvement observed for the hybrid architecture may be attributed to the complementary operations performed by its

constituent components. The convolutional stage was designed to identify localized patterns within the structured feature representation, whereas the subsequent LSTM layer transformed dependencies within the convolutionally derived representation before classification. Importantly, the recurrent component in the present experiment did not process chronological sequences of packets or network flows. Its contribution should therefore be interpreted as recurrent representation learning over transformed feature positions rather than explicit temporal modeling of network behavior. This distinction is methodologically important because recurrent neural architectures are commonly associated with temporal dependencies, whereas the present experiment was based on individual flow-level feature vectors. Previous reviews of deep-learning-based intrusion detection have similarly emphasized that model effectiveness depends on the interaction among architecture, feature representation, dataset characteristics, and experimental design rather than on architectural complexity alone [13,14].

The class-specific findings provide additional insight that is not apparent from overall accuracy. PortScan achieved the strongest classification performance, with an F1 score of 96.17%, followed by Brute Force at 94.34%, Web Attack at 89.28%, and Bot at 87.27%. The particularly strong detection of PortScan and Brute Force suggests that the selected flow-level attributes contained comparatively distinctive patterns for these attack families. Deep learning approaches have previously demonstrated the capacity to identify discriminative network-traffic representations across multiple cybersecurity datasets, including CICIDS2017. Likewise, convolutional approaches have shown that network-flow characteristics can provide sufficiently informative patterns for effective attack discrimination, particularly for traffic classes exhibiting recognizable behavioral structures [15]. Nevertheless, direct numerical comparisons across studies should be interpreted cautiously because differences in class definitions, preprocessing, sampling, train-test partitioning,

and evaluation metrics can substantially influence reported performance.

In contrast, BENIGN and DoS traffic remained considerably more difficult to classify, producing F1 scores of 63.50% and 67.88%, respectively. Their lower performance indicates greater overlap between these categories and other learned traffic representations. This result is important because a network intrusion detection system must not only identify highly distinctive attacks but also reliably distinguish malicious activity from legitimate traffic. Misclassification involving benign traffic can translate into false alarms or missed attacks in operational environments. Previous IDS research has consistently identified false-positive control and reliable discrimination between normal and malicious traffic as persistent challenges in anomaly-based intrusion detection. The present results therefore demonstrate why overall accuracy alone is insufficient for evaluating multiclass IDS performance. A model can perform strongly for several attack classes while remaining substantially weaker for other operationally important categories.

The use of macro-averaged metrics was particularly relevant in the present study because the seven experimental classes were not equally represented. Whereas the major classes contributed 5,000 observations each to the experimental dataset, Web Attack and Bot contained 2,180 and 1,966 observations, respectively. Class-weighted categorical cross-entropy was consequently employed during optimization to reduce the influence of unequal class frequencies. Dataset imbalance is a well-established challenge in intrusion detection because models trained on strongly unequal distributions can become biased toward frequently represented classes, while performance for rare attacks may remain unstable [16,]. The macro-F1 score of 82.74% is therefore particularly informative because it gives equal importance to each evaluated class rather than allowing the largest classes to dominate the summary measure.

The confusion-matrix analysis further supports the importance of class-sensitive evaluation. Strong diagonal classification was observed for Brute Force, PortScan, DDoS, and Bot, whereas

BENIGN and DoS exhibited greater cross-class confusion. Together with the class-wise precision, recall, and F1 scores, these patterns indicate that the hybrid architecture did not provide uniform discrimination across all attack families. Such heterogeneity is expected in multiclass network intrusion detection because different attack categories vary considerably in their traffic signatures and their degree of similarity to legitimate activity. Network intrusion datasets also differ in traffic-generation environments, attack composition, labeling procedures, and class distributions, all of which can materially influence model evaluation and comparability. Accordingly, reporting confusion matrices and class-level measures alongside aggregate metrics provides a more transparent assessment of the practical strengths and weaknesses of an IDS.

The baseline comparison provides empirical evidence for the contribution of the hybrid configuration within the present experimental setting. The CNN-only model achieved 52.98% accuracy and a weighted F1 score of 48.65%, while the LSTM-only model achieved 56.59% accuracy and a weighted F1 score of 52.59%. In comparison, the CNN-LSTM model achieved 82.20% accuracy and a weighted F1 score of 81.63%. The magnitude of this improvement suggests that combining the two representation-learning mechanisms was beneficial for the selected dataset and preprocessing configuration. Nevertheless, this finding should not be interpreted as evidence that CNN-LSTM architectures are universally superior to alternative intrusion-detection approaches. Machine-learning and deep-learning IDS performance is strongly dependent on the benchmark dataset, feature representation, class structure, optimization strategy, and evaluation protocol. The present comparison therefore establishes superiority only relative to the evaluated standalone CNN and LSTM baselines under the specified experimental conditions.

From an operational perspective, the findings demonstrate both the potential and the remaining challenges of deep-learning-based network intrusion detection. The high F1 scores obtained for PortScan, Brute Force, Web Attack, and Bot

indicate that the model successfully learned discriminative representations for several heterogeneous attack families. However, the weaker discrimination of BENIGN and DoS traffic indicates that further architectural or data-level refinement is necessary before the proposed framework could be considered suitable for operational deployment. Previous work on practical deep-learning-based intrusion detection has emphasized that predictive performance should ultimately be considered together with computational cost, processing latency, scalability, and the ability to operate on realistic network traffic [17]. Future evaluations of the proposed architecture should therefore incorporate inference latency, computational complexity, additional contemporary datasets, and externally collected network traffic.

Limitations

Several limitations should be considered when interpreting the findings. First, the experiment employed a stratified subset of 29,146 observations rather than the complete CICIDS2017 archive of approximately 2.83 million network-flow records. Although this approach produced a computationally manageable experimental dataset and reduced extreme dominance by the largest classes, it necessarily excluded a substantial proportion of the available traffic and may have altered the empirical class distributions present in the original dataset. Consequently, the reported performance should be interpreted as applying to the defined experimental subset rather than to the complete CICIDS2017 distribution.

Second, Infiltration and Heartbleed were excluded because the available dataset contained only 36 and 11 observations for these categories, respectively. Their inclusion would have produced highly unstable class-specific estimates under the adopted partitioning strategy. Nevertheless, their exclusion means that the proposed model was evaluated as a seven-class classifier and does not establish its ability to detect these rare attack categories. This limitation is particularly relevant because reliable recognition of low-frequency

attacks remains an important challenge for practical intrusion-detection systems.

Third, the LSTM component processed a transformed representation derived from the 78 flow-level features rather than chronological packet or flow sequences. The recurrent layer should therefore not be interpreted as learning explicit temporal attack evolution. Future research could construct chronological traffic sequences or session-based windows to determine whether explicit temporal information improves discrimination, particularly for attack categories exhibiting sequential behavior.

Fourth, the study relied exclusively on CICIDS2017. Although the dataset is widely used for IDS benchmarking, intrusion-detection datasets vary substantially in their traffic environments, attack scenarios, feature distributions, and labeling characteristics [7]. Validation on additional datasets and independently collected contemporary network traffic is therefore required before the generalizability of the model can be established. Finally, the baseline evaluation was restricted to standalone CNN and LSTM architectures. Although these baselines enabled direct assessment of the contribution of architectural integration, they do not establish superiority over other machine-learning or deep-learning classifiers. Future experiments should incorporate stronger comparative models, such as multilayer perceptron's, random forests, gradient-boosting methods, GRU/BiLSTM architectures, and other contemporary deep-learning approaches. Repeated experiments across multiple random seeds, confidence intervals for principal evaluation metrics, ablation analysis, and computational-efficiency measurements would further strengthen the statistical and practical validity of the proposed framework.

Conclusion

This study presented a hybrid CNN-LSTM model for multiclass network intrusion detection using CICIDS2017 flow-level traffic. The model achieved 82.20% test accuracy, 82.25% macro-precision, 83.90% macro-recall, and 82.74% macro-F1. PortScan and Brute Force achieved the

strongest class-level F1-scores, while BENIGN and DoS remained more difficult categories. Comparison with CNN-only and LSTM-only baselines demonstrated stronger performance from the hybrid architecture under the experimental configuration. Future evaluation should include the complete dataset, additional contemporary intrusion-detection benchmarks, temporal session representations, and independently collected traffic.

REFERENCES

- Sharafaldin I, Lashkari AH, Ghorbani AA. 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)*. 2018. p. 108–116. doi:10.5220/0006639801080116. (CiNii) DOI: 10.5220/0006639801080116
- Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A. A survey of network-based intrusion detection data sets. *Comput Secur.* 2019;86:147–167. doi:10.1016/j.cose.2019.06.005. (DBLP) DOI: 10.1016/j.cose.2019.06.005
- Khraisat A, Gondal I, Vamplew P, Kamruzzaman J. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity.* 2019;2:20. doi:10.1186/s42400-019-0038-7. (Deakin Research Online) DOI: 10.1186/s42400-019-0038-7
- Liu H, Lang B. Machine learning and deep learning methods for intrusion detection systems: a survey. *Appl Sci.* 2019;9(20):4396. doi:10.3390/app9204396. (MDPI) DOI: 10.3390/app9204396

- Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. *IEEE Access*. 2019;7:41525–41550. doi:10.1109/ACCESS.2019.2895334. (DOI) DOI: 10.1109/ACCESS.2019.2895334
- Al-Garadi MA, Mohamed A, Al-Ali AK, Du X, Ali I, Guizani M. A survey of machine and deep learning methods for Internet of Things (IoT) security. *IEEE Commun Surv Tutor*. 2020;22(3):1646–1685. doi:10.1109/COMST.2020.2988293. (ScienceOpen) DOI: 10.1109/COMST.2020.2988293
- Ferrag MA, Maglaras L, Moschoyiannis S, Janicke H. Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study. *J Inf Secur Appl*. 2020;50:102419. doi:10.1016/j.jisa.2019.102419. (ScienceDirect) DOI: 10.1016/j.jisa.2019.102419
- Doriguzzi-Corin R, Millar S, Scott-Hayward S, Martínez-del-Rincón J, Siracusa D. LUCID: a practical, lightweight deep learning solution for DDoS attack detection. *IEEE Trans Netw Serv Manag*. 2020;17(2):876–889. doi:10.1109/TNSM.2020.2971776. (Queen's University Belfast) DOI: 10.1109/TNSM.2020.2971776
- Thakkar A, Lohiya R. A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges. *Arch Comput Methods Eng*. 2021;28:3211–3243. doi:10.1007/s11831-020-09496-0. (CiNii) DOI: 10.1007/s11831-020-09496-0
- Maseer ZK, Yusof R, Bahaman N, Mostafa SA, Foozy CFM. Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. *IEEE Access*. 2021;9:22351–22370. doi:10.1109/ACCESS.2021.3056614. (Directory of Open Access Journals) DOI: 10.1109/ACCESS.2021.3056614
- Liu H, Lang B. Machine learning and deep learning methods for intrusion detection systems: a survey. *Appl Sci*. 2019;9(20):4396. doi:10.3390/app9204396. https://doi.org/10.3390/app9204396
- Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. *IEEE Access*. 2019;7:41525–41550. doi:10.1109/ACCESS.2019.2895334. https://doi.org/10.1109/ACCESS.2019.2895334
- Ferrag MA, Maglaras L, Moschoyiannis S, Janicke H. Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study. *J Inf Secur Appl*. 2020;50:102419. doi:10.1016/j.jisa.2019.102419. https://doi.org/10.1016/j.jisa.2019.102419
- Thakkar A, Lohiya R. A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges. *Arch Comput Methods Eng*. 2021;28:3211–3243. doi:10.1007/s11831-020-09496-0. https://doi.org/10.1007/s11831-020-09496-0
- Doriguzzi-Corin R, Millar S, Scott-Hayward S, Martínez-del-Rincón J, Siracusa D. LUCID: a practical, lightweight deep learning solution for DDoS attack detection. *IEEE Trans Netw Serv Manag*. 2020;17(2):876–889. doi:10.1109/TNSM.2020.2971776. https://doi.org/10.1109/TNSM.2020.2971776

Khraisat A, Gondal I, Vamplew P, Kamruzzaman J. Survey of intrusion detection systems: techniques, datasets and challenges. **Cybersecurity**. 2019;2:20.
doi:10.1186/s42400-019-0038-7.
<https://doi.org/10.1186/s42400-019-0038-7>

Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A. A survey of network-based intrusion detection data sets. **Comput Secur**. 2019;86:147-167.
doi:10.1016/j.cose.2019.06.005.
<https://doi.org/10.1016/j.cose.2019.06.005>

