ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

Improved and Efficient DDoS Attack Detection Mechanism Using Machine Learning

B. Nagarjuna
Research Scholar
Department of Computer Science & Engineering
GIET University, Gunupur, Odisha, India

Dr. Satya Narayan Das
Professor
GIET University, Gunupur, Odisha, India

Abstract—Distributed denial-of-service (DDoS) traffic exhibits packet-rate bursts, flow asymmetry, and temporal variation that complicate attack identification. A parallel convolutional neural network–long short-term memory (CNN–LSTM) framework integrates spatial and temporal network-flow representations for binary and multiclass detection. CICDDoS2019 traffic undergoes numerical preprocessing, 19-feature selection, normalization, and class balancing, followed by fused CNN and recurrent learning. The experimental evaluation achieves 99.85% accuracy for multiclass classification and 99.92% for binary classification. A second evaluation on CICIoT2023 yields 99.12% and 99.24% accuracy, respectively. Stream-based classification interfaces visualize traffic characteristics, confusion matrices, and detection activity. The analysis emphasizes benchmark-specific evidence and the distinction between traffic-stream emulation and live network deployment.

Keywords—DDoS attack detection, parallel CNN–LSTM, feature fusion, CICDDoS2019, CICIoT2023, SMOTE, network traffic classification.

I. INTRODUCTION

A) Research Background

Distributed denial-of-service attacks compromise service availability by overwhelming a target with coordinated traffic from multiple network hosts. Volumetric, protocol-level, and application-layer attacks differ in their operational signatures, which makes a single fixed threshold insufficient for heterogeneous environments. Network intrusion detection therefore requires accurate classification while limiting processing overhead and false alarms. Learning-based methods examine flow duration, packet statistics, inter-arrival times, and protocol-related attributes rather than depending exclusively on static signatures [1]–[5].

B) Problem Definition and Motivation

Serial CNN–LSTM implementations process spatial and temporal representations through successive stages. Their architectural dependence may constrain the exploitation of complementary traffic characteristics. The parallel feature-fusion approach evaluates shared convolutional features through a dense branch and a recurrent branch before combining the representations. This arrangement seeks to retain compact features while differentiating benign connections from diverse DDoS classes. Evaluation requires both attack-level multiclass performance and the simpler benign-versus-attack binary task [6]–[9].

C) Research Contributions

The investigation integrates 19 selected flow features, class balancing, parallel CNN–LSTM processing, stratified cross-validation, and independent binary/multiclass assessment. It also examines classification on CICIoT2023 and operational visualization through a stream emulator.

Model quality is assessed using accuracy, precision, recall, F1-score, and confusion matrices. Cross-dataset evaluation is reported separately because distinct capture conditions and target distributions influence generalization [10]–[12].

II. RELATED WORK

A) Learning-Based Detection

Earlier intrusion detection systems employ support vector machines, random forests, multilayer perceptrons, and convolutional models to learn the statistical structure of network-flow features. Classical models can operate efficiently on structured attributes, although their decision boundaries may become less effective for correlated attack patterns and changing traffic distributions. CNNs learn localized relationships among input variables, whereas LSTMs provide gated memory for ordered and temporal dependencies [13]–[16].

B) Hybrid CNN–LSTM Architectures

Hybrid architectures combine convolutional feature extraction with recurrent information processing for detection tasks on software-defined networks, Internet of Things deployments, and cloud infrastructure. Existing studies investigate serial concatenation, bidirectional recurrence, attention, and more compact feature-fusion strategies [17]–[22]. The parallel architecture separates the 64-unit dense representation and the recurrent representation containing 64 and 32 LSTM units; these feature vectors are concatenated into a 96-dimensional embedding for final classification.

C) Research Gap

High benchmark accuracy does not independently establish performance on different network distributions. Class imbalance, redundant features, preprocessing order, and emulated traffic conditions can affect model rankings. Accordingly, this work evaluates the selected architecture under separate binary and multiclass tasks and includes a second network-flow benchmark. Results are interpreted alongside the distinct objectives of predictive classification, stream monitoring, and mitigation-oriented interfaces [23]–[26].

D) Evaluation Challenges in Network Security

Direct comparisons among intrusion detectors must account for the label taxonomy, the granularity of input

observations, and the evaluation partition. A model evaluated only on benign-versus-attack labels cannot be assumed to identify particular DDoS attack families with equal reliability. Moreover, a high classification rate on randomly mixed traffic may reflect repeated flow signatures, shared capture sessions, or feature leakage. Multiclass recall and cross-dataset sensitivity therefore complement the overall classification accuracy. These evaluation considerations motivate the separate experiments in the present framework [9], [14], [18].

The assessed literature also identifies a trade-off between network depth and deployment cost. Feature reduction can decrease the number of input variables, while parallel model branches can capture complementary relations without requiring a fully sequential deep hierarchy. Neither mechanism independently proves reduced inference latency; reproducible timing experiments must disclose hardware configuration, batch size, feature extraction overhead, and traffic source. Model complexity, classification capability, and stream-level visualization are evaluated as distinct attributes of the overall system.

III. MATERIALS AND METHODS

A) Dataset Preparation and Feature Selection

The development dataset is CICDDoS2019. Initial preprocessing addresses invalid values, normalization, label encoding, and recursive feature elimination (RFE). The resulting 19 attributes include flow duration, packet length, packet and byte rates, TCP header sizes, and forward/backward inter-arrival-time statistics. The pre-resampling feature matrix contains 291,696 rows with 19 selected attributes. The class-balancing procedure produces 2,243,520 rows in the experimental configuration.

The resampled dataset is partitioned into training (70%), validation (15%), and testing (15%) sets, corresponding to 1,570,464, 336,528, and 336,528 instances. Synthetic Minority Oversampling Technique (SMOTE) is applied to address unequal attack-class frequencies. As a methodological consideration, preprocessing and resampling should be isolated to training folds in further independent validation to prevent information transfer between training and evaluation samples.

B) Feature Semantics and Class Preparation

Traffic flows are represented through 19 features selected by RFE from packet, protocol, and timing measurements. Flow Duration reflects connection persistence; Flow Bytes/s and Flow Packets/s quantify throughput; Total Fwd Packets and Bwd/Fwd Header Length describe directional volume and structure. Packet Length Mean, Standard Deviation, Minimum, and Maximum characterize frame-size behavior. Forward and backward inter-arrival features, including mean and dispersion, capture traffic burstiness and possible coordination among attacking hosts. Active Mean and Idle Mean provide additional duration statistics for separating regular communication from sustained request floods.

The multiclass objective represents benign traffic and differentiated attack categories, whereas the binary task groups attacks into a single malicious label. Preprocessing assigns consistent numerical class codes and maps them back to the corresponding semantic categories for the evaluation outputs. Label distribution is reviewed before resampling to reduce bias in rare classes. Class balancing increases the sample count substantially; accordingly, the evaluated resampled matrix must be distinguished from the number of unique original flow observations.

C) Parallel CNN–LSTM System Architecture

A shared one-dimensional convolutional stage contains 64- and 128-filter Conv1D layers with batch normalization and max pooling. Pooled feature maps are processed through two concurrent pathways: a flattened representation followed by a 64-unit dense layer, and a sequence-processing pathway comprising 64-unit and 32-unit LSTMs. Their outputs are concatenated into a 96-dimensional representation. A final classification layer provides either binary traffic status or a 12-class attack prediction. Figure 1 illustrates the feature-fusion and evaluation pipeline.

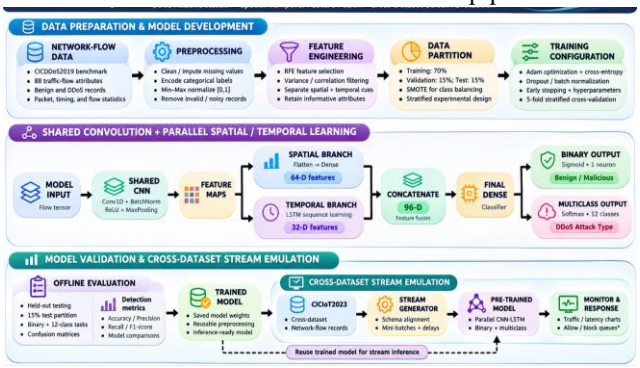


Fig. 1. System Architecture

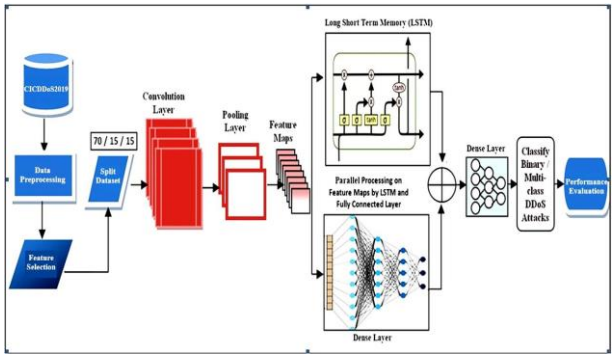


Fig. 2. Detailed parallel feature-processing architecture.

The parallel CNN–LSTM architecture illustrates the independent feature-processing branches: shared convolutional and pooling operations produce feature maps for both the LSTM pathway and the fully connected pathway. Their outputs are fused prior to the decision layer. This differs from a serial design in which recurrent processing necessarily follows a single dense representation. The fused 96-dimensional vector contains 64 dense components and 32 recurrent components, providing the input for the final attack classifier.

D) Training Configuration and Validation

Training uses the Adam optimizer with learning rate 0.001, mini-batches of 32, and dropout regularization of 0.5 within the dense pathway. Stratified five-fold cross-validation is followed by validation and held-out testing. The experimental description identifies approximately 50 training epochs and a reported GPU training duration of 30–35 minutes. Evaluation across independently trained binary and multiclass configurations avoids interpreting a multiclass prediction as an automatically verified binary decision.

Validation operates at several levels. During stratified five-fold training, each iteration estimates classification quality on a held-out fold of development data. A subsequent static validation partition is used to assess the selected

configuration, followed by an untouched test partition for final classification estimates. The binary and multiclass models are evaluated independently to avoid conflating their objectives. Training is conducted in a GPU-enabled environment, while predictive accuracy is evaluated separately from resource consumption and attack-prevention behavior.

Model dimensionality also influences the inference path. The Conv1D transformation produces 128-channel feature maps, and temporal pooling reduces their sequence length to nine. The dense branch produces a 64-unit representation; the recurrent pathway applies 64 units before a 32-unit output. Concatenation is followed by a dataset-appropriate output layer. This structure supplies the compact fused representation reported in the architectural configuration and enables both binary and multiclass learning under a common feature-engineering pipeline.

E) Performance Evaluation Equations

For a positive attack class, true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) define the classification criteria. Multiclass metrics require explicit aggregation across classes; the binary definitions below provide the constituent measures.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$\text{F1} - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

F) Reproducibility and Experimental Controls

The dataset split, random states, feature indices, normalization parameters, and label mappings are essential components of reproducible classifier evaluation. The same order of 19 features must be maintained when preparing an inference record; otherwise, a scaled vector can be misinterpreted despite satisfying input dimensionality checks. For cross-dataset testing, corresponding feature definitions and units should be aligned before the prediction stage. Further measurements under time-separated captures would establish how the detector responds when traffic distributions change over sustained periods.

The system architecture separates computational classification from attack prevention. Classification predicts the traffic label, whereas the DDoS Deflate utility accepts an address-management decision and supplies monitoring controls. The detection dashboard and prevention utility should therefore be assessed independently when testing false-positive blocking, operator confirmation, and throughput overhead. This separation prevents a classifier score from being interpreted as an independently measured prevention rate.

IV. EXPERIMENTAL RESULTS

A) Classification Performance on CICDDoS2019

The evaluation reports closely aligned cross-validation, static-validation, and held-out test outcomes. Multiclass detection achieves 99.85% accuracy with 99.51% precision, 99.11% recall, and 99.30% F1-score. Binary detection achieves 99.92% accuracy, 99.70% precision, 99.47% recall, and 99.54% F1-score. The slightly higher binary scores are consistent with the reduced difficulty of separating two label groups rather than individual attack classes.

TABLE I. CICDDoS2019 CLASSIFICATION PERFORMANCE

Task	Accuracy	Precision	Recall	F1-score
Multiclass	99.85	99.51	99.11	99.30
Binary	99.92	99.70	99.47	99.54

Five-fold cross-validation means match the test-set values stated above. The multiclass confusion matrix exhibits pronounced diagonal classification, while the binary confusion matrix separates attack and benign instances with comparatively few off-diagonal entries. These observations support strong within-benchmark discrimination, although the aggregate classification metrics do not establish real-time detection latency under an independent live traffic feed.

TABLE II. FIVE-FOLD CROSS-VALIDATION SUMMARY

Task	Accuracy	Precision	Recall	F1-score
Multiclass mean	99.85	99.51	99.11	99.30
Binary mean	99.92	99.70	99.47	99.54

Across five folds, multiclass accuracy ranges from 99.84% to 99.86%, while the binary classification values range from 99.91% to 99.93%. The narrow ranges indicate limited variability among the reported development partitions. Precision and F1-score similarly remain near their mean values. Static validation and final testing yield the same aggregate metrics in the experimental evaluation; identical summary statistics do not by themselves establish complete independence of preprocessing operations.

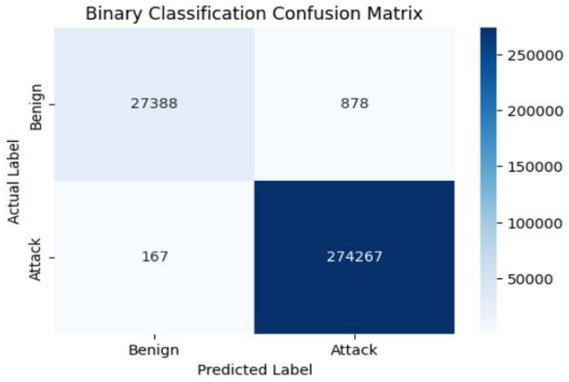


Fig. 3. Binary-class confusion matrix for CICDDoS2019.

B) Cross-Dataset Evaluation on CICIoT2023

Additional evaluation applies the detection framework to CICIoT2023, which contains a different selection of traffic behaviors and attack configurations. Multiclass scores are 99.12% accuracy, 99.14% precision, 99.13% recall, and 99.13% F1-score. Corresponding binary results are 99.24%, 99.45%, 99.30%, and 99.37%. The reduction relative to CICDDoS2019 indicates that classification remains effective in the reported cross-dataset experiment but is sensitive to changes in feature and class distributions.

TABLE III. CICIoT2023 CLASSIFICATION PERFORMANCE

Task	Accuracy	Precision	Recall	F1-score
Multiclass	99.12	99.14	99.13	99.13
Binary	99.24	99.45	99.30	99.37

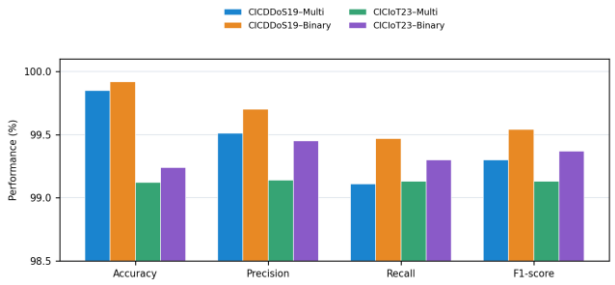


Fig. 5. Combined comparison of Accuracy, Precision, Recall, and F1-score (%).

The relative drop in multiclass accuracy between the two evaluations is 0.73 percentage points, while the binary accuracy decreases by 0.68 percentage points. The corresponding F1-score decreases are 0.17 and 0.17 percentage points. These differences are small in absolute terms but important for assessing distributional sensitivity, especially where class frequencies and collection environments differ. Cross-dataset scores also depend on the consistency of label alignment and whether the input features represent equivalent measurement definitions.

A network-monitoring system must detect malicious behavior without unnecessarily blocking legitimate traffic. Therefore, binary recall captures missed attacks, while binary precision captures false alarms among the predicted malicious flows. Multiclass error analysis additionally identifies confusion between attack families with related traffic signatures. For example, rate-based floods may share characteristic burst intervals despite differing in transport protocol; such cases should be examined through class-level confusion and error attribution.

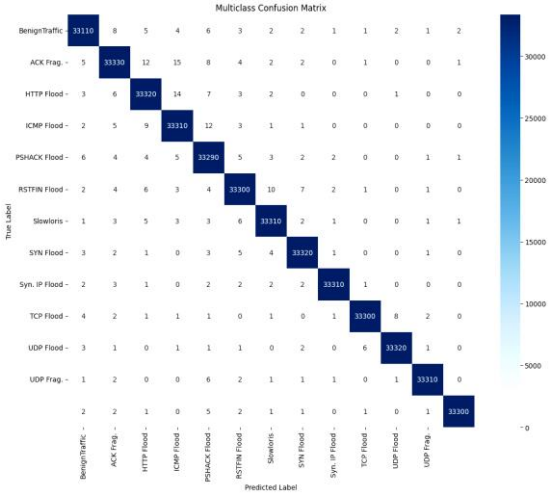


Fig. 4. CICIoT2023 multiclass confusion-matrix evaluation.

The CICIoT2023 multiclass confusion matrix provides a class-level view of the evaluation beyond aggregate accuracy. Most observations appear along the main diagonal, while some off-diagonal entries indicate confusion among traffic patterns with related protocol and rate characteristics. This distinction matters in intrusion detection because classification of a malicious connection as the wrong attack family has a different operational consequence from classifying the same connection as benign. Thus, both the binary and multiclass evaluations are required for a complete interpretation of detector behavior.

The cross-dataset comparison is also useful for diagnosing potential dependence on the original traffic collection. Changes in network topology, device mix, background traffic volume, and flow-generation tools can alter the distribution of the 19 selected features. The reported results indicate strong discrimination for the tested CICIoT2023 records, but repeated evaluation with time-separated captures and a documented feature-mapping protocol would further strengthen evidence for deployment across independent networks.

C) Comparison with Alternative Architectures

Comparative analysis of the parallel CNN–LSTM framework against selected serial and parallel CNN–LSTM studies reveals an improvement in the reported classification measures under the respective benchmark settings. A compact fused representation and selected input subset reduce the feature dimension relative to broader models. Because published studies use different traffic data, partitioning schemes, and training budgets, their metric differences are contextual comparisons rather than controlled proof of superiority under identical conditions [27]–[30].

TABLE IV. MODEL AND EVALUATION CONFIGURATION

Parameter	Configuration
Selected input features	19
CNN convolution filters	64, 128
Parallel fusion width	96
Training epochs	50
Train / val / test	70% / 15% / 15%

TABLE V. SELECTED SERIAL CNN–LSTM COMPARISONS

Approach	Acc.	Prec.	Rec.	F1
Parallel CNN–LSTM (binary)	99.92	99.70	99.47	99.54
Salmi & Oughdir (2022)	94.40	95.90	92.20	94.05
Issa & Albayrak (2023)	99.20	91.94	93.37	92.41
Nassar et al. (2023)	98.96	98.00	98.00	98.00

The comparative evaluation incorporates performance metrics from established intrusion-detection approaches, each assessed under its respective experimental protocol. The tabulated comparisons indicate the range of performance reported for serial CNN–LSTM and related detection models. Differences in labels, capture environments, preprocessing, and class balance can affect every metric. Within this context, the proposed model combines a compact input set and high measured classification scores, making it suitable for more controlled latency and drift evaluations.

D) Stream Emulation and Visualization

The stream emulator processes batches of traffic observations and displays source/destination information, protocol, transmission rate, and predicted label. A network-traffic dashboard visualizes the active stream, classification indicators, traffic-rate variation, and historical processing information. Figure 6 illustrates the network traffic monitoring dashboard and classification workflow evaluated through stream emulation rather than direct production-network traffic.

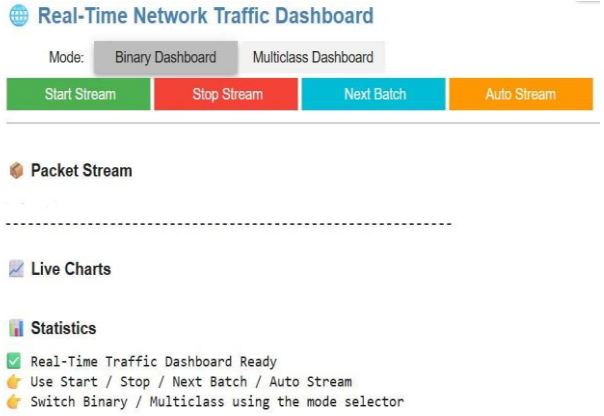


Fig. 6. Network traffic stream-emulation dashboard.

The multiclass stream evaluation provides attack-category distributions and confusion-matrix analysis. The binary stream view presents benign-versus-attack decisions and monitored packet rates. Distinguishing emulator-generated traffic characteristics from real packet-capture measurements is important when interpreting response-time and throughput claims. The implementation also presents an intrusion-prevention interface using DDoS Deflate control and status logs.



Fig. 7. DDoS Deflate attack-prevention monitoring interface.

A further interface presents packet or flow-level identifiers, packet rate, and the associated predicted class. These fields provide contextual interpretation for an analyst who must distinguish traffic surges from legitimate load variation. Figures that visualize stream activity are linked to the emulation process; the tested event stream is not equivalent to a continuously captured, independently instrumented operational network. A larger-scale deployment would require production traffic ingestion, calibration and monitoring of shifts in the statistical properties of each feature.



Fig. 8. Attack-prevention event and status records.

E) Interpretation and Limitations

The strong benchmark scores indicate that the parallel feature pathways capture complementary flow attributes in the evaluated datasets. However, the effects of oversampling order, dataset overlap, traffic-drift severity, class-mapping consistency, and calibration should be examined through leakage-controlled and time-separated experiments. The reported streaming demonstration supports interface and classification workflow evaluation; operational mitigation efficacy requires independent measurements of blocking accuracy, latency, and legitimate-traffic impact.

F) Generalization, Mitigation, and Operational Readiness

The cross-dataset evaluation demonstrates sustained classification quality in the reported CICIoT2023 configuration. The benchmark analysis nevertheless leaves practical questions about operational robustness. Captured production traffic can include novel protocols, encrypted applications, changes in flow exporter configuration, and class distributions that are not represented by the experimental partitions. Reliable adoption would require periodic evaluation against new capture windows, uncertainty-aware decision thresholds, and explicit monitoring of false alarm rates on benign workloads.

A prevention layer introduces additional requirements beyond classification. Address blocking must account for shared client addresses, changing network endpoints, and traffic routed through gateways or network address translation. Security controls should preserve auditability and permit reversible mitigation decisions. Accordingly, the traffic-classification model, monitoring display, and DDoS Deflate interface constitute connected but analytically distinguishable stages. The measured classifier scores provide evidence for threat discrimination; they do not independently quantify the effectiveness of blocking policies or availability improvement during an attack.

G) Error Analysis and Classification Thresholds

An intrusion detector must balance missed attacks against alarms generated for benign traffic. The aggregate binary precision of 99.70% and recall of 99.47% on CICDDoS2019 describe this trade-off at the selected decision threshold. The corresponding values on CICIoT2023 are 99.45% and 99.30%, indicating a modest change in the false-alarm and missed-detection balance across the evaluated datasets. The multiclass confusion matrices provide complementary information by identifying which attack categories remain separable and which show overlapping features. For deployment assessment, thresholds should be examined using receiver-operating and precision–recall analyses across plausible traffic priors, rather than relying only on the fixed classification decision used for the reported tables.

V. CONCLUSION

The parallel CNN–LSTM feature-fusion framework performs binary and multiclass DDoS classification using a compact 19-feature network-flow representation. On CICDDoS2019, the reported final accuracies are 99.92% for binary classification and 99.85% for multiclass identification; cross-dataset evaluation on CICIoT2023 yields 99.24% and 99.12%, respectively. The model integrates convolutional spatial extraction, recurrent sequence processing, and a 96-dimensional fused representation. A traffic-stream emulator extends the assessment to dashboard visualization and mitigation-oriented monitoring. Future evaluation should prioritize leakage-controlled preprocessing, temporal validation, calibrated predictions, and measured end-to-end operational performance.

REFERENCES

[1] Abid, Y. A., Wu, J., Xu, G., Fu, S., & Waqas, M., “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, 2024.

[2] Acharya, T., Annamalai, A., & Chouikha, M. F., "Efficacy of CNN-Bidirectional LSTM Hybrid Model for Network-Based Anomaly Detection," 13th IEEE Symposium on Computer Applications and Industrial Electronics, ISCAIE 2023, 348–353. https://doi.org/10.1109/ISCAIE57739.2023.10165088, 2023.

[3] Ahmed, S., Khan, Z. A., Mohsin, S. M., Latif, S., Aslam, S., Mujlid, H., Adil, M., & Najam, Z., "Effective and Efficient DDoS Attack Detection Using Deep Learning Algorithm, Multi-Layer Perceptron," Future Internet, 15(2), 1–24. https://doi.org/10.3390/fi15020076, 2023.

[4] Ahmim, A., Maazouzi, F., Ahmim, M., Namane, S., & Dhaou, I. Ben., "Distributed Denial of Service Attack Detection for the Internet of Things Using Hybrid Deep Learning Model," IEEE Access, 11 (November), 119862–119875. https://doi.org/10.1109/ACCESS.2023.3327620, 2023.

[5] Akgun, D., Hizal, S., & Cavusoglu, U., "A new DDoS attacks intrusion detection model based on deep learning for cybersecurity," Computers and Security, 118, 102748. https://doi.org/10.1016/j.cose.2022.102748, 2022.

[6] Al-hasani, A. O. K., Abdelmaksoud, I. R., & Rezk, A., "A Novel Hybrid CNN-LSTM Framework for Robust DDoS Attack Detection and Classification," Journal of Cybersecurity and Information Management, 17(1), 21–34. https://doi.org/10.54216/jcim.170103, 2026.

[7] Al-Zubidi, A. F., Farhan, A. K., & Towfek, S. M., "Predicting DoS and DDoS attacks in network security scenarios using a hybrid deep learning model," Journal of Intelligent Systems, 33(1). https://doi.org/10.1515/jisys-2023-0195, 2024.

[8] Alasmari, T., Eshmawi, A., Alshomrani, A., & Hsairi, L., "CNN-LSTM Based Approach for DDoS Detection," Proceedings of the 2023 8th International Conference on Mobile and Secure Services, MobiSecServ 2023, CFP23RAC-A, 1–6. https://doi.org/10.1109/MobiSecServ58080.2023.10329028, 2023.

[9] Aldhyani, T. H. H., & Alkahtani, H., "Cyber Security for Detecting Distributed Denial of Service Attacks in Agriculture 4.0: Deep Learning Model," Mathematics, 11(1), 1–19. https://doi.org/10.3390/math11010233, 2023.

[10] AlHaddad, U., Basuhail, A., Khemakhem, M., Eassa, F. E., & Jambi, K., "Ensemble Model Based on Hybrid Deep Learning for Intrusion Detection in Smart Grid Networks," Sensors, 23(17). https://doi.org/10.3390/s23177464, 2023.

[11] Alqahtani, H. M., & Abdullah, M., "A Review on DDoS Attacks Classifying and Detection by ML/DL Models," International Journal of Advanced Computer Science and Applications, 15(2), 824–833. https://doi.org/10.14569/IJACSA.2024.0150283, 2024.

[12] Alrumaih, T. N. I., & Alenazi, M. J. F., "ERINDA: A novel framework for Enhancing the Resilience of Industrial Networks against DDoS Attacks with adaptive recovery," Alexandria Engineering Journal, 121(March), 248–262. https://doi.org/10.1016/j.aej.2025.02.042, 2025.

[13] Alshattnawi, S., & Alshboul, H. R., "Combined Deep Learning Approaches for Intrusion Detection Systems," International Journal of Interactive Mobile Technologies, 17(15), 135–154. https://doi.org/10.3991/ijim.v18i19.49907, 2024.

[14] Aslam, N., Srivastava, S., & Gore, M. M., "ONOS Flood Defender: An Intelligent Approach to Mitigate DDoS Attack in SDN," Transactions on Emerging Telecommunications Technologies, 33(9), 0–35. https://doi.org/10.1002/ett.4534, 2022.

[15] Aslam, N., Srivastava, S., & Gore, M. M., "A Comprehensive Analysis of Machine Learning- and Deep Learning-Based Solutions for DDoS Attack Detection in SDN," Arabian Journal for Science and Engineering, 49(3), 3533–3573. https://doi.org/10.1007/s13369-023-08075-2, 2024.

[16] Aswad, F. M., Ahmed, A. M. S., Alhammadi, N. A. M., Khalaf, B. A., & Mostafa, S. A., "Deep learning in distributed denial-of-service attacks detection method for Internet of Things networks," Journal of Intelligent Systems, 32(1). https://doi.org/10.1515/jisys-2022-0155, 2023.

[17] Bao, X., Chen, E., & Tu, X., "Hybrid Deep Neural Network-Based Detection of DDoS Attacks in Software-Defined IIoT Networks," Preprints, 0–13. https://doi.org/10.20944/preprints202501.0925.v1, 2025.

[18] Belhaj Mohamed, M., Bouzidi, D., Khalid Ibraheem, M., Al-Abadi, A. A. J., & Fakhfakh, A., "Cyber Approach for DDoS Attack Detection Using Hybrid CNN-LSTM Model in IoT-Based Healthcare," Future Internet, 18(1), 52. https://doi.org/10.3390/fi18010052, 2026.

[19] Danang, D., & Mustofa, Z., "CLSTMNet Architecture: A CNN–LSTM-Based hybrid deep learning model for DDOS attack detection and mitigation in network security," Journal of Artificial Intelligence and Technology. https://doi.org/10.37965/jait.2025.0887, 2026.

[20] Deepthy, J., Radhika, N., & Denny, A., "Regularized Hybrid Deep Learning for DDoS Attack Prediction in Software Defined Internet of Things (SD-IoT)," 2024 3rd International Conference on Power, Control and Computing Technologies, ICPC2T 2024, 83–87. https://doi.org/10.1109/ICPC2T60072.2024.10474620, 2024.

[21] Eguavoen, V. O., Olanrewaju, B. S., & Okafor, C. N., "A Hybrid CNN-LSTM and Adaboost Model for Classifying Intrusion In IoT Networks," Fudma Journal Of Sciences, 9(5), 204-212. https://doi.org/10.33003/fjs-2025-0905-3495, 2025.

[22] Elubeyd, H., & Yiltas-Kaplan, D., "Hybrid Deep Learning Approach for Automatic DoS/DDoS Attacks Detection in Software-Defined Networks," Applied Sciences (Switzerland), 13(6). https://doi.org/10.3390/app13063828, 2023.

[23] Ghorsad, T. N., & Zade, A. V., "Hybrid CNN+LSTM Deep Learning Model for Intrusions Detection Over IoT Environment," International Journal on Recent and Innovation Trends in Computing and Communication, 11(10s), 01–11. https://doi.org/10.17762/ijritcc.v11i10s.7588, 2023.

[24] Gueriani, A., Kheddar, H., & Mazari, A. C., "Enhancing IoT Security with CNN and LSTM-Based Intrusion Detection Systems," PAIS 2024 - Proceedings: 6th International Conference on Pattern Analysis and Intelligent Systems. https://doi.org/10.1109/PAIS62114.2024.10541178, 2024.

[25] Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., & Ahmad, R., "CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System," IEEE Access, 10 (September), 99837–99849. https://doi.org/10.1109/ACCESS.2022.3206425, 2022.

[26] Issa, A. S. A., & Albayrak, Z., "DDoS Attack Intrusion Detection System Based on Hybridization of CNN and LSTM," Acta Polytechnica Hungarica, 20(2), 105–123. https://doi.org/10.12700/APH.20.2.2023.2.6, 2023.

[27] Jayakrishna, N., & Prasanth, N. N., "A Hybrid Deep Learning Model For Detection And Mitigation of DDoS Attacks in VANETs," Scientific Reports (Vol. 15, Issue 1). https://doi.org/10.1038/s41598-025-15215-1, 2025.

[28] Jayalaxmi, P. L. S., Saha, R., Kumar, G., Conti, M., & Kim, T. H., "Machine and Deep Learning Solutions for Intrusion Detection and Prevention in IoTs: A Survey," IEEE Access, 10, 121173–121192. https://doi.org/10.1109/ACCESS.2022.3220622, 2022.

[29] Ji, C., Liu, H., & Dai, W., "Hybrid Model for Network Traffic Anomaly Detection Based on Parallel Two-Stage Feature Fusion," IEEE Access, 13(January), 27310– 27324. https://doi.org/10.1109/ACCESS.2025.3538319, 2025.

[30] Jiang, R., Weng, Z., Shi, L., Weng, E., Li, H., Wang, W., Zhu, T., & Li, W., "Intelligent Botnet Detection in Iot Networks Using Parallel CNN-LSTM Fusion," Concurrency and

Computation: Practice and Experience, 36(24), 1–16. https://doi.org/10.1002/cpe.8258, 2024.