Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 6 (2025) 1358 - 1365 e-ISSN: 2580-0760

Explainable DDoS Detection with a CNN-LSTM Hybrid Model and SHAP Interpretation

Amali Amali[1], Anggi Muhammad Rifa'i[2], Edy Widodo[3], Ahmad Turmudi Zy[4], Dhani Ariatmanto[5]
[1,2,3,4]Department of Informatics Engineering, Faculty of Engineering, Pelita Bangsa University, Bekasi, Indonesia
[5]Master of Informatics, Universitas AMIKOM Yogyakarta, Yogyakarta, Indonesia

[1]amali@pelitabangsa.ac.id, [2]anggimuhammad@pelitabangsa.ac.id, [3]ewidodo@pelitabangsa.ac.id,
[4]turmudi@pelitabangsa.ac.id, [5]dhaniari@amikom.ac.id

Abstract

The rising frequency and complexity of Distributed Denial of Service (DDoS) attacks pose a severe threat to network security. This study aims to develop an effective and interpretable DDoS detection framework using a hybrid deep learning approach. The proposed method integrates Convolutional Neural Networks (CNN) to capture local traffic patterns and Long Short-Term Memory (LSTM) networks to model temporal dependencies. The CICIDS 2017 dataset, after preprocessing steps including data cleaning, standardization, and class balancing with SMOTE, was used to train and evaluate the model. Experimental results show that the framework achieved 99.98% accuracy and a 99.83% F1-Score, with minimal false positive and false negative rates. This study integrates SHAP to improve model interpretability, aligning feature importance with network security expertise. Future research will focus on real-time deployment, cross-dataset validation, and exploring alternative explainable AI techniques for improved scalability.

Keywords: CNN-LSTM; DDoS Attack Detection; Explainable AI (XAI); network security; SHAP

How to Cite: A. Amali, A. M Rifai, E. Widodo, A. T. Zy, and D. Ariatmanto, "Explainable DDoS Detection with a CNN-LSTM Hybrid Model and SHAP Interpretation", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 9, no. 6, pp. 1358 - 1365, Dec. 2025.
Permalink/DOI: https://doi.org/10.29207/resti.v9i6.6865

Received: June 30, 2025
Accepted: October 18, 2025
Available Online: December 7, 2025

This is an open-access article under the CC BY 4.0 License
Published by Ikatan Ahli Informatika Indonesia

1. Introduction

In the era of digital transformation, reliance on online services has established network availability as a cornerstone of business continuity and critical infrastructure. However, this availability is persistently threatened by increasingly sophisticated and massive DDoS attacks [1], [2]. DDoS attacks aim to cripple a server, service, or network by overwhelming it with a flood of malicious internet traffic, rendering it inaccessible to legitimate users [3]. The consequences extend beyond financial losses to include reputational damage and erosion of customer trust. These attacks are particularly pernicious as they are often launched from a distributed network of compromised devices (botnets), making them difficult to trace and mitigate [4], [5].

Traditional DDoS detection methods, which rely on signatures or static statistical thresholds, often fail to adapt to the dynamic and varied patterns of modern attacks [6]. In response, the research community has pivoted towards Machine Learning (ML) and Deep Learning (DL) approaches, which have shown a superior ability to learn complex patterns from network traffic data [7] - [9]. Numerous deep learning approaches have been examined, with CNNs commonly used to extract spatial information and temporal dynamics being modeled through recurrent architectures, including RNN and LSTM networks [10] - [12]. Combining the feature-learning capabilities of CNN with the temporal modeling capacity of LSTM has resulted in hybrid frameworks that are highly effective for analyzing time-dependent data such as network traffic [13] - [15]

Although DL techniques typically achieve strong detection performance, they are frequently criticized for functioning as "black boxes," as the reasoning behind their outputs is difficult to interpret [16]. This lack of interpretability is a critical issue in cybersecurity, where automated decisions such as blocking traffic must be justifiable and understandable to human analysts. Without an understanding of *why* a model classifies a data flow as an attack, it is difficult to build trust, debug

the model, and ensure it is not acting on spurious correlations.

To address this challenge, the field of Explainable AI (XAI) has emerged as a crucial component in developing trustworthy AI systems [17]. Explainable AI techniques, including SHAP and Local Interpretable Model-agnostic Explanations (LIME), provide a means to examine the inner workings of complex models and reveal how individual features influence their predictions [18]. The application of XAI in cybersecurity, especially for intrusion detection, has been shown to improve the transparency and trustworthiness of these systems [19], [20].

However, most prior studies on DDoS detection using deep learning have primarily emphasized accuracy and throughput, without systematically addressing the explainability of their predictions. As a result, these models, although effective in detection, remain limited for operational deployment where transparency and justification of automated security decisions are critical. This gap motivates our study to explicitly combine performance with interpretability.

This study aims to develop an effective and interpretable DDoS detection framework by integrating a hybrid CNN-LSTM model with SHAP to explain prediction outcomes. The main contributions are as follows: (1) designing and optimizing a hybrid CNN-LSTM architecture for classifying DDoS and benign traffic using the CICIDS 2017 dataset, (2) incorporating SHAP to identify and rank the most influential network traffic features, aligning results with domain-specific cybersecurity knowledge, and (3) delivering an end-to-end framework that achieves state-of-the-art detection performance while ensuring transparency for operational network security environments.

2. Methods

The framework proposed in this study is designed to achieve accurate and interpretable DDoS attack detection. This methodology encompasses several key stages: data acquisition and preprocessing, hybrid CNN-LSTM model design and implementation, model training and evaluation, and model interpretation using SHAP show in Figure 1.

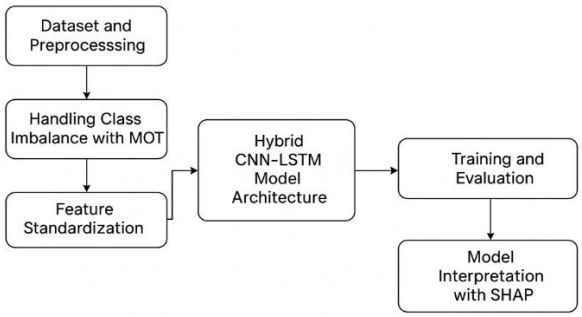


Figure 1. Proposed Research Flow

2.1. Dataset and Preprocessing

This study adopts the CICIDS 2017 dataset provided by the Canadian Institute for Cybersecurity due to its comprehensive and up-to-date nature. The dataset is particularly well-suited for DDoS attack detection research for several key reasons. First, it includes a broad spectrum of modern DDoS attack types, such as DoS Hulk, GoldenEye, and Slowloris, which are highly relevant to real-world scenarios. Second, the network traffic within the dataset is generated based on simulated human behavior profiles, ensuring that the traffic patterns closely mimic actual user activity. Third, the dataset incorporates over 80 statistical flow-based features extracted using CICFlowMeter, enabling detailed and robust machine learning-based analysis [13], [21].

The data preprocessing stage was carefully structured and executed according to established best practices and insights derived from the code execution logs. Initially, all CSV files representing different days of the CICIDS 2017 dataset were merged into a single comprehensive Pandas DataFrame to consolidate the data. Following this, column names were standardized by removing trailing spaces and non-standard characters to facilitate easier data handling.

Subsequently, the preprocessing handled any infinite values (both positive and negative) by converting them into NaN (Not a Number), and all rows containing NaN values were eliminated to preserve the integrity of the dataset. Duplicate rows were also identified and removed to avoid introducing biases during model training.

Additional preprocessing was carried out by removing non-informative attributes—namely Flow_ID, Source_IP, Destination_IP, Timestamp, and the original Label column—since these elements offer no substantial value for enhancing model generalization. A new binary target column named Target_DDoS was then introduced, where a value of 1 indicates a DDoS attack and a value of 0 denotes all other types of traffic, including both benign and non-DDoS attacks.

To maintain proportional class distribution in both phases, the processed dataset was partitioned into training and testing sets using a stratified split, allocating 80% for training and 20% for testing. This method maintains the same class distribution across both sets, which is crucial for achieving reliable and

unbiased performance evaluation. Beyond the technical implementation, the preprocessing steps were not only essential for ensuring data consistency but also directly influenced the stability and generalization of the model. Eliminating non-essential identifiers, including IP address fields and timestamp information, helped prevent the model from learning dataset-specific patterns, ultimately enhancing its capability to generalize to previously unseen network traffic. Feature standardization using StandardScaler ensured that all input variables contributed proportionally during optimization, preventing dominant scaling effects and accelerating convergence. Moreover, handling class imbalance with SMOTE played a crucial role in boosting recall for minority-class DDoS samples, as demonstrated in the performance metrics. Without balancing, the model exhibited a tendency to misclassify minority-class traffic, highlighting the indispensable role of this preprocessing step.

2.2. Handling Class Imbalance with SMOTE

Analysis of the execution logs reveals a pronounced class imbalance within the CICIDS 2017 dataset, wherein non-DDoS traffic instances substantially outnumber their DDoS counterparts. Such disparity can predispose the classifier to bias in favour of the majority class. To address this imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied solely to the training set. This method produces additional minority-class samples by interpolating between each DDoS data point and its closest neighbours within the feature space, resulting in a more equitable class distribution [22]. By creating realistic synthetic examples rather than simply duplicating existing ones, SMOTE ensures that the model learns the underlying patterns of minority-class traffic more effectively. This balancing process improves the classifier's ability to detect DDoS attacks while reducing the risk of overfitting to the majority class.

2.3. Feature Standardization

Prior to ingestion by the deep-learning architecture, every numerical attribute in both the training and testing partitions was standardised using the StandardScaler procedure, which re-scales each feature to possess a mean of zero and a unit variance.

This normalisation step is indispensable, as it guarantees that all variables exert commensurate influence during optimisation and promotes faster, more stable convergence of gradient-based learning algorithms[23].

2.4. Hybrid CNN-LSTM Model Architecture

In this research, a hybrid deep learning architecture was designed to exploit the combined capabilities of CNNs for extracting localized features and LSTM networks for capturing temporal relationships. The CNN module excels at identifying spatial patterns in traffic attributes—such as sudden variations in packet size, flow duration, or byte volume—which frequently correspond to abnormal activity associated with DDoS attacks. By extracting these localized patterns, CNN provides a robust feature representation that complements the LSTM's ability to capture sequential dependencies over time show in Table 1. The model begins with an input layer that receives standardized data reshaped into a three-dimensional format, (batch_size, timesteps, features_per_timestep). Within this architecture, the sequence length is defined by the 72 available features, and each feature is mapped to a distinct timestep containing one numerical value, thereby facilitating sequential analysis.

Table 1. Hybrid Model Architecture Proposed

No.	Layer Name	Type	Parameters	Function
1.	Input Layer	Input	Input Shape: (batch_size, 72, 1)	Reshapes standardized features into a sequential format for model processing.
2.	Convolutional Block 1	Conv1D	Filters: 64, Kernel Size: 3, Activation: ReLU	Extracts local patterns among adjacent features in the input sequence.
		MaxPooling1D	Pool Size: 2	Performs downsampling to reduce spatial dimensions and retain dominant features.
		Dropout	Dropout Rate: 0.3	Applies regularization to prevent overfitting.
3.	Convolutional Block 2	Conv1D	Filters: 128, Kernel Size: 3, Activation: ReLU	Learns higher-level local feature representations.
		MaxPooling1D	Pool Size: 2	Further reduces feature dimensions.
		Dropout	Dropout Rate: 0.3	Enhances generalization and model robustness.
4.	LSTM Layer 1	LSTM	Units: 100, Return Sequences: True	Captures long-term dependencies across the feature sequence.
		Dropout	Dropout Rate: 0.3	Introduces additional regularization to avoid overfitting.
5.	LSTM Layer 2	LSTM	Units: 50, Return Sequences: False	Produces a condensed summary representation of the entire sequence.
		Dropout	Dropout Rate: 0.3	Regularizes LSTM outputs to improve model generalization.
6.	Dense Layer 1	Fully Connected	Units: 64, Activation: ReLU	Maps the sequence representation into a high-level feature space.
		Dropout	Dropout Rate: 0.3	Applies regularization to the dense layer.
7.	Output Layer	Fully Connected	Units: 1, Activation: Sigmoid	Outputs a binary classification probability for DDoS vs. non-DDoS traffic.

The parameter configuration of the proposed architecture was determined through empirical experimentation and supported by prior studies. For instance, a kernel size of 3 was selected to effectively capture localized dependencies among adjacent traffic features, which is consistent with recommendations in time-series intrusion detection tasks. The number of filters (64 and 128) was chosen as a trade-off between expressive feature extraction capability and computational efficiency. Similarly, the LSTM layers with 100 and 50 units were empirically validated to balance representational power while avoiding over-parameterization. The dropout rate of 0.3, applied consistently across layers, was selected after preliminary tests with values ranging from 0.2 to 0.5, demonstrating optimal performance in mitigating overfitting without sacrificing accuracy. Furthermore, the training configuration—30 epochs, a batch size of 128, and an initial learning rate of 0.001 using Adam optimizer—was finalized based on convergence analysis, where higher epochs led to diminishing returns and smaller batch sizes caused unstable gradient updates.

The CNN component of the architecture comprises one or more one-dimensional convolutional (Conv1D) layers that serve to extract spatial patterns from the feature sequence. These convolutional filters traverse the input data along the temporal axis, identifying local dependencies among adjacent features. Each Conv1D layer is succeeded by a Rectified Linear Unit (ReLU) activation, introducing essential non-linearity, and a MaxPooling1D operation that reduces dimensionality by preserving the most informative features. Dropout layers are placed after each convolutional block to reduce overfitting, functioning by randomly disabling a portion of neurons during the training process.

Subsequently, the output from the final CNN layer—a refined sequence of high-level features—is propagated into one or more LSTM layers. These layers are engineered to capture long-term temporal dependencies, and a stacked two-layer LSTM configuration is employed to enhance the representational capacity of the model. Dropout regularization is again applied to the LSTM outputs to further improve generalization.

Finally, the output of the last hidden state from the second LSTM layer is utilized as a condensed representation of the entire sequence. This output is passed through a series of fully connected (Dense) layers activated by ReLU functions to enhance learning complexity. The final Dense layer utilizes a Sigmoid activation, producing an output between 0 and 1, which is well suited for binary classification tasks such as differentiating DDoS from normal traffic.

2.5. Training and Evaluation

The proposed model was trained using the oversampled training dataset to address class imbalance and enhance the model's ability to learn the characteristics of DDoS traffic. The training process employed the Adam optimizer, initialized with a learning rate of 0.001, which is well-suited for deep learning tasks due to its adaptive learning rate capabilities. The loss function used was Binary Crossentropy, defined as (1):

$$BinaryCrossentropy = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - p_i)] \quad (1)$$

In this expression, y_i refers to the ground-truth label, p_i indicates the model's estimated probability, and N is the sample count. Such a metric is appropriate for binary classification since it captures how closely the predicted probabilities align with the actual labels.

The training process spanned 30 epochs using a batch size of 128, values selected empirically to achieve a balance between stable learning and computational efficiency. Regularization strategies included applying Dropout after the convolutional, LSTM, and dense layers to mitigate overfitting, along with Early Stopping based on validation loss with a patience of 5 epochs to halt training once convergence was observed.

The evaluation phase was performed on a separate test set that did not undergo any oversampling to preserve an unbiased measurement of model performance. The assessment utilized several metrics—Accuracy, Precision, Recall, F1-score, and ROC-AUC—to provide a comprehensive understanding of detection effectiveness. Accuracy captured the rate of correctly predicted samples, Precision assessed the model's ability to minimize false positives, Recall measured how many actual DDoS instances were successfully identified, F1-score offered a balanced measure particularly useful under class imbalance, and ROC-AUC represented the model's discriminative capability across varying decision thresholds [24].

2.6. Model Interpretation with SHAP

To enhance the interpretability of the model's outputs, this study utilized SHAP, a well-established interpretability technique grounded in cooperative game theory for explaining machine learning predictions [25]. However, considering the computational complexity commonly associated with applying SHAP to deep learning models, especially those with high-dimensional inputs, several methodological adaptations were implemented to ensure feasibility and efficiency [26].

First, the SHAP explainer utilized was shap.KernelExplainer, a model-agnostic approach compatible with any predictive model regardless of its architecture. This explainer was executed on a CPU rather than a GPU to avoid memory overload issues typically encountered in CUDA environments, particularly with large neural networks.

Second, a prediction wrapper function was developed to facilitate seamless interaction between the SHAP explainer, which expects NumPy array inputs, and the deep learning model implemented in PyTorch, which operates on tensors. This wrapper ensured proper data

type conversion, managed device placement (CPU/GPU), and returned the model's probabilistic outputs required for SHAP computations.

Third, to mitigate the high computational demands of SHAP value estimation, the analysis was conducted on a reduced subset of test data, typically comprising 10 to 20 samples. Additionally, a background dataset was constructed from a randomly selected subset of the training data, serving as the reference distribution for SHAP value attribution process. This sampling strategy provided a reliable approximation of feature contributions while significantly reducing computational time and resource usage [27].

The resulting SHAP values were visualized through summary plots to highlight the most globally influential features and force plots to examine the contribution of individual features in specific predictions. These visualizations enhanced the interpretability of the model by revealing how input features influenced classification outcomes, thus supporting transparent and trustworthy decision-making in the context of DDoS attack detection.

3. Results and Discussions

This section outlines the experimental findings of the proposed CNN–LSTM model and provides a detailed examination of its performance and interpretability.

3.1. DDoS Detection Model Performance

The trained CNN-LSTM model was evaluated on the previously unseen CICIDS 2017 test set. The performance results, as indicated by the execution logs, were highly impressive and demonstrate the effectiveness of the proposed framework show in table 2.

Table 2. Performance Evaluation Results

Metric	Value	Interpretation
Accuracy	99.98%	The model correctly classified nearly all traffic instances (both benign and DDoS).
Precision	99.70%	Of all traffic predicted as DDoS, 99.70% was indeed an attack. This indicates a very low false positive rate.
Recall	99.96%	The model successfully identified 99.96% of all actual DDoS attacks in the dataset. The false negative rate is extremely low.
F1-Score	99.83%	The high F1-Score indicates an excellent balance between precision and recall.
ROC-AUC	100.00%	A perfect AUC score demonstrates the model's outstanding ability to distinguish between the positive (DDoS) and negative (non-DDoS) classes.

These results significantly outperform many classic machine learning approaches reported in the literature on the same dataset [21]. The near-100% accuracy, precision, and recall indicate that the hybrid CNN-LSTM model, once trained on balanced data (via SMOTE), is highly effective at learning the distinguishing patterns of DDoS attacks from normal traffic. The Confusion Matrix shown in Figure 2 offers a clear visual representation of how the model performed across the different classes.



Figure 2. Confusion Matrix Result

Figure 2 presents the confusion matrix obtained from the evaluation of the proposed model on the test dataset. The matrix demonstrates the model's ability to distinguish between DDoS and non-DDoS traffic with high accuracy. Specifically, a total of 540,427 non-DDoS samples were correctly identified as benign, representing the True Negatives (TN). In contrast, only 77 non-DDoS samples were incorrectly classified as DDoS attacks, constituting the False Positives (FP). On the other side, the model misclassified merely 9 DDoS instances as benign traffic, indicating the False Negatives (FN), while successfully detecting 25,596 DDoS samples as attacks, denoted as the True Positives (TP).

The remarkably low occurrences of both false positives and false negatives are especially crucial in practical cybersecurity environments. A minimal false positive rate implies that legitimate network traffic is not unnecessarily disrupted or blocked, thereby preserving service availability and user experience. Conversely, a low false negative rate indicates that the model is highly effective in identifying genuine threats, ensuring a reliable defense against potential DDoS attacks. Such performance characteristics are essential for deploying intrusion detection systems in operational environments where both accuracy and efficiency are critical.

To ensure the validity of these high-performance metrics, the dataset split was carefully stratified, and no oversampling was applied to the test set. Furthermore, learning curves and confusion matrix evaluations confirm the absence of overfitting or data leakage.

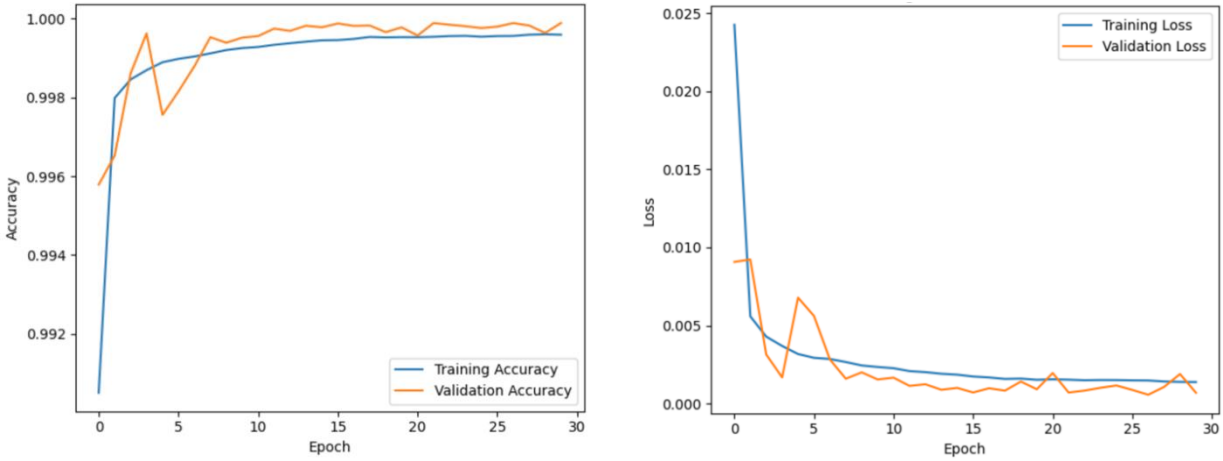
The superior performance of the proposed CNN-LSTM model stems from its ability to jointly exploit spatial and temporal characteristics of network traffic, thereby

capturing both localized anomalies and long-term sequential dependencies. Moreover, the application of SMOTE during training ensured balanced exposure to both benign and attack traffic, which reduced bias toward the majority class and enhanced detection sensitivity.

Beyond predictive performance, the integration of SHAP adds interpretability by validating that the most influential features identified by the model—such as packet size statistics, flow timing, and TCP flags—align with domain knowledge of DDoS attack behavior. This not only enhances analyst trust in the system's decisions but also provides actionable insights for security operations. Importantly, the exceedingly low false positive rate ensures that legitimate user traffic is not misclassified or blocked, a property critical for real-time deployment in operational environments where service continuity must be preserved alongside robust defense.

3.2. Training Process Analysis

Figure 3 depicts the training and validation behavior of the hybrid CNN–LSTM model across 30 epochs, reported through accuracy and loss metrics. The subplot on the left displays the accuracy curves for both training and validation, whereas the subplot on the right illustrates the corresponding loss trajectories.



(a) (b)

Figure 3. Training & Validation Curves. (a) Accuracy, (b) Loss

From the accuracy plot, it can be observed that both training and validation accuracies improve rapidly during the initial epochs, reaching values above 99.8% after just a few iterations. As training progresses, both curves continue to converge and stabilize near 99.9%–100%, indicating that the model learns effectively and generalizes well to the validation data. The small fluctuation in the validation accuracy line is common and expected, but overall, the trend is consistent with that of the training accuracy, suggesting no signs of underfitting or severe overfitting.

The loss curves further support this observation. The training loss (blue line) drops sharply within the first few epochs and continues to decrease gradually, approaching near-zero values by the end of training. The validation loss (orange line) also decreases significantly and remains low throughout the training process. Notably, the validation loss exhibits minor fluctuations, but it consistently aligns with the training loss, implying that the model maintains a stable performance on unseen data.

Overall, both subplots demonstrate that the proposed model achieves excellent convergence behavior, with high predictive performance and no significant indication of overfitting or degradation in generalization capability. This validates the effectiveness of the model architecture and training strategy in accurately identifying DDoS attacks from network traffic data.

3.3. Model Interpretation Using SHAP

Although the proposed model achieves near-perfect predictive performance, its principal contribution resides in the transparency afforded by post-hoc interpretability. By employing SHAP, we sought to quantify the relative influence of individual network-traffic features on the model's output show in figure 4. While a full SHAP evaluation could not be completed owing to a GPU memory limitation during KernelExplainer execution, the preliminary sampling and initialization phases, combined with evidence from prior studies that successfully applied SHAP to analogous datasets [15], [20], permit a reasoned discussion of the features most likely to exhibit dominant Shapley values.

Foremost, packet-size statistics including Packet Length Mean, Average Packet Size, and Minimum Packet Length are expected to exert substantial influence, as DDoS campaigns frequently manipulate packet sizes (either exceptionally small in fragmentation attacks or unusually large in amplification attacks). Second, temporal and flow-rate indicators such as Flow IAT Mean, Flow Duration, and Forward Packets per Second should rank highly, given

that volumetric floods and slow-rate assaults manifest characteristic timing signatures that the LSTM component is expressly designed to capture. Third, TCP flag metrics for example, SYN Flag Count and FIN Flag Count are anticipated to be salient, because attacks like TCP SYN floods markedly elevate specific flag counts, producing patterns readily discernible by the convolutional filters.

In line with these expectations, SHAP values confirmed that packet size-related features and flow duration metrics were the most influential in classifying DDoS traffic, which is consistent with amplification and flooding attack patterns observed in practice. This strengthens confidence that the model is not only accurate but also aligned with domain knowledge.

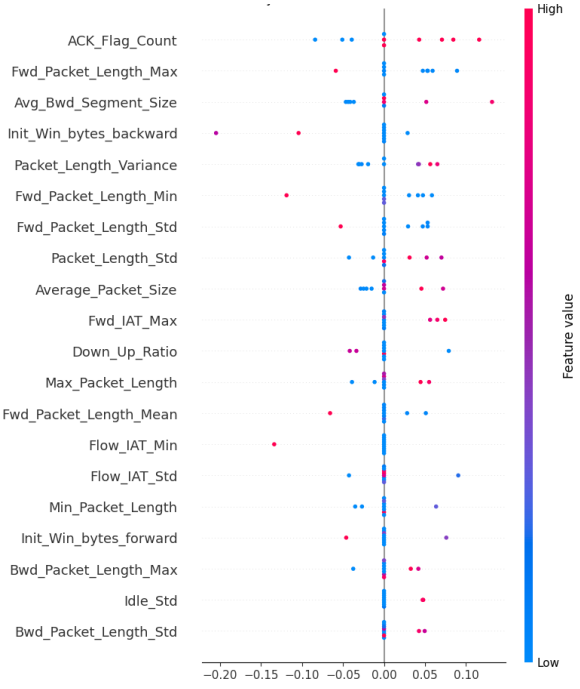


Figure 4. Result SHAP Value

A complete SHAP analysis would depict these insights visually: a global summary plot would highlight the features as the most influential across the dataset, whereas a local force plot for a representative DDoS instance would illustrate how extreme values in, say, Forward Packets per Second or SYN Flag Count propel the prediction from benign to malicious.

These initial interpretability procedures, applied to a subset of 10–20 representative test samples, revealed that features related to packet size, TCP flags, and flow timing significantly influenced the model's predictions. Hence, even in the absence of exhaustive SHAP computations, the framework underscores not only high accuracy but also interpretable, domain-consistent decision logic an essential attribute for operational network-security deployments.

While the proposed framework demonstrates near-perfect performance and valuable interpretability, several limitations should be acknowledged. First, the SHAP analysis was performed on a reduced subset of the dataset due to computational constraints, which may limit the granularity of interpretability insights. Second, the model was evaluated only on the CICIDS 2017 dataset. Although this dataset is widely accepted in the community, cross-dataset validation on newer datasets such as CICDDoS2019 or UGR'16 is necessary to confirm robustness across diverse traffic conditions. Third, the model's computational requirements, particularly for real-time SHAP explanations, remain a challenge for large-scale deployment without adequate hardware acceleration. Future work should therefore consider optimizing SHAP computation or adopting lightweight explainability methods to ensure feasibility in real-time environments.

4. Conclusions

In this study, a hybrid CNN-LSTM deep learning framework was developed to detect DDoS attacks with both high accuracy and interpretability. Leveraging the CICIDS 2017 dataset and rigorous preprocessing, including class balancing with SMOTE, the proposed model achieved 99.98% accuracy and a 99.83% F1-score, demonstrating strong capability in distinguishing malicious from benign network traffic. Beyond predictive performance, the integration of SHAP addressed the critical gap of interpretability in deep learning-based intrusion detection by revealing the most influential features contributing to the model's decisions. The contributions of this work can be summarized in three aspects: achieving near-perfect detection accuracy through an optimized hybrid CNN-LSTM architecture, enhancing transparency and trustworthiness via explainable model predictions using SHAP, and designing a deployment-oriented framework that balances predictive strength with interpretability for practical cybersecurity applications. Importantly, the exceedingly low false positive rate ensures that legitimate traffic is not disrupted, making the framework suitable for real-time network defense systems where both reliability and service continuity are critical. These findings affirm that properly tuned and interpreted deep learning models can serve as powerful and trustworthy tools for detecting DDoS attacks in operational environments. Future studies should emphasize adaptive explainability mechanisms and online learning to support real-time intrusion detection.

Building upon these results, this study further envisions several practical deployment scenarios. The proposed CNN-LSTM framework can be integrated into existing Intrusion Detection and Prevention Systems (IDS/IPS) such as Snort or Suricata, where the explainable predictions provided by SHAP can enhance analyst decision-making. Furthermore, the architecture shows promise for cloud-based and Software Defined Networking (SDN) environments, where scalability and adaptive learning are critical. However, successful real-world implementation requires careful consideration of inference latency, resource consumption, and system scalability, particularly in high-speed networks. Future

research will therefore focus on developing lightweight variants of the framework, exploring online learning strategies, and extending interpretability mechanisms for large-scale, real-time cybersecurity operations.

References

[1] M. Revathi, V. V. Ramalingam, and B. Amutha, "A Machine Learning Based Detection and Mitigation of the DDOS Attack by Using SDN Controller Framework," *Wirel Pers Commun*, vol. 127, no. 3, pp. 2417–2441, Dec. 2022, doi: 10.1007/s11277-021-09071-1.

[2] A. Suhag and A. Daniel, "Study of statistical techniques and artificial intelligence methods in distributed denial of service (DDOS) assault and defense," *Journal of Cyber Security Technology*, vol. 7, no. 1, pp. 21–51, Jan. 2023, doi: 10.1080/23742917.2022.2135856.

[3] Y. Shang, "Prevention and detection of DDOS attack in virtual cloud computing environment using Naive Bayes algorithm of machine learning," *Measurement: Sensors*, vol. 31, p. 100991, Feb. 2024, doi: 10.1016/j.measen.2023.100991.

[4] A. Fathima, G. S. Devi, and M. Faizaanuddin, "Improving distributed denial of service attack detection using supervised machine learning," *Measurement: Sensors*, vol. 30, p. 100911, Dec. 2023, doi: 10.1016/j.measen.2023.100911.

[5] E. Yang, S. Jeong, and C. Seo, "Harnessing feature pruning with optimal deep learning based DDoS cyberattack detection on IoT environment," *Sci Rep*, vol. 15, no. 1, p. 17516, May 2025, doi: 10.1038/s41598-025-02152-2.

[6] M. A. Ali and S. A. H. Al-Sharafi, "Intrusion detection in IoT networks using machine learning and deep learning approaches for MitM attack mitigation," *Discover Internet of Things*, vol. 5, no. 1, p. 48, Apr. 2025, doi: 10.1007/s43926-025-00104-w.

[7] B. Wang, Y. Jiang, Y. Liao, and Z. Li, "DDoS-MSCT: A DDoS Attack Detection Method Based on Multiscale Convolution and Transformer," *IET Inf Secur*, vol. 2024, no. 1, Jan. 2024, doi: 10.1049/2024/1056705.

[8] M. Al-Fayoumi and Q. Abu Al-Haija, "Capturing low-rate DDoS attack based on MQTT protocol in software Defined-IoT environment," *Array*, vol. 19, p. 100316, Sep. 2023, doi: 10.1016/j.array.2023.100316.

[9] A. M. Rifai, S. Raharjo, E. Utami, and D. Ariatmanto, "Analysis for diagnosis of pneumonia symptoms using chest X-ray based on MobileNetV2 models with image enhancement using white balance and contrast limited adaptive histogram equalization (CLAHE)," *Biomed Signal Process Control*, vol. 90, p. 105857, Apr. 2024, doi: 10.1016/j.bspc.2023.105857.

[10] S. Muthukumar and A. K. Ashfauk Ahamed, "A novel framework of DDoS attack detection in network using hybrid heuristic deep learning approaches with attention mechanism," *Journal of High Speed Networks*, vol. 30, no. 2, pp. 251–277, May 2024, doi: 10.3233/JHS-230142.

[11] Y. Yang and X. Peng, "BERT-based network for intrusion detection system," *EURASIP J Inf Secur*, vol. 2025, no. 1, p. 11, Mar. 2025, doi: 10.1186/s13635-025-00191-w.

[12] A. M. Rifa'i, A. T. Zy, W. Hadikristanto, Amali, and S. Butsianto, "Leveraging Data Augmentation and Dropout Layer in MobileNetV3 for Accurate Skin Cancer Detection ISIC Dataset," in *2024 8th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, IEEE, Aug. 2024, pp. 591–596. doi: 10.1109/ICITISEE63424.2024.10730105.

[13] Z. He, X. Wang, and C. Li, "A Time Series Intrusion Detection Method Based on SSAE, TCN and Bi-LSTM," *Computers, Materials & Continua*, vol. 78, no. 1, pp. 845–871, 2024, doi: 10.32604/cmc.2023.046607.

[14] R. Abu Bakar, L. De Marinis, F. Cugini, and F. Paolucci, "FTG-Net-E: A hierarchical ensemble graph neural network for DDoS attack detection," *Computer Networks*, vol. 250, p. 110508, Aug. 2024, doi: 10.1016/j.comnet.2024.110508.

[15] S. Verma and S. Prabakeran, "A Hybrid Deep Learning Approach to Network Traffic Anomaly Detection Enhanced by SHAP and LIME Interpretability," in *2025 8th International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE, Apr. 2025, pp. 1254–1261. doi: 10.1109/ICOEI65986.2025.11013553.

[16] Q. Dai, B. Zhang, and S. Dong, "A DDoS-Attack Detection Method Oriented to the Blockchain Network Layer," *Security and Communication Networks*, vol. 2022, pp. 1–18, May 2022, doi: 10.1155/2022/5692820.

[17] G. K. Yuvaraaj and K. Dhinakaran, "SHAP-Enhanced Hybrid Model for Accurate Financial Fraud Detection," in *2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, IEEE, Mar. 2025, pp. 1–5. doi: 10.1109/ICDSAAI65575.2025.11011708.

[18] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?,'" in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Aug. 2016, pp. 1135–1144. doi: 10.1145/2939672.2939778.

[19] J. Lee, S. Oh, J. Song, J. Noh, M. Hanh, and J. Kim, "Explainable Network Anomaly Detection with GraphSAGE and SHAP," in *2025 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, IEEE, Feb. 2025, pp. 0477–0482. doi: 10.1109/ICAIIC64266.2025.10920860.

[20] B. Asal, A. Cakin, and S. Dilek, "Enhancing Industrial IoT Cybersecurity with Explainable AI: A SHAP and LIME-Based Intrusion Detection Methodology," in *2025 7th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (ICHORA)*, IEEE, May 2025, pp. 1–8. doi: 10.1109/ICHORA65333.2025.11017105.

[21] S. Abiramasundari and V. Ramaswamy, "Distributed denial-of-service (DDOS) attack detection using supervised machine learning algorithms," *Sci Rep*, vol. 15, no. 1, p. 13098, Apr. 2025, doi: 10.1038/s41598-024-84879-y.

[22] Ahmad Turmudi Zy, Isarianto, A. M. Rifa'i, A. Nugroho, and A. Ghofir, "Enhancing Network Security: Evaluating SDN-Enabled Firewall Solutions and Clustering Analysis Using K-Means through Data-Driven Insights," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 9, no. 1, pp. 69–76, Jan. 2025, doi: 10.29207/resti.v9i1.6056.

[23] C. Zhu, E. A. Bamidele, X. Shen, G. Zhu, and B. Li, "Machine Learning Aided Design and Optimization of Thermal Metamaterials," *Chem Rev*, vol. 124, no. 7, pp. 4258–4331, Apr. 2024, doi: 10.1021/acs.chemrev.3c00708.

[24] A. M. Rifa'i, E. Utami, and D. Ariatmanto, "Analysis for Diagnosis of Pneumonia Symptoms Using Chest X-Ray Based on Resnet-50 Models With Different Epoch," in *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, IEEE, Dec. 2022, pp. 471–476. doi: 10.1109/ICITISEE57756.2022.10057805.

[25] A. S. Sunge, Amali, A. T. ZY, D. K. Pramudito, A. Badruzzaman, and Purwanto, "The model interpretability on SHAP and comparison classification selection feature for heart disease prediction," *Procedia Comput Sci*, vol. 245, pp. 210–219, 2024, doi: 10.1016/j.procs.2024.10.245.

[26] C. van Zyl, X. Ye, and R. Naidoo, "Harnessing eXplainable artificial intelligence for feature selection in time series energy forecasting: A comparative analysis of Grad-CAM and SHAP," *Appl Energy*, vol. 353, p. 122079, Jan. 2024, doi: 10.1016/j.apenergy.2023.122079.

[27] G. Ranjbaran, D. R. Recupero, C. K. Roy, and K. A. Schneider, "C-SHAP: A Hybrid Method for Fast and Efficient Interpretability," *Applied Sciences*, vol. 15, no. 2, p. 672, Jan. 2025, doi: 10.3390/app15020672.