Contents lists available at IdNNS
Indonesian Journal of Cybersecurity and Emerging Risks
Volume 1 ; Issue 1
journal homepage: https://journal.idnns.org/index.php/ijcyber

Explainable Deep Learning for Network Attack Detection and Classification: A SHAP-Based Interpretation Framework

Anupama Mishra[a], Pramod Kumar[b,*] and Vinay Rishiwal[c]

[a]Computer Science & Engineering, School of Science & Technology, Swami Rama Himalayan University, Dehradun, Uttarakhand, 248016, India
[b]School of Science & Technology, Swami Rama Himalayan University, Dehradun, Uttarakhand, 248016, India
[c]Dept. of CSIT, M.J.P. RohilKhand University, Bareilly, India

ARTICLE INFO

Keywords:
CNN
LSTM
SHAP
Distributed Denial of Service
DDoS
Network Security
Machine learning
Deep Learning
Web Security

ABSTRACT

There are many attacks on network such as DNS, LDAP, MSSQL, NTP, NetBIOS, PortMap, SNMP, and DDoS. Among all Distributed Denial-of-Service (DDoS) attacks are major security threats in networks. These attacks affect the availability and performance of network services. Traditional intrusion detection methods often face difficulties in identifying different types of attacks accurately. In our research paper, a framework based on an explainable deep learning is proposed to detect and predict multi-classification DDoS attacks. Here, we used the CIC-DDoS2019 dataset for experimentation. Initially, data preprocessing and feature selection are carried out using the Random Forest technique to identify important network traffic features. The selected features are then provided to a hybrid model mix of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) model for classification. The results we achieve as accuracy of 75.39% and a weighted F1-score of 72%. SHAP analysis is applied to explain the prediction results. The tool is used to enhance the interpretability of the model. The analysis shows that Flow Bytes/s, Flow IAT Std, and Fwd Packets Length Total are the most important features for attack detection. The proposed approach provides effective classification along with better understanding of model decisions.

1. Introduction

The speedy development of communication networks along with technologies such as cloud computing, Internet of Things (IoT) devices, and networking infrastructures has been increased enormously as well as the complexity. Though these technologies are base of improved connectivity and play a vital role in operational efficiency. However, they also exposed to a number of cyber threats due to its complex structure. Among all these threats, Distributed Denial-of-Service (DDoS) attacks Mishra et al. (2024), network intrusion activities, and attacks based on signature, anomaly, and protocol remain serious concerns. These attacks are capable enough to disrupt server services, degrade network performance, and compromise critical resources.

Earlier to identify suspicious activities, intrusion detection systems (IDSs) Abdulganiyu et al. (2023) were used that mostly based on signature-based and rule-based mechanisms. The performance was high majorly on known attack pattern, however difficult to detect evolving behaviors of attacks.

To address these evolving and dynamic limitations, there is a need of AI techniques based on deep learning models. Deep learning includes CNNs Mohammadpour et al. (2022), LSTM Imrana et al. (2021) networks, and Transformer architectures. The ability to learn complex pattern and identify distinguishing malicious activities from legitimate network behavior is quite high. Intrusion detection systems based on deep learning are mostly act as a black box. It is hard to find and interpret its decision-making processes. This behavior adds on challenges for security analyst who needs transparent and trustworthy explanations before deploying automated security solutions in real-world environments.

One of the major limitations of artificial intelligence systems in network security is the lack of transparency and interpretability. Many deep learning models provide accurate predictions, but their decision-making process is often difficult to understand. When the process is not known for decision, and trust is the main concern, it becomes challenge in deploying such systems in critical security environments. Explainable Artificial Intelligence (XAI) has come out as an important research area that focuses on making model predictions more understandable, transparent and trustworthy. So, we use XAI tools to address trust, transparency and reliable issues with reasons. There are

*Corresponding author
anupamamishra@srhu.edu.in (A. Mishra); tulasacademic@gmail.com (P. Kumar); anupamamishra@srhu.edu.in (V. Rishiwal)
ORCID(s):

many tools based on XAI concepts such as SHAP (SHapley Additive exPlanations), Permutations Feature Importance, Integrated Gradients, LIME (Local Interpretable Model-Agnostic Explanations) and Attention Visualizations. Among all, in this research we use SHAP tool for explainable reasoning. The tool will integrate the reasoning portion into our proposed framework.

SHAP is an XAI tool and helps in measuring the contribution of each feature towards a particular prediction and provides valuable insights into model behavior. A reason-based explanations helps in understanding attack detection decisions and improving confidence in the system.

In this research work, an explainable deep learning framework is proposed for DDoS attack detection and classification. Our research contributions are as follows:

1. Framework development using deep learning for DDoS detection and classification.
2. Integration of SHAP analysis to make the model understandable and predictable.
3. Identification of features during network traffic features for attack detection.
4. Experimental evaluation using the CIC-DDoS2019 benchmark dataset.
5. Discussion of explainable deep learning that validate transparency, trust, and decision-making by using it results.

2. Literature Review

There are many researchers who have researched the XSS attacks by applying machine learning and deep learning that enhanced web application security.

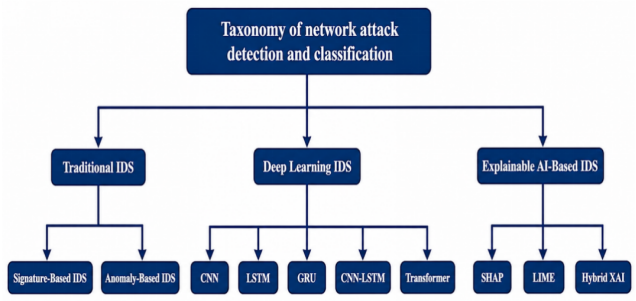

Figure 1: Taxonomy of Network Attack Detection and Classification

2.1. Traditional Intrusion Detection Systems

The authors proposed a hybrid approach based on ML and DL Ahmed et al. (2025), they applied the K-Nearest Neighbors, Decision Tree, Support vector Machine, Long Short-Term Memory, Random Forest, and Artificial Neural Network for their research work. The author also validated their work by their results.

In reference Varma et al. (2025); Nguyen and Harper (2026), the proposed system addresses a problem of high false positive rates, evasion techniques, and scalability challenges. Signature-based combined with anomaly-based detection was applied for known threats and novel behaviors.

Intrusion Detection Systems (IDS) Blessing et al. (2023), In this paper, the authors focused on two primary detection methodologies: signature-based combined with anomaly-based detection based on predefined patterns of known threats. The results was a good high accuracy with low false positive alarms. However, unable to perform against zero-day vulnerabilities and unknown threats.

2.2. Deep Learning Intrusion Detection Systems

The researchers discussed about Connected and Autonomous Vehicles (CAVs) He et al. (2026). Though the topic is important for enabling intelligent mobility and autonomous driving, however, its inherent connectivity exposed to critical vulnerabilities to cyber-attacks. The cyber-attacks breaches privacy and lead to system failures in transportation networks. Escalation of deployment of CAVs, for robust cybersecurity is challenge and traditional mechanisms are not enough. Decision Trees (DT) and Support Vector Machines (SVM), are majorly incorporated but suffer from inadequate feature extraction in high-dimensional network traffic data, significantly compromising the accuracy of cyber-attack identification.

So, deep learning has become a better way to find complicated patterns and simplify the process of feature extraction and selection. The author suggested LSTM-1DResNet, an intruder detection model based on deep learning. It has a classifier and an autoencoder. A Multilayer Perceptron (MLP) is used by the algorithm to accurately classify attacks. The model was tested on two datasets, the NSL-KDD dataset and the CICIDS2017 dataset. It got 94.38% accuracy, which was 11% better than standalone CNN and LSTM models. On the CICIDS2017, scores of 97.98% accuracy, 98.11% precision, and 97.98% memory, along with high F1-scores of 96.90%.

The writers Kaur et al. (2026) suggested a mixed deep learning model that is made up of one-dimensional Convolutional Neural Networks (1D-CNN) and Gated Recurrent Units (GRU). This model learns the spatial-temporal patterns needed to find intrusions. The CNN part is organised in a hierarchy, and the retrieved features can be learned locally. The GRU layer, on the other hand, learns about how network session sequences change over time and interact with each other over long distances. The hybrid model combines the best features of the convolutional and recurrent learning models to make them work better together.

The proposed SDN-FL-GRU model Alsarray (2026) lets distributed IoT nodes work together to train a global intrusion detection model without sharing raw data. It does this by using the FedAvg algorithm to combine the data in an efficient way. Testing the proposed framework on the CICIDS2017dataset shows that it can accurately detect complex attacks while keeping data private, with a detection accuracy of 93.4% and an F1-score of 92.8%.

The authors discussed about DDoS Gupta et al. (2026) as one of the serious challenges of cloud infrastructure security and availability. The authors discussed Recurrent Neural

Table 1
Comparative Analysis of Literature Review

Ref.	Methodology	Key Findings	Limitations
Ahmed et al. (2025)	SVM, KNN, RF, DT, LSTM, ANN	Compared ML and DL models for intrusion detection and prevention.	Limited interpretability and explanation of model decisions.
Varma et al. (2025); Nguyen and Harper (2026)	Signature-based + Anomaly-based detection	Improved detection of known and unknown threats through hybrid detection.	Scalability and explainability challenges remain.
Blessing et al. (2023)	Signature-based IDS and Anomaly-based IDS	Demonstrated strengths and weaknesses of both detection approaches.	Signature-based methods fail on zero-day attacks; anomaly-based methods suffer high false positives.
He et al. (2026)	LSTM-1DResNet + MLP	Achieved 94.38% and 97.98% accuracy through enhanced spatiotemporal feature extraction.	Lack of model transparency and explainability.
Kaur et al. (2026)	1D-CNN + GRU Hybrid Model	Improved spatial-temporal learning and robust feature representation.	Black-box behavior; limited analyst interpretability.
Alsarray (2026)	Federated Learning + GRU (SDN-FL-GRU)	Detection accuracy of 93.4% with privacy-preserving distributed learning.	Explainability not considered.
Gupta et al. (2026)	GRU, LSTM, Simple RNN with Feature Selection	LSTM achieved best trade-off between accuracy and computational efficiency.	Focused on performance metrics without interpretation of predictions.
Gupta and Singh (2026)	Feature Selection + Ensemble Learning + XAI	Achieved 99.95% accuracy with transparent ensemble classification.	Limited analysis of deep learning explainability.
Kumar et al. (2026)	SHAP + LIME Framework	Improved transparency, feature-level explanations, and analyst trust.	Primarily focused on explainability rather than advanced deep learning architectures.
Ghosh et al. (2026)	GAN + EfficientNet-B0 + Attention CNN + SHAP/LIME	Achieved up to 99.99% accuracy with explainable attack classification.	High model complexity and computational cost.

Network based models as Gated Recurrent Unit, Long Short-Term Memory, and Simple RNN for detecting DDoS attacks using CICDDoS2019 dataset. Feature selection was done using Information Gain, Backward Search, and Principal Component Analysis to perform on GRU, LSTM, and Simple RNN models. All models achieved accuracy of 99.70%, 99.71%, and 99.75%.

2.3. Explainable Artificial Intelligence Based Intrusion Detection Systems

In their work Gupta and Singh (2026), the authors talked about using an explainable ensemble learning-based intrusion detection system to sort different types of cyberattacks into different groups. The suggested structure is made up of three separate parts: a module for choosing features, a module for reducing features, and a vote ensemble learning module. Tenfold cross-validation is used on two benchmark datasets to thoroughly test the suggested framework's performance. On both datasets, the suggested model was accurate 99.95% of the time.

The ability to use artificial intelligence (AI) Kumar et al. (2026) to find threats is now an important part of modern cyber security. Because current ML and DL behave in a "black box" way, Explainable Artificial Intelligence (XAI) is suggested as a way to make cyber security decisions that are easier to understand. So, the intrusion detection dataset uses the SHAP (SHapley Additive Explanations) and LIME (Local Interpretable Model-Agnostic Explanations) tools to find feature-level explanations for the model forecasts. The outcome shows that using a mix of XAI techniques can make things easier to understand, boost confidence in the researchers, and allow people to make useful decisions about cyber security.

In the study Ghosh et al. (2026), the researchers suggested an Explainable Artificial Intelligence (XAI)-driven intrusion detection system (IDS) that uses deep learning and optimisation techniques to better identify DoS attacks. For adding more data, a reweighted supplementary classifier generative adversarial network with gradient penalty is used. An EfficientNet-B0 with mixed attention modules improves

Figure 2: Methodology of proposed work

the process of feature extraction and finds important patterns. We use a multiplicative Luong attention deep residual multi-scale convolutional neural network to improve the accuracy and stability of grouping network traffic samples by context features. The suggested method is tested on five datasets: NSL-KDD, CSE-CIC-IDS2018, CCIDS2019, KDD-Cup99, and UNSW-NB15. It achieves classification accuracy levels of 99.99%, 99.96%, 99.97%, and 99.95%, in that order, and it works better than the current methods.

Table 1 presents a comparative analysis based on literature review. The table shows a development of technology and methodology to identify and detect DDoS attacks. The table shows impact and evolution from traditional IDS to XAI along with their findings and limitations which are required to be addresses in further research.

3. Methodology

Our research proposes an explainable deep learning framework for network attack detection and classification. The framework combines data preprocessing, feature selection, deep learning-based classification, and SHAP-based

explainability. The objective is to accurately identify malicious network traffic and provide explanations for the model predictions. Figure 2 presents a methodology of our research work. It consists of six stages: dataset collection, data preprocessing, feature selection, deep learning-based attack detection, attack classification, and explainability analysis. The CIC-DDoS2019 dataset is used to evaluate the effectiveness of the proposed approach.

Also, simple flow of implementing the methodology is shown by Figure 3.

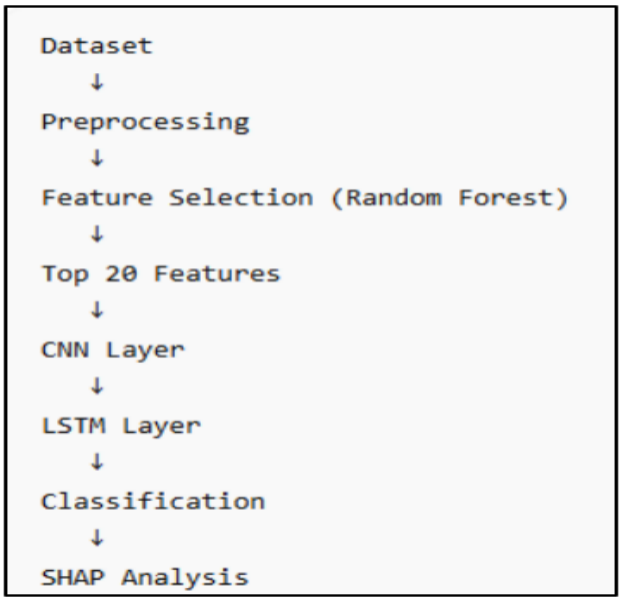

Figure 3: Flow of research methodology

3.1. Data Description

The CIC-DDoS2019 dataset is used in our proposed work. The dataset contains both benign and malicious network traffic records. It includes different types of DDoS attacks and a large number of network flow features. These features describe the behaviour of network traffic and are useful for attack detection. Figure 4 presents the distribution of various attacks in this dataset.

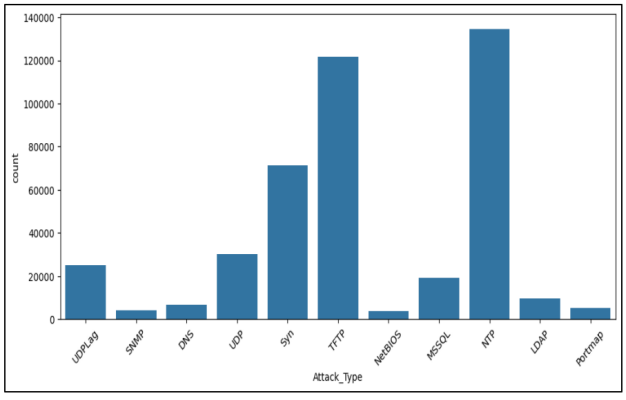

Figure 4: Types of DDoS Attack

3.2. Data Preprocessing

Data preprocessing is performed to improve data quality before model training. In the available dataset, there was no missing values, and null values, however the preprocessing is required for encoding, normalization and splitting the dataset for training and testing.

The following steps are applied:

- Conversion of categorical labels into numerical values.
- Feature normalization using Min-Max scaling.
- Division of the dataset into training 80% and testing sets 20%.

3.3. Feature Selection

The CIC-DDoS2019 dataset contains a large number of features. Feature selection will find only features which are important and relevant features. Random Forest feature importance is used to rank the features according to their contribution. The top-ranked features are selected and used for model training. While selecting a few instead of all helps to reduces computational complexity and improves learning efficiency. Figure 5 shows top 20 features selected from Random Forest feature importance technique. It can be seen from the visualization that fwd packet length min has contributed maximum as an important feature.

3.4. Deep Learning-Based Detection Model

In our proposed work, a hybrid CNN-LSTM model is adopted. Table 2 depicts the model summary. The architecture consists of CNN and LSTM both.

The CNN layer is used to learn local patterns from network traffic data. It extracts important feature representations from the selected attributes.

The LSTM layer is then applied to learn temporal relationships among the extracted features. This helps the model understand sequential patterns that may indicate malicious activities.

The extracted information is passed to fully connected layers. Finally, the Softmax function is used to classify the traffic into attack and normal categories.

Table 2: CNN-LSTM Model Summary

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 18, 64)	256
batch_normalization (BatchNormalization)	(None, 18, 64)	256
max_pooling1d (MaxPooling1D)	(None, 9, 64)	0
lstm (LSTM)	(None, 64)	33,024
dense (Dense)	(None, 64)	4,160
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 11)	715

Total params: 38,411 (150.04 KB)
Trainable params: 38,283 (149.54 KB)
Non-trainable params: 128 (512.00 B)

3.5. Attack Classification

The trained model able to predict that a network flow belongs to a benign class or to any attack class from various attack classes. During testing, the model analyses unseen network traffic and generates classification results.

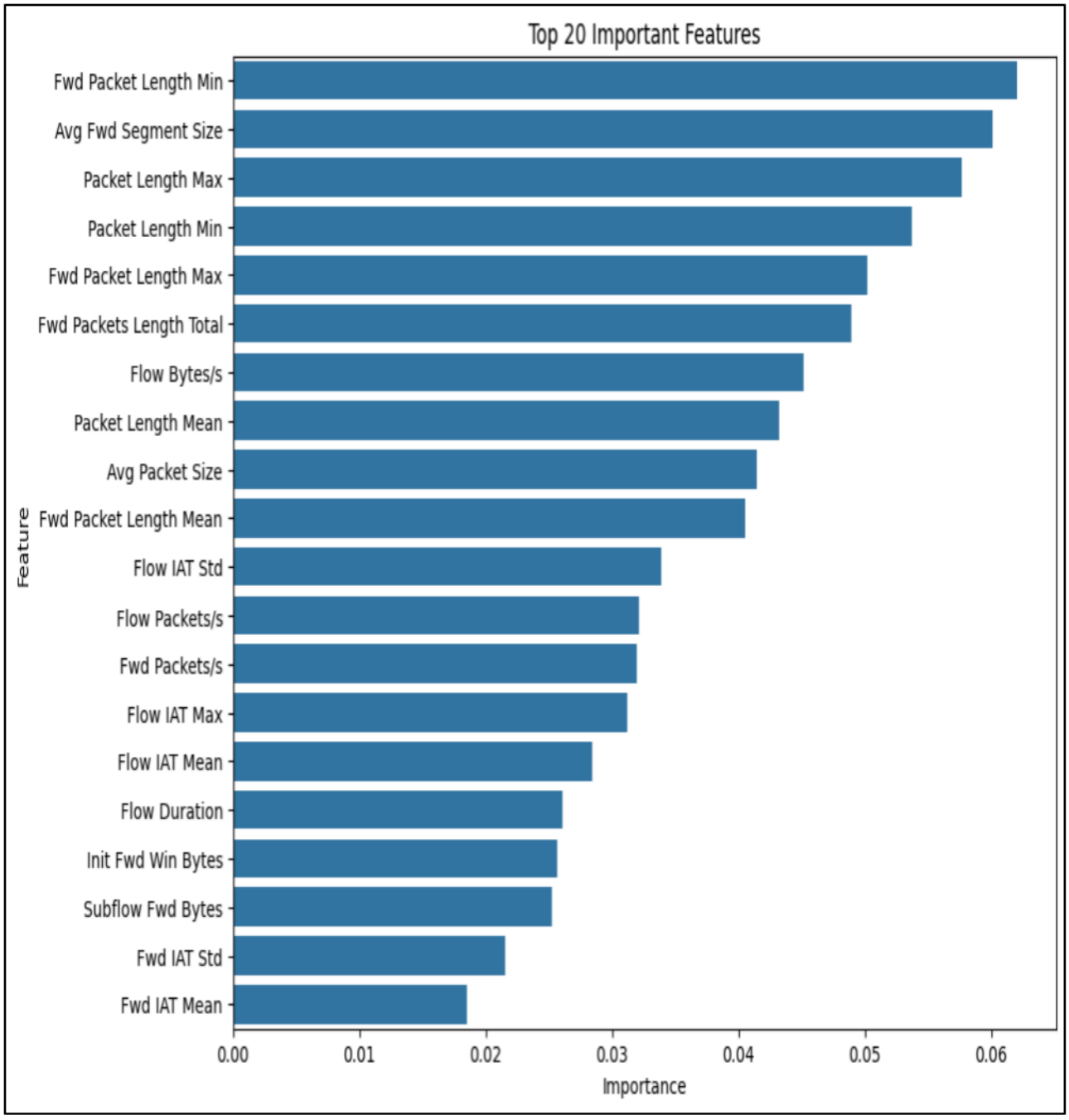

Figure 5: Top 20 selected features

3.6. SHAP-Based Explainability Analysis

Though the deep learning models provide good detection accuracy, yet their decisions are hard to be interpreted and understand sometimes due to lack of reasoning. To strengthen this part and improve transparency, SHAP (SHapley Additive exPlanations) tools is integrated into the framework.

SHAP objective is to measures each feature's contribution from beginning to the final prediction. Therefore, how much each feature affecting the network can be identified and accordingly decision can be taken. The tool generates two levels of explanation as discuss below:

3.6.1. Global Explanation

Global explanations present the overall importance of features across the entire dataset. It provides feature ranking and a summary plot. These two things are used to identify the most important and influential parameters among all the parameters.

3.6.2. Local Explanation

Local explanations provide reasoning for individual predictions. It creates a reason base to categorize a specific network flow into normal or attack category. The SHAP tool

improves the interpretability of the proposed framework and make it more understandable for decision making.

3.6.3. Interpretable and Explainable

At the last, it focuses on interpreting the results obtained from SHAP analysis. It helps to understand attack behavior and supports decision-making during suspicious traffic identification from the normal traffic.

The integration of explainability with deep learning provides both accurate detection and transparent decision support for network security applications.

4. Experimental Setup

For the experiment, we use the following setup:

- Python 3.10.4 (64-bit)
- Jupyter Notebook environment integrated with Visual Studio Code
- 11th Generation Intel Core i5-1135G7 processor operating at a clock speed ranging from 2.40 GHz to 3.32 GHz
- 16 GB DDR4 RAM for efficient execution of the proposed model

In addition, the storage configuration consisted of a 1 TB hard disk drive (HDD) and a 256 GB solid-state drive (SSD), which provided adequate storage capacity and improved data access speed during data processing and model training.

5. Performance Evaluation Metrics

Though the problem statement is a categorical problem. Therefore, to evaluate the proposed models, we use the following metrics Gupta and Singh (2026); Kumar et al. (2026); Ghosh et al. (2026):

5.1. Accuracy

It presents the overall correctness of the classification model by calculating the ratio of correctly predicted instances to the total number of instances.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

5.2. Precision

It evaluates the number of correctly identify positive instances while reducing false positive predictions.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

5.3. Recall

Recall measures correctly detect actual positive instances.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

5.4. F1-score

The F1-score performs harmonic mean between precision and recall and presents a balance between these two.

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

In the case of imbalance dataset, F1-score is very useful. Therefore, instead taking only accuracy, we need to have all four metrics to see the performance.

6. Result and Discussion

Our proposed framework implemented a hybrid CNN-LSTM architecture. In this architecture, the CNN layer was utilized to extract local traffic patterns from the selected network features, while the LSTM layer captured temporal dependencies among the extracted representations. Batch normalization and dropout layers were incorporated to improve model stability and reduce overfitting. Finally, a Softmax output layer was used to classify the network traffic into eleven attack categories.

The classification performance of the CNN-LSTM model across eleven attack classes was presented by Figure 6. Most attack categories were correctly classified, particularly the dominant classes such as NTP, TFTP, and Syn attacks. When we talk about DDoS and its various type under network traffic, then it is interesting to see which variant of DDoS affects the network most and hard to be identified by applying traditional mechanisms.

Figure 6: Confusion matrix

Learning curve is also important to be plotted. It shows how the model is learning during training time and validation

time. The losses while the training and validation are enough to give information about the overfitting or underfitting issues. We also present these two training and validation curve through our experiments.

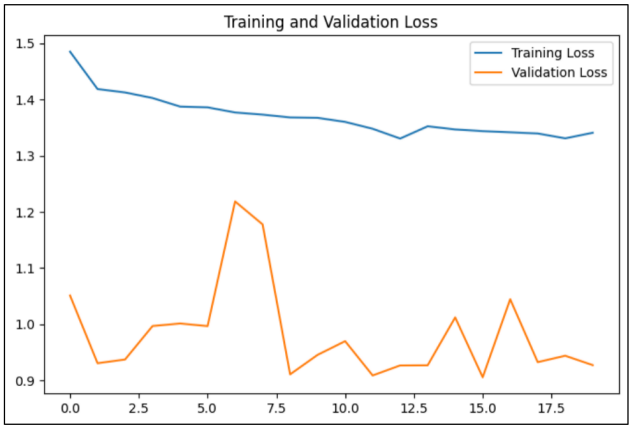

Figure 7: Training and Validation Loss

Figure 7 depicts the variation of training and validation loss across training epochs. A gradual reduction in training loss can be seen that interprets the effective learning of attack patterns by the model. The relatively stable validation loss recommends that the model maintains acceptable generalization performance without severe overfitting.

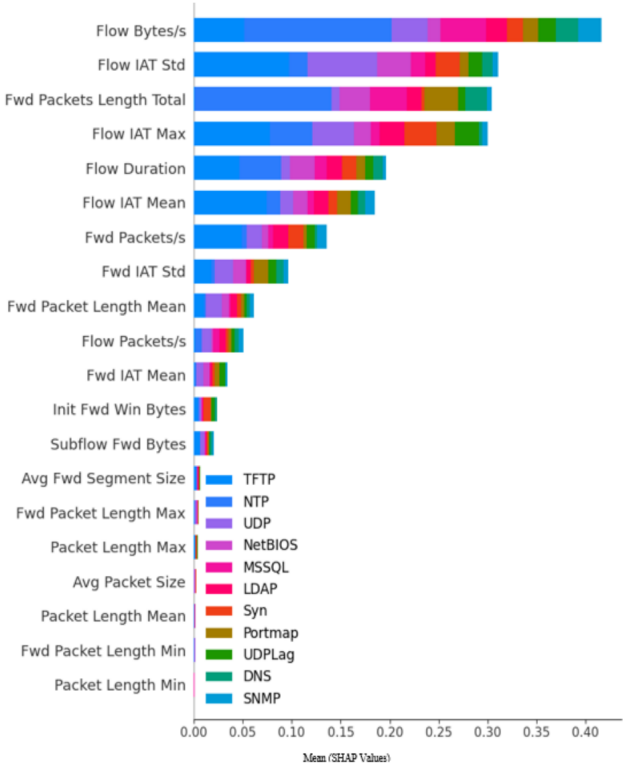

Figure 8: Shap Analysis

Figure 8 presents the global feature importance obtained using SHAP analysis. The results reveal that Flow Bytes/s, Flow IAT Std, and Fwd Packets Length Total are the most influential features affecting attack classification decisions. These findings indicate that traffic volume and packet timing characteristics are critical indicators of DDoS attack behavior.

SHAP analysis was conducted to improve the interpretability of the proposed CNN-LSTM framework. The results revealed that Flow Bytes/s was the most influential feature affecting model decisions, followed by Flow IAT Std, Fwd Packets Length Total, and Flow IAT Max. These findings interpret characteristics of traffic volume and packet timing that helps in differentiating variants of DDoS attack. The SHAP tool analysis shows that the proposed framework depends more on the network traffic behavior rather than arbitrary feature patterns, therefore it also helps to improve the model transparency and trustworthiness.

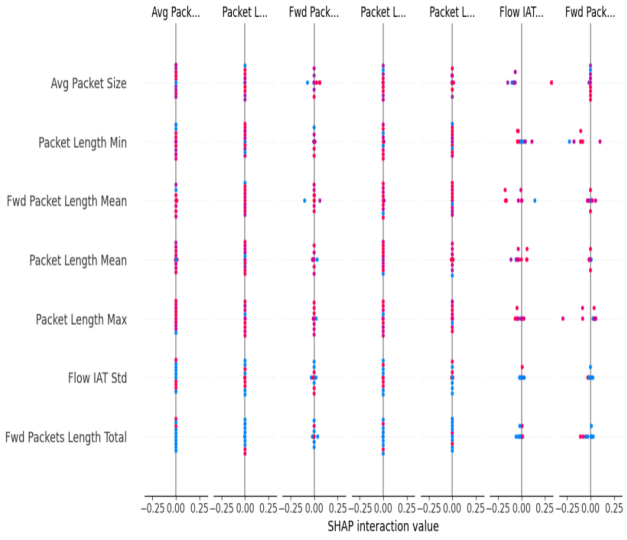

Figure 9: Shap Interaction Value

Figure 9 depicts the interaction effects among the most significant network traffic features. The visualization shows that different attack categories are affected by different combinations of flow statistics and packet characteristics. These explanations enhance the transparency of the proposed model and provide valuable insights.

Table 3: Evaluation Metrics of CNN+LSTM

Metric	Value
Accuracy	75.39%
Weighted Precision	73%
Weighted Recall	75%
Weighted F1-Score	72%
Macro F1-Score	43%

Table 3 presents the performance of the proposed hybrid model, the proposed CNN-LSTM model achieved a classification accuracy of 75.39% on the testing dataset. The weighted F1-score reached 72%, indicating satisfactory overall classification performance. The model shows a strong performance for major attack categories such as NTP, TFTP, UDP, and Syn attacks. However, lower performance was observed for minority attack classes due to limited learning in those classes. Limited learning in a class means imbalance

issues. Despite this limitation, the model successfully learned discriminative attack patterns from the selected network traffic features.

To address the issue of class imbalance, weights were incorporated in those classes during model training. The weighted model improved the macro F1-score from 0.43 to 0.48, indicating better recognition of minority attack classes. Although the overall accuracy decreased to 69%, the model achieved more balanced classification performance across different attack categories. This result highlights the trade-off between overall accuracy and minority-class detection capability.

Table 4: Evaluation Metrics of CNN+LSTM (weighted F1)

Model	Accuracy	Macro F1	Weighted F1
CNN-LSTM	75.39%	0.43	0.72
CNN-LSTM + Class Weights	69.00%	0.48	0.75

7. Conclusion

In our proposed research work, an explainable deep learning framework was developed for multi-class DDoS attack classification using the CIC-DDoS2019 dataset. We applied Random Forest feature selection to identify the most important and relevant network traffic features to train the model.

The experimental results showed that the proposed model achieved an accuracy of 75.39% and a weighted F1-score of 72%. The use of class-weighted learning also improved the detection of less frequent attack classes. These results indicate that deep learning methods can be effectively used for network intrusion detection and attack classification.

To improve the understanding of model decisions, SHAP analysis was employed. The analysis identified Flow Bytes/s, Flow IAT Std, and Fwd Packets Length Total as the most influential features in the classification process. The explanations generated by SHAP help security analysts understand the factors responsible for attack detection.

The proposed framework provides both effective attack classification and improved interpretability. In future work, advanced deep learning models and larger datasets may be explored to further enhance detection performance and reliability in practical network environments.

References

Abdulganiyu, O. H., Ait Tchakoucht, T., and Saheed, Y. K. (2023). A systematic literature review for network intrusion detection system (ids). *International Journal of Information Security*, 22(5):1125.

Ahmed, U., Nazir, M., Sarwar, A., Ali, T., Aggoune, E. H. M., Shahzad, T., and Khan, M. A. (2025). Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering. *Scientific Reports*, 15(1):1726.

Alsarray, Z. A. (2026). Gru-based federated learning for privacy-preserving intrusion detection in sdn-enabled iot networks. *Journal Port Science Research*, 9(2):405–413.

Blessing, A. I., Chole, A., and Themmah, Y. (2023). Signature-based vs. anomaly-based detection.

Ghosh, S., Goyal, R. K., and Chowdhury, K. (2026). Explainable ai-driven intrusion detection system for dos attack classification using deep learning and optimization techniques. *IEEE Access*, 14:5618–5642.

Gupta, S., Kaveri, P. R., Gupta, K. K., Awasthi, P., and Shaik, M. (2026). Performance comparison of gru, lstm, and rnn models for detecting ddos attacks in cloud infrastructure. In *AIP Conference Proceedings*, volume 3410, page 020069. AIP Publishing LLC.

Gupta, S. and Singh, B. (2026). Lightweight ensemble learning based intrusion detection framework with explainable artificial intelligence. *Engineering Applications of Artificial Intelligence*, 163:1129366.

He, Q., Zhang, Y., Xu, A., Ye, Z., Zhou, W., Lin, Q., and Zhang, T. (2026). Lstm-1dresnet: An intrusion detection model for connected and autonomous vehicles based on deep learning. *IEEE Transactions on Vehicular Technology*.

Imrana, Y., Xiang, Y., Ali, L., and Abdul-Rauf, Z. (2021). A bidirectional lstm deep learning approach for intrusion detection. *Expert Systems with Applications*, 185:115524.

Kaur, S., Bajaj, R., Bansal, S., and Liu, H. (2026). Learning spatio-temporal patterns for network intrusion detection using a hybrid cnn-gru model. In *2026 13th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 1–6. IEEE.

Kumar, P. V. P., Swetha, M., Kumari, P. M., Akshitha, P., Sreeja, R., and Harshitha, R. (2026). Explainable ai for cyber security: Interpretable threat analysis using shap and lime. In *2026 13th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 1–6. IEEE.

Mishra, A., Gupta, N., Gupta, B. B., Bhatia, K., and Aswal, M. S. (2024). Prediction of variants of ddos attacks based on statistical analysis and machine learning algorithms. *International Journal of Innovative Computing and Applications*, 15(1):14–25.

Mohammadpour, L., Ling, T. C., Liew, C. S., and Aryanfar, A. (2022). A survey of cnn-based network intrusion detection. *Applied Sciences*, 12(16):8162.

Nguyen, S. K. and Harper, J. P. (2026). Hybrid signature and anomaly-based detection for api-centric microservices.

Varma, M. A. D., Vaasist, G. S., Reddy, B. C., Reddy, M. P., and Nair, R. (2025). Intrusion detection system using signature and anomaly based algorithm. In *2025 International Conference on Inventive Computation Technologies (ICICT)*, pages 838–842. IEEE.