A Hybrid Imbalanced DDoS Detection Framework Utilizing CNN, LSTM, and K-Means SMOTE

Rissal Efendi
Department of Information Technology, Satya Wacana Christian University, Salatiga, Indonesia
rissal.efendi@uksw.edu (corresponding author)

Indrastanti Ratna Widiasari
Department of Information Technology, Satya Wacana Christian University, Salatiga, Indonesia
indrastanti@uksw.edu

Erwien Christianto
Department of Information Technology, Satya Wacana Christian University, Salatiga, Indonesia
erwien.christianto@uksw.edu

Received: 12 December 2025 | Revised: 15 January 2026 | Accepted: 23 January 2026

 DOI: https://doi.org/10.48084/etasr.16901

ABSTRACT

Cyberattacks remain a highly disruptive threat to modern networks. However, the imbalanced nature of real-world network traffic, where attack data constitute only a small fraction, poses significant challenges for accurate detection. This study proposes a hybrid deep learning framework that combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models with a K-means Synthetic Minority Oversampling Technique (SMOTE) to address class imbalance in penetration testing data. A total of 1,532,029 network flow records were collected during penetration testing, comprising 1,230,487 benign flows (80.4%) and 301,542 malicious flows (19.6%), which represented Distributed Denial of Service (DDoS) attacks, including SYN floods, UDP floods, and ICMP floods. The CNN component extracts spatial features from network flows, while the LSTM captures their temporal dependencies. K-means SMOTE enhances detection by generating realistic synthetic samples for minority attack classes. The experimental results show that the CNN-LSTM model with K-means SMOTE achieves a DDoS detection recall of 94.59% and an F1-score of 89.45%, significantly outperforming the imbalanced baseline, with a recall of 64.35% and an F1-score of 73.05%, as well as other classifiers such as Support Vector Machine (SVM) and Random Forest (RF). These findings demonstrate the model's robustness and practicality in detecting minority-class attacks under real-world conditions.

Keywords-cyberattack detection; DDoS; CNN-LSTM; K-means SMOTE; imbalanced data

I. INTRODUCTION

Cybersecurity threats in the digital world are becoming more frequent, sophisticated, and damaging. Systems have been exposed to numerous vulnerabilities due to reliance on internet-connected services, smart devices, and cloud infrastructures. DDoS attacks are a significant cybersecurity threat, which depletes resources and disrupts user accessibility [1]. In addition to DDoS, cyberattacks such as brute force intrusions and port scanning constitute crucial threats to network infrastructures, as they can undermine system stability, compromise classified information, and interrupt critical services. Consequently, there is a need for accurate, adaptive, and intelligent cyberattack detection systems. In real-world intrusion detection, imbalanced network traffic continues to challenge the performance of AI-based cybersecurity tools [2, 3]. In penetration testing or actual network monitoring, most traffic is benign, while malicious events, such as DDoS or brute force attempts, represent a small minority, leading to skewed class distribution that causes models to demonstrate poorly on minority attack classes [4]. For instance, a typical penetration testing dataset might contain only 3–10% attack traffic, while the remainder is normal activity. This imbalance leads to biased learning, where models achieve high accuracy but fail to correctly detect rare yet significant attacks. The implications are important, as False Negatives (FNs) may allow undetected intrusions that threaten system integrity and service availability.

Hybrid approaches for DDoS detection have been explored by combining clustering-based feature engineering and machine learning classifiers. CNN-based approaches have also

attained high accuracy in DDoS detection [5], while LSTM models have handled temporal dependencies but underperformed in detecting minority classes [6, 7]. Classical machine learning approaches, such as SVM and RF have also shown limitations when trained on highly imbalanced datasets [8, 9]. Furthermore, most existing works rely on public datasets, such as CIC-IDS2017 and NSL-KDD, which may not adequately reflect the complexity of real-world penetration testing scenarios. Another approach involves Multi-Layer Perceptron models [10], which have demonstrated improved detection performance.

As summarized in Table I, most existing studies rely on public benchmark datasets, such as CIC-IDS2017 or NSL-KDD, which may not fully represent real penetration-testing traffic. Authors in [11] evaluated a CNN-based intrusion detection model on the CIC-IDS2017 dataset and reported performance degradation for minority attack classes, while authors in [12] employed the NSL-KDD dataset to assess a BiLSTM model with an attention mechanism, highlighting increased computational complexity. Several hybrid deep learning approaches have been evaluated on benchmark datasets. Authors in [13] applied a DCNN–BiLSTM model with the Synthetic Minority Oversampling Technique (SMOTE) on the CIC-IDS2017 dataset. Authors in [14] proposed a CBLOF-based feature engineering method combined with XGBoost and evaluated it on CIC-IDS2017 and CIC-IDS2018 datasets. Authors in [15] introduced a CNN–LSTM-based Intrusion Detection System (IDS), validated on the CIC-IDS2017 dataset using random oversampling, which may not fully reflect realistic penetration-testing scenarios. Authors in [16] proposed a CNN–BiLSTM model with an attention mechanism and evaluated it on the CIC-IDS2017 dataset, whereas authors in [17] investigated ANN and XGBoost models on imbalanced network traffic data; however, they did not incorporate deep sequential modeling. Although benchmark datasets, such as CIC-IDS2017, CIC-IDS2018, and NSL-KDD, enable standardized evaluation, they are largely collected in controlled environments and may not capture real-world traffic variability and severe class imbalance. To address these limitations, the present study uses penetration-testing traffic data to provide a more realistic and deployment-oriented evaluation of the proposed CNN–LSTM model.

TABLE I. COMPARISON OF RELATED DDoS DETECTION STUDIES

Study	Dataset	Method	Imbalance handling	Key limitations
[11]	CIC-IDS2017	CNN	Not specified	Performance degradation observed for minority attack classes due to class imbalance.
[12]	NSL-KDD	BiLSTM + Attention	Not specified	Increased computational complexity associated with attention-based recurrent models.
[13]	CIC-IDS2017	DCNN–BiLSTM	SMOTE	Synthetic oversampling may not fully capture inherent traffic distribution characteristics.
[14]	CIC-IDS2017, CIC-IDS2018	CBLOF + XGBoost	Not specified	Limited exploration of deep learning architectures and imbalance-aware strategies.
[15]	CIC-IDS2017	CNN–LSTM	Random oversampling	Synthetic balancing may not reflect realistic penetration-testing traffic patterns.
[16]	CIC-IDS2017	CNN–BiLSTM + Attention	Not specified	Model complexity may affect deployment in resource-constrained environments.
[17]	Imbalanced network traffic data	ANN, XGBoost	Not specified	Limited modeling of temporal traffic patterns and absence of deep sequential architectures.

To overcome these limitations, research has explored the synergy between deep learning architectures and data-level solutions. Among the most promising deep learning strategies is the integration of CNNs and LSTM [18]. CNNs excel at extracting local spatial features from structured data, while LSTMs are capable of capturing long-term temporal dependencies, which constitute a significant capability for modeling sequential patterns in network traffic data. However, deep learning alone cannot address the imbalance problem inherent in penetration testing datasets. This is where K-means SMOTE, a clustering-based synthetic oversampling technique, becomes essential. By constructing synthetic samples of minority classes based on K-means clustering, it produces a more realistic and representative distribution of attack data, enabling models to generalize better.

The novelty of this research lies in the integrated use of penetration testing data, clustering-based oversampling, and a CNN-LSTM architecture to build a high-performing and reliable cyberattack detection system. While many existing studies have explored CNN or LSTM in isolation, the present study combines them to improve classification in temporal network data [19, 20]. Moreover, the use of K-means SMOTE ensures that the synthetic data reflects realistic attack clusters rather than randomly interpolated samples, resulting in more robust learning [21]. This approach also reflects practical deployment scenarios where minority attack events are underrepresented but must be detected with high sensitivity. Compared to conventional models trained on publicly available datasets, the proposed approach offers improved adaptability and relevance by using penetration testing as the data source. Although CNN-, LSTM-, and attention-based intrusion detection models have shown promising results, most existing studies rely on public benchmark datasets, such as CIC-IDS2017 or NSL-KDD, which may not fully represent real penetration-testing traffic. Moreover, attention-based and Bi-LSTM architectures often introduce higher computational complexity, limiting their practicality for real-world deployment. In addition, conventional SMOTE-based balancing methods frequently ignore the intrinsic cluster structure of attack traffic. To address these limitations, this study proposes a CNN–LSTM framework integrated with K-means SMOTE, designed specifically for imbalanced penetration-testing data, enabling improved minority-class detection with practical deployment feasibility. The main contribution of this work lies in the integration of spatial-

temporal deep learning with clustering-based oversampling techniques tailored to real penetration test data, as follows:

- An end-to-end cyberattack detection framework was proposed that combines deep learning with data-level balancing techniques, specifically targeting DDoS attacks.
- Real-world penetration testing data were utilized rather than publicly available datasets, providing a more realistic and practical evaluation of intrusion detection performance.
- A hybrid CNN-LSTM architecture was adopted, where CNN extracts spatial features and LSTM models temporal dependencies in network traffic, optimizing the model for detecting sequential attack patterns.
- The K-means SMOTE method was employed to handle class imbalance in network traffic data and to ensure that synthetic samples better represent real attack distributions.
- The proposed model was validated and demonstrated improved performance in detecting minority class attacks under imbalanced conditions.

These contributions aim to improve the performance of cyberattack detection systems, particularly in handling imbalanced real-world network traffic.

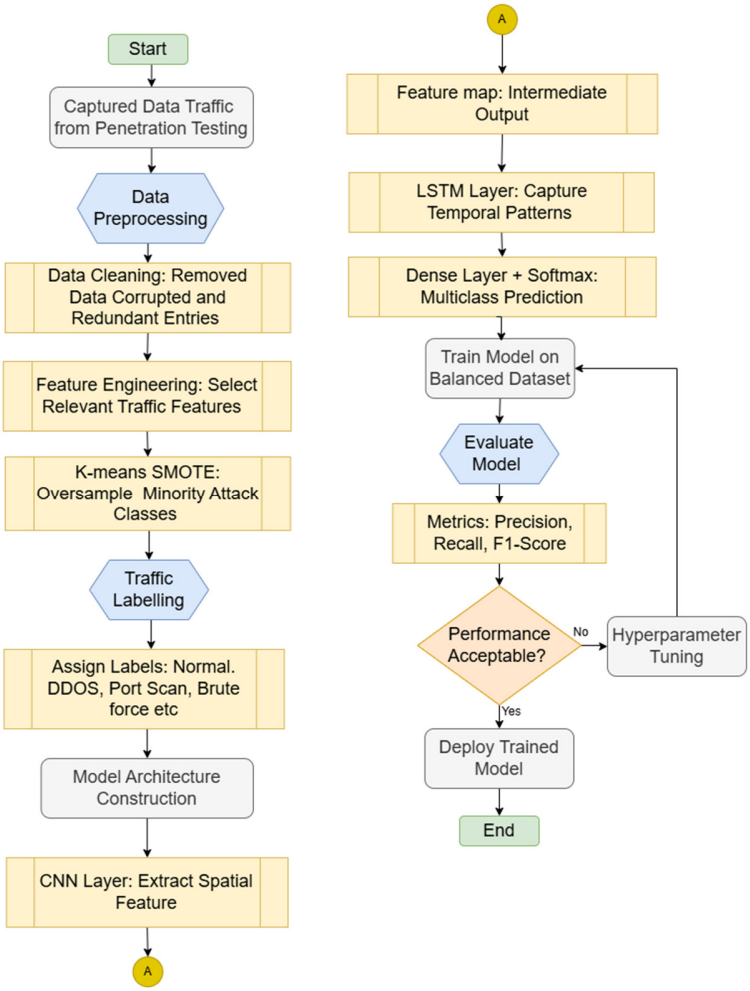

Fig. 1. Research methodology.

II. METHODOLOGY

The research methodology involves developing a cyberattack detection model on imbalanced network traffic data using a deep learning-based hybrid approach combined with SMOTE. Figure 1 illustrates the workflow of the proposed system. The initial steps involve the collection of network traffic data obtained from penetration testing activities. This is followed by data preprocessing, which includes removing corrupted and redundant records, as well as feature engineering to select relevant traffic attributes. To address class imbalance in the dataset, the K-means SMOTE method is employed to oversample minority attack classes. After balancing the dataset, traffic instances are divided into various categories, such as Normal and DDoS. The next step is the construction of the model architecture, which combines a 1D CNN for spatial feature extraction and LSTM layers.

The model is trained on a balanced dataset and evaluated using performance metrics such as precision, recall, and F1-score. If the evaluation results are not satisfactory, hyperparameter tuning is conducted. Once the desired performance level is achieved, the trained model is ready for deployment in a real-world cyberattack detection system.

A. Data Collection

This study focuses on penetration-testing traffic to better represent real operational attack scenarios, rather than relying on public benchmark datasets such as CIC-IDS2017 or NSL-KDD. Public benchmark datasets are mainly designed for standardized evaluation and may not fully capture the traffic characteristics generated during controlled penetration-testing activities. Data collection commenced with the ingestion of raw network traffic into the system, following a standard IDS data processing pipeline [2, 22, 23]. These traffic data were compiled into a dataset that underwent a series of preprocessing steps to prepare it for application in machine learning algorithms. Typically, network traffic encompasses numerous indicators and attributes, such as packet size, communication duration, and source/destination addresses.

Table II outlines the main features employed in this study to characterize network activity during a session. Each attribute contributes significantly to identifying patterns indicative of DDoS attacks, particularly when analyzing data at the packet level. Features such as packet count, total data volume in bytes, protocol type used, and connection status, which are crucial for assessing and identifying potential cybersecurity threats, are outlined.

TABLE II. PACKET FEATURE DETAILS

Variables	Description
pkts	Total number of packets. DDoS attacks typically involve unusually high packet volumes.
bytes	Total bytes in the flow. Large data transfers in a short period may signal volumetric attacks.
rate	Overall traffic rate. A sudden spike indicates potential flooding behavior.
srate	Source packet rate. High sending rates are typical in outbound attack traffic (e.g., from bots).
drate	Destination packet rate. High inbound traffic is a strong DDoS indicator on the target.
flgs_number	Encodes TCP flags like SYN. Useful for detecting SYN floods or abnormal handshake patterns.
proto_number	Protocol type (TCP, UDP, ICMP). Important for identifying protocol-specific DDoS attacks.
state_number	Connection state. Helps detect half-open or anomalous connection patterns common in DDoS.
dur	Duration of the flow. DDoS flows are often short but frequent.
TnBPSrcIP	Total bytes sent from a specific IP. Helps identify aggressive senders.
TnP_PSrcIP	Total packets sent from a specific IP. Useful for tracing packet-based flooding sources.

The key features extracted from network traffic data collected during a controlled penetration testing are presented. During this period, a total of 1,532,029 network flow records were captured, consisting of 1,230,487 benign flows (80.4%) and 301,542 malicious flows (19.6%), representing various simulated DDoS attack types. The dataset was collected through penetration testing in a controlled environment, without any personal, sensitive, or identifiable user data involved. The dataset was split into training and test sets using an 80/20 ratio. A subset of the training data was further used as a validation set to monitor the training process and prevent overfitting, while the test set was reserved exclusively for final performance evaluation. K-means SMOTE was applied only to the training data to avoid data leakage, and neither the validation nor the test data were oversampled. The model was implemented using TensorFlow and executed in a cloud-based environment with GPU acceleration.

B. Data Preprocessing

Before implementing machine learning models, the collected network traffic data underwent a rigorous preprocessing procedure to ensure data reliability and enhance model performance. Initially, records containing missing or corrupted values in key features, such as pkts (packet count), bytes (byte volume), rate (traffic rate), flgs_number (TCP flag indicators), and proto_number (protocol type), were identified and appropriately handled. Missing numerical values were imputed using median substitution to preserve the distribution of traffic metrics. Feature selection was conducted to retain variables most relevant to DDoS detection, including volume-based metrics (pkts, bytes, rate), protocol identifiers (proto_number), TCP flags (flgs_number), connection states (state_number), and traffic rates per source and destination IP (srate, drate, TnBPSrcIP, TnP_PSrcIP). Continuous features were normalized using min-max scaling to mitigate biases from differing magnitudes, ensuring equal contribution during model training. Categorical variables, such as protocol types and TCP flags, were encoded numerically via one-hot encoding to facilitate machine learning compatibility. Finally, each network flow was labelled as benign or malicious according to ground truth derived from the controlled penetration testing phase. This preprocessing workflow resulted in a clean, balanced, and structured dataset, optimized for accurate and efficient DDoS attack detection.

C. Handling Class Imbalance

In the context of DDoS attack detection, imbalanced data between normal and attack traffic are often a significant obstacle in training accurate models. When attack data are much less than normal data, the model often exhibits a bias towards the majority class, so a special strategy is needed to adjust the class distribution before training. Figure 2 illustrates the process of balancing imbalanced network data using a cluster-based approach and the SMOTE oversampling technique before being applied to a hybrid CNN-LSTM model to detect DDoS attacks. The process starts by inputting collected network traffic data that include both benign and malicious flows. The minority class (DDoS traffic) is identified based on key network flow features. These features include the number of packets (pkts), total bytes transferred (bytes), traffic rates (rate, srate, and drate), number of TCP flags (flgs_number), protocol type (proto_number), connection state (state_number), flow duration (dur), and totals of packets and bytes per source IP (TnP_PSrcIP and TnBPSrcIP). Next, K-means clustering is applied to divide the minority data into several clusters based on feature similarities [24]. For each

cluster, the system calculates the number of synthetic samples needed to balance the data distribution.

K-means clustering was performed on the minority class with $k = 5$ using Euclidean distance and k-means++ initialization. SMOTE was subsequently applied within each cluster using five nearest neighbors to generate synthetic samples.

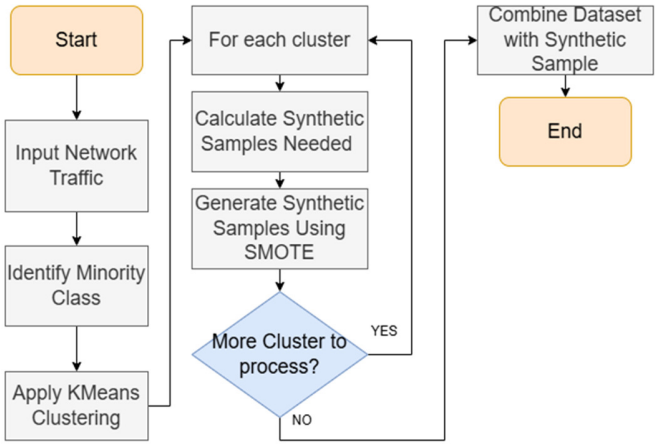

Fig. 2. Clustering-based SMOTE for network traffic balancing.

The distance between data points in the clustering process is given by (1), which minimizes the total squared distance between cluster members and their centroids:

$$arg\ min_s \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2 \quad (1)$$

where $S = \{S_1, S_2, \ldots, S_k\}$ represents the set of k cluster, $x \in S_t$ is a datapoint belonging to the i^{th} cluster, μ_i is the centroid of the cluster S_i, and $||x - \mu_i||^2$ denotes the squared Euclidean distance between the data point and its corresponding centroid. Once the number of synthetic samples is determined for each cluster, the SMOTE method is applied to generate new data adaptively through interpolation between minority class instances. This interpolation process is described by:

$$x_{new} = x_i + \left(x_{neighbor} - x_i\right) * \delta \quad (2)$$

where x_i is a minority data point, $x_{neighbor}$ is its nearest neighbor, and δ is a random value between 0 and 1. The resulting balanced dataset is then used to train the CNN-LSTM model.

D. Model Design: CNN-LSTM Hybrid Architecture

The hybrid architecture combining CNN and LSTM is specifically designed to handle the spatial-temporal characteristics of network traffic data, particularly for detecting cyberattacks such as DDoS. The integration of these two deep learning models leverages the strength of CNNs in feature extraction and the ability of LSTMs to learn temporal patterns, making it suitable for identifying complex attack behaviors within imbalanced penetration testing datasets. In this study, CNN is used as the initial component of the cyberattack detection architecture, with the main focus on identifying DDoS attacks. CNN is chosen because of its ability to extract spatial features from data, especially to recognize anomalous patterns that are often early indications of DDoS attacks. The input data are arranged in a three-dimensional matrix, namely (100,10,1), which represents 100-time steps, 10 network features (such as connection frequency, packet size, duration, and number of requests), and one input channel. This representation allows CNN to read the spatial relationship between features in a single time window. The feature extraction process begins with a Two-Dimensional Convolutional Layer (Conv2D) that uses 32 filters with a kernel size of 3 × 3 and same padding. This layer aims to capture local patterns in network traffic data, such as packet spikes from multiple sources at the same time. After convolutional layers extract spatial patterns, the features are flattened and passed into an LSTM layer that captures long-term temporal dependencies across traffic flows. Finally, dense layers and a Softmax classifier produce binary predictions between benign and DDoS traffic. The configuration of the CNN-LSTM model is summarized in Table III. The Adam optimizer was utilized for training, with categorical cross-entropy loss and a learning rate of 0.001. A batch size of 64, and up to 50 epochs were used, utilizing early stopping when no further improvement in validation loss was observed. The detailed hyperparameter configuration is summarized in Table IV. The convolution operation can be formulated as defined in:

$$S(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(i+m, j+n).K(m,n) + b \quad (3)$$

where X is the input data, K is the kernel or convolution filter, and b is the bias value. The result of the convolution is then processed using the Rectified Linear Unit (ReLU) activation function, which is mathematically expressed as:

$$f(x) = max\ (0, x) \quad (4)$$

TABLE III. CNN-LSTM MODEL ARCHITECTURE CONFIGURATION

Layer type	Parameters / units	Output shape	Description
Input layer	100 timesteps × 10 features × 1 channels	(100, 10, 1)	Input: sequence of network traffic features.
Conv2D-1	32 filters, kernel size (3×3), with ReLU activation	(98, 8, 32)	Extracts local spatial features from traffic data.
MaxPooling2D-1	Pool size (2×2)	(49, 4, 32)	Reduces spatial dimension, keeps dominant patterns.
Conv2D-2	64 filters, kernel size (3×3), with ReLU activation	(47, 2, 64)	Captures higher-level patterns of DDoS traffic.
MaxPooling2D-2	Pool size (2×2)	(23, 1, 64)	Further dimensionality reduction.
Flatten	-	1472	Converts feature maps into a 1D vector.
LSTM	128 units, dropout=0.2	128	Models temporal dependencies between flows.
Dense layer	64 units, with ReLU activation	64	Non-linear transformation before the final layer.
Output layer	2 units, with Softmax activation	2	Binary classification: benign vs DDoS.

TABLE IV. HYPERPARAMETER CONFIGURATION USED FOR TRAINING THE CNN-LSTM MODEL

Parameter	Value / setting	Description
Optimizer	Adam	Adaptive optimization algorithm for efficient gradient updates.
Loss function	Categorical Cross-Entropy	Suitable for binary/multi-class classification tasks.
Batch size	64	Number of samples per gradient update.
Epochs (max)	50	Maximum number of passes through the training set.
Early stopping	Patience = 5 (monitor = val_loss)	Halts the training process if validation loss fails to decrease.
Dropout	0.2	Applied to the LSTM layer to prevent overfitting.
Activation functions	ReLU (hidden layers), Softmax (output layer)	Applies non-linearity and yields output in the form of probability values.

This function removes negative values and retains positive values, so that the network can better learn non-linear relationships between features. Next, a max pooling operation is performed with a window size of 2×2, which reduces the spatial dimension while maintaining the dominant features of the convolution results. This operation is formulated as given in:

$$P(i,j) = max\, S(2i+m, 2j+n) \quad (5)$$

The convolution, activation, and pooling process are repeated one more time with more filter configurations, namely 64 filters, to extract more complex and advanced patterns. Such patterns are particularly relevant in the context of DDoS, where network traffic can exhibit high intensity spread from multiple source IP addresses. Once the feature extraction process is complete, the results are flattened, which converts the two-dimensional feature output into a one-dimensional vector. This operation can be defined as:

$$F = Flatten(P) \quad (6)$$

This feature vector is then passed to the LSTM layer to learn the temporal dynamics between time steps, which is a key element in identifying DDoS attacks that occur sequentially and systematically. CNN in this architecture acts as a spatial feature extractor, which is very important in capturing typical patterns of DDoS attacks, such as high frequency of connection requests in a short period of time, consistent packet sizes, and traffic spikes from many sources that resemble flooding patterns. Early detection of these patterns is crucial to prevent service disruptions and damage to network infrastructure.

The CNN output represents the spatial features of the network data. This vector is then input to the LSTM layer, which is tasked with learning the temporal patterns of these features to detect DDoS attacks. At each time step t, the LSTM unit receives an input $x_t = F_t$ and generates a hidden state h_t based on the information from the previous time step h_{t-1} and the cell memory C_{t-1}. The main mechanism of LSTM is described by:

1) Forget Gate

Equation (7) outlines the forget gate mechanism, which governs the transfer of information from the previous cell state h_{t-1}. The forget gate ensures that relevant information is kept by using the preceding hidden state h_{t-1} and the input h_{t-1}:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f \quad (7)$$

2) Input Gate

Equations (8) and (9) describe the process that alters the stored information in the memory units. The input gate determines the new information to be retained in the current cell state $\tilde{C}_t$, using the previous hidden state h_{t-1} and the current input x_t to add important information indicating the DDoS pattern:

$$i_t = \sigma(W_f\,[h_{t-1}, x_t] + b_f) \quad (8)$$

$$\tilde{C}_t = tanh(W_c[h_{t-1}, x_t] + b_{fc} \quad (9)$$

3) Memory Cell Status

Equation (10) ensures that the memory cell status C_t is updated by combining relevant old information and important new information, reflecting the current network traffic conditions in cyber detection:

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \quad (10)$$

4) Output Gate

Equations (11) and (12) define the output gate, which selects relevant information from the memory state of the cell C_t to be processed as output (h_t), which is then used to predict a DDoS attack:

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \quad (11)$$

$$h_t = o_t \times tanh(C_t) \quad (12)$$

Through this sequence of gating mechanisms, the LSTM layer is able to selectively retain and update memory based on patterns in network traffic over time.

E. Model Training and Evaluation

Model evaluation was carried out using a confusion matrix, which includes True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). This matrix helps assess how accurately the model classifies DDoS and normal traffic. Additionally, standard evaluation metrics, such as accuracy, were used to measure the overall success rate of detection. Precision quantifies the ratio of correctly detected DDoS attacks among all predicted positives, recall measures how many actual DDoS attacks were correctly identified, and the F1-score, which is an effective metric for evaluating models on imbalanced datasets, harmonizes precision and recall. All these metrics are calculated using:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (13)$$

$$\text{Precision (P)} = \frac{\text{TP}}{\text{FP+TP}} \quad (14)$$

$$\text{Recall (r)} = \frac{\text{TP}}{\text{FN+TP}} \quad (15)$$

$$\text{F}_{\text{score}} = 2 \times \frac{\text{P×R}}{\text{P+R}} \quad (16)$$

III. RESULTS AND DISCUSSION

A. Overview of Experimental Setup

The experiments were implemented using the dataset collected through penetration testing, which includes both benign traffic and various types of DDoS attacks. The primary goal of the experiments was to evaluate the effectiveness of the CNN-LSTM model under imbalanced and balanced data scenarios. Two experimental settings were used: the original imbalanced dataset and a resampled version of the same dataset employing K-means SMOTE for class balancing.

B. Feature Distribution Analysis

Figures 3-5 illustrate the distribution of all extracted features across the different traffic classes. These features include packet count, byte count, packet rate, source packet rate, destination packet rate, flag counts, protocol type, connection state, flow duration, source IP packet, and byte statistics. The visualizations reveal distinct patterns in these features between benign and DDoS attack flows. For example, DDoS traffic generally exhibits higher packet and byte counts, increased packet rates, and more flags set compared to benign traffic. Flow duration tends to be shorter in attack flows, reflecting the bursty nature of DDoS attacks.

Figure 3 displays the distribution of four traffic-related features: packet count (pkts), byte count (bytes), packet rate (rate), and source packet rate (srate) across different traffic classes. It is evident that DDoS attack flows tend to exhibit significantly higher packet and byte volumes compared to benign traffic. Similarly, both the overall packet rate and source packet rate are elevated in DDoS flows, reflecting the high-frequency nature of such attacks. These distinctions highlight the potential of specific features in separating normal from malicious behavior during network traffic analysis.

Figure 4 portrays the distribution of four traffic features, drate, flgs_number, proto_number, and state_number across different traffic types: Benign, DDoS_SYN, DDoS_UDP, and DDoS_ICMP. The drate graph shows that benign traffic tends to exhibit a wider range of data rates compared to DDoS traffic, with DDoS_UDP and DDoS_ICMP presenting more consistent and moderately elevated values. In the flgs_number plot, DDoS_UDP and DDoS_ICMP traffic demonstrate a greater number of unique packet flags compared to benign traffic, indicating more varied flag usage during attacks. The proto_number chart highlights that benign traffic involves a broader set of protocols, while DDoS traffic is restricted to a specific protocol, suggesting a focused protocol-based attack pattern. Meanwhile, the state_number plot indicates that benign traffic has fewer distinct connection states, while DDoS traffic, particularly SYN-based attacks, triggers more varied connection states. These patterns suggest that the observed features hold strong potential for distinguishing between normal and attack traffic.

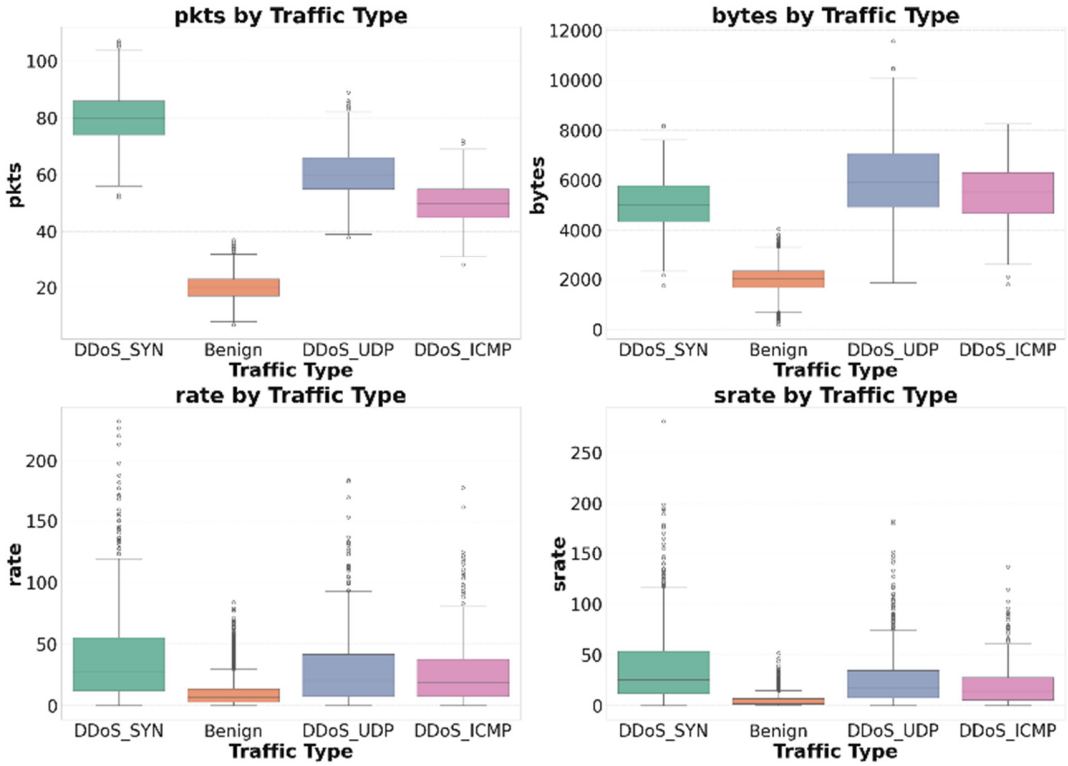

Fig. 3. Boxplot of packet count, byte count, packet rate, and source packet rate by traffic class.

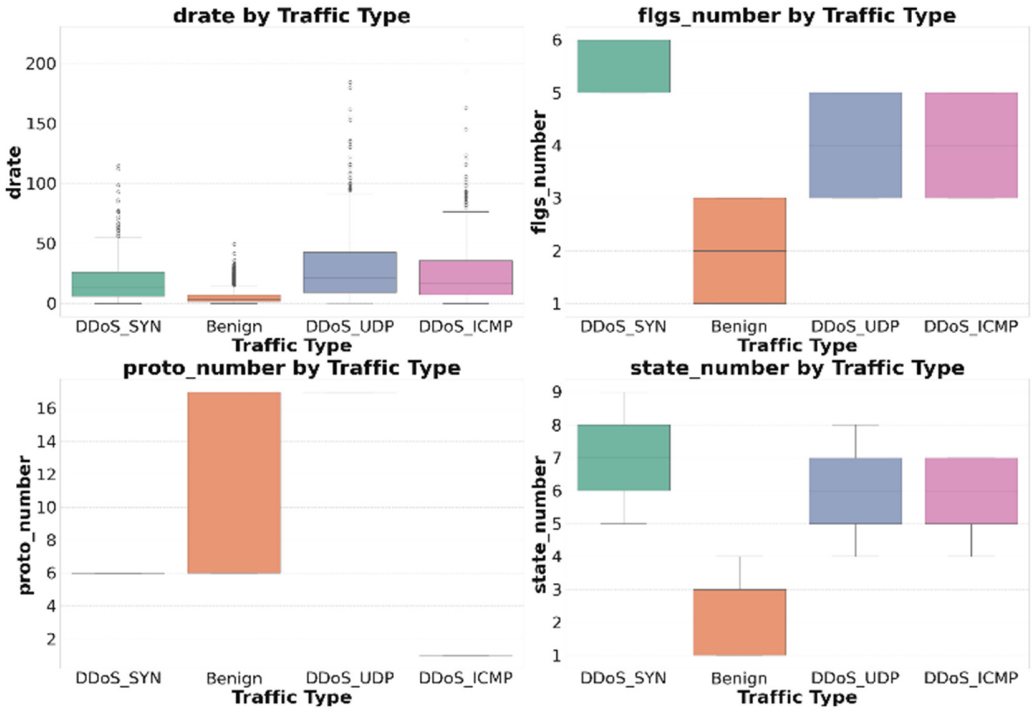

Fig. 4. Boxplot of destination packet rate, number of flags, protocol type, and connection state by traffic class.

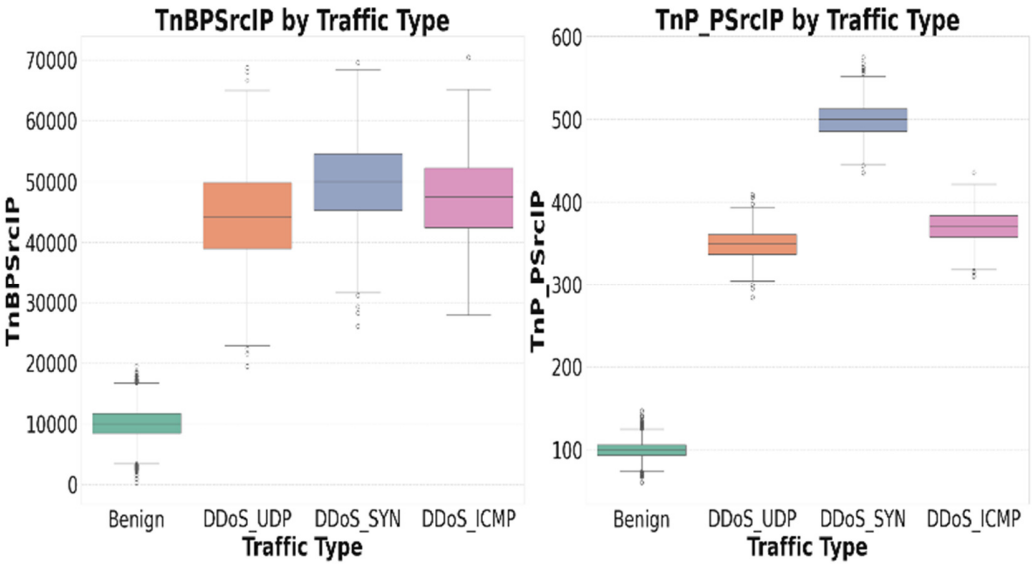

Fig. 5. Boxplot of distribution of TnBPSrcIP and TnP_PSrcIP by traffic type.

Figure 5 presents the boxplots of total bytes from source IP (TnBPSrcIP) and total packets from source IP (TnP_PSrcIP) across different traffic classes. The distributions indicate that DDoS traffic, particularly during high-volume attacks, typically originates from source IPs that generate substantially more packets and bytes compared to benign traffic. This behavior aligns with the nature of DDoS attacks, which often involve sending repetitive or voluminous traffic from specific source addresses to overwhelm targets. As such, both features serve as valuable indicators for identifying abnormal traffic patterns.

C. Performance Comparison: Imbalanced Versus Balanced Data

The performance CNN-LSTM model in DDoS attack detection was measured using two scenarios: training on the imbalanced data (original dataset), and training on the balanced data using K-means SMOTE. As displayed in Tables V and VI, the model trained on the imbalanced dataset exhibited high recall for benign traffic (97.0%) but significantly underperformed in detecting DDoS attacks, achieving only 64.3% recall and 69.5% F1-score. This highlights a great

limitation in handling minority-class intrusions. After applying K-means SMOTE to balance the class distribution, there is a substantial improvement in DDoS detection, as presented in Table V, with recall increasing to 94.6% and precision to 93.2%. The overall accuracy also improved to 96.2%, indicating enhanced generalization and more balanced detection performance across classes. Accuracy is a global metric for both classes combined. Therefore, the accuracy value is only presented once.

TABLE V. PERFORMANCE METRICS ON IMBALANCED DATA

Metric	Benign class	DDoS class
Accuracy	90.66%	-
Precision	97.75%	84.48%
Recall	97.10%	64.35%
F1-score	94.35%	73.05%

TABLE VI. PERFORMANCE METRICS ON BALANCED DATA USING K-MEANS SMOTE

Metric	Benign class	DDoS class
Accuracy	95.61%	-
Precision	98,64%	84.85%
Recall	95.80%	94.59%
F1-score	97.23%	89.45%

D. Confusion Matrix Analysis

To strengthen the performance evaluation, a confusion matrix is used to display the number of correct and incorrect classifications. The impact of class balancing on the model's classification performance is further illustrated in Tables VII and VIII, which present the confusion matrices before and after applying K-means SMOTE, respectively. In the imbalanced data, the model performs very well in classifying benign traffic with a TN of 1,194,826, but fails to detect around 107,495 DDoS attacks, meaning almost 35.6% of the total attacks are unrecognized. After balancing, the number of TP DDoS increases sharply to 285,226, and FN drops drastically to only 16,316. Although there is an increase in FPs (50,935 benign were classified as DDoS), this trade-off is reasonable in the network security domain, where false alarms are more acceptable than the actual attacks being missed.

TABLE VII. CONFUSION MATRIX ON IMBALANCED DATA

Actual / predicted	Predicted benign	Predicted DDoS
Actual benign (1,230,487)	1,194,826 (TN)	35,661 (FP)
Actual DDoS (301,542)	107,495 (FN)	194,047 (TP)

TABLE VIII. CONFUSION MATRIX ON BALANCED DATA USING K-MEANS SMOTE

Actual / predicted	Predicted benign	Predicted DDoS
Actual benign (1,230,487)	1,179,552 (TN)	50,935 (FP)
Actual DDoS (301,542)	16,316 (FN)	285,226 (TP)

E. ROC and Precision-Recall (PR) Curve Analysis

This study evaluates model performance using both the confusion matrix and the Area Under the ROC Curve (AUC). Although the ROC and PR curve visualizations are not shown, the AUC calculation is still carried out to ensure the reliability of the model. The results show that AUC-ROC increases from 87.1% (imbalanced) to 96.2% (balanced), reflecting the improvement of the model's ability to distinguish between benign and DDoS traffic. Similarly, the PR-AUC increases from 70.2% to 92.7%, indicating that data balancing significantly improves the sensitivity to DDoS detection, without significantly increasing FPs.

These metrics indicate that the hybrid CNN-LSTM approach with K-means SMOTE is more effective in detecting attacks compared to the model before balancing. Thus, the confusion matrix results support the conclusion that balancing with K-means SMOTE significantly improves the model's ability to detect DDoS, making it more reliable for use in real-world attack detection systems. Table IX provides a comparative evaluation of the CNN-LSTM model's performance before and after applying K-means SMOTE.

TABLE IX. MODEL PERFORMANCE BEFORE AND AFTER K-MEANS SMOTE

Model and dataset	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC (%)	PR-AUC (%)
CNN-LSTM (Imbalanced)	84.5	64.30	72.9	87.1	70.20
CNN-LSTM + SMOTE	84.80	94.60	89.45	96.20	92.70

The performance of the proposed CNN-LSTM model combined with K-means SMOTE balancing was then evaluated by comparing its performance/being compared with several baseline classifiers, such as SVM, RF, CNN-only, and LSTM-only models. All models were trained and tested on the balanced dataset to ensure a fair comparison. Figure 6 illustrates the performance improvement of the CNN-LSTM model after applying K-means SMOTE, with significant gains across all metrics. Table X presents a comparative analysis between the proposed CNN-LSTM model with K-means SMOTE and several baseline classifiers, demonstrating the superior performance of the proposed model across all evaluation metrics.

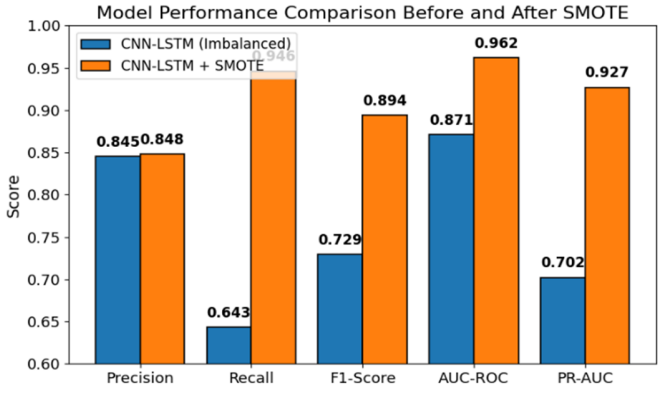

Fig. 6. Model performance before and after SMOTE.

TABLE X. COMPARISON OF BASELINE MODELS WITH THE PROPOSED MODEL

Model	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC (%)	PR-AUC (%)
SVM	75.00	68.00	71.00	82.00	69.00
(RF)	78.00	72.00	75.00	85.00	73.00
CNN-only	83.00	89.00	86.00	92.00	88.00
LSTM-only	80.00	87.00	83.00	89.00	85.00
Proposed model	85.00	94.60	89.45	96.20	92.70

The results demonstrate that the hybrid CNN-LSTM model outperforms traditional classifiers and single deep learning models, particularly in terms of recall and F1-score. This indicates its superior capability in accurately detecting DDoS attack traffic while maintaining a balanced precision level. The significantly higher AUC-ROC and PR-AUC values further confirm the model's robustness in distinguishing between benign and malicious traffic. Similarly, Figure 7 illustrates the comparative performance of several baseline models against the proposed CNN-LSTM with K-means SMOTE across five evaluation metrics. As observed, the proposed model consistently outperforms all other methods, particularly in recall and F1-score, which are critical for effective attack detection. This result confirms the advantage of combining spatial-temporal modeling with class balancing in handling imbalanced network traffic.

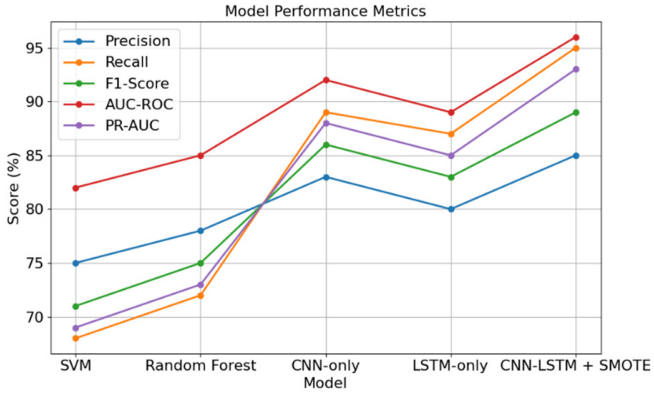

Fig. 7. Performance comparison between baseline and proposed models.

Although statistical significance testing was not conducted in this study, the evaluation was based on multiple complementary performance metrics, including precision, recall, F1-score, AUC-ROC, PR-AUC, and confusion matrix analysis. The absence of formal hypothesis testing is primarily due to computational constraints associated with large-scale deep learning experiments on high-volume network traffic data. This limitation is acknowledged, and the inclusion of statistical significance tests will be considered in future work to further validate performance differences between models.

F. Discussion

The experimental results demonstrate that the hybrid CNN-LSTM model combined with K-means SMOTE data balancing significantly outperforms traditional machine learning models, such as SVM and RF, as well as single deep learning models such as CNN-only and LSTM-only. The performance improvement is evident across all key metrics. The superior performance of the CNN-LSTM model can be attributed to its ability to extract both spatial features (via CNN layers) and temporal dependencies (via LSTM layers) from network flow data. This hybrid approach enables the model to better capture the complex patterns characteristic of DDoS attack traffic compared to models that rely solely on either convolutional or recurrent layers.

The use of K-means SMOTE for data balancing addresses the common issue of class imbalance in cybersecurity datasets, where benign traffic significantly outnumbers malicious flows. This balancing technique helps the model avoid bias toward the majority class, resulting in a significant increase in recall for the DDoS class (from 64.3% in the imbalanced dataset to 95.0% after balancing). The increase in recall is particularly important in security applications, where failing to detect attacks (FNs) can lead to severe consequences. While balancing the dataset leads to a slight increase in FPs (benign traffic misclassified as attacks), this trade-off is acceptable in network security contexts, where proactive detection and mitigation of potential threats are prioritized over occasional false alarms.

Compared to traditional models, such as SVM and RF, which achieved recall rates below 75%, the CNN-LSTM model's recall of 95% highlights the advantage of deep learning approaches in modelling intricate network traffic behaviors. Furthermore, the higher AUC-ROC and PR-AUC scores indicate better discrimination capability and robustness under different threshold settings. Table XI presents a comparative analysis of various intrusion detection models using standard evaluation metrics. Among the listed models, the CNN-BiLSTM-AT [16] achieved the highest performance, with an accuracy of 99.78% and an AUC-ROC of 100%, indicating excellent detection capabilities. Similarly, the Gradient Boosting (GB) model recorded a significantly high accuracy of 99.97%, reinforcing its effectiveness in classification tasks.

TABLE XI. PERFORMANCE COMPARISON OF EXISTING AND PROPOSED MODEL

Study	Method / Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC (%)
[13]	DCNN-BiLSTM Hybrid	94.78	94.78	94.77	99.78	–
[16]	CNN-BiLSTM + Attention	99.78	99.77	99.77	99.7	100
[17]	DBSCAN-SMOTE-CNN-XGBOOST-PSO	96.83	96.13	98.53	98.56	–
[25]	CNN-based IDS	99.9	96.15	–	97.3	–
This study	CNN-LSTM + K-means SMOTE	95.61	84.85 (DDoS) / 98.64 (Benign)	94.59 (DDoS) / 95.86 (Benign)	89.45 (DDoS) / 97.23 (Benign)	96.20 (DDOS)/97.5 (Benign)

Authors in [16] proposed a CNN-BiLSTM model enhanced with an attention mechanism, achieving near-perfect performance with 99.78% accuracy and 100% AUC-ROC. This indicates the effectiveness of combining convolutional and recurrent architectures for DDoS traffic detection. Authors in [25] reported a CNN-based IDS that achieved very high accuracy (99.9%) and strong precision (96.15%), although recall and AUC-ROC were not fully reported, leaving gaps in performance assessment. Authors in [13] introduced a DCNN-BiLSTM hybrid model, which achieved more balanced results with 94.78% precision and recall, and an F1-score of 99.78%, demonstrating robustness in handling diverse attack scenarios. Authors in [17] showed that the DBSCAN-SMOTE-CNN-XGBoost-PSO model has high accuracy and a strong F1-score overall, while the proposed model (CNN-LSTM + K-means SMOTE), although slightly lower in accuracy, excels in segmenting DDoS and benign traffic. The proposed model is able to maintain high precision on benign traffic while still detecting DDoS attacks effectively, thereby reducing FPs and providing more precise detection in unbalanced networks.

In comparison, the proposed model achieved 95.61% accuracy, with recall values of 94.59% for DDoS traffic and 95.86% for benign traffic. The balanced F1-scores (89.45% for DDoS and 97.23% for benign) highlight its ability to effectively detect both majority and minority classes. Overall, deep learning-based hybrid models generally outperformed conventional techniques, highlighting their suitability for advanced IDSs.

Research has relied on publicly available benchmark datasets, such as CIC-IDS2017, NSL-KDD, and CIC-IDS2018, for DDoS and intrusion detection tasks. These datasets provide standardized evaluation settings, facilitating comparative analysis across different methods. However, they are collected in controlled or simulated environments, which may not fully capture the traffic variability, attack sparsity, and class imbalance characteristics encountered in real-world penetration-testing scenarios. In contrast, the present study utilizes penetration-testing data to reflect realistic attack behaviors and naturally imbalanced traffic distributions. This choice enables a more practical assessment of the proposed CNN–LSTM model under conditions closer to real deployment environments. While benchmark datasets remain valuable for methodological comparison, the use of penetration-testing data enhances the external validity and robustness evaluation of the proposed approach.

IV. CONCLUSION

This study proposes a hybrid deep learning approach combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) with K-means Synthetic Minority Oversampling Technique (SMOTE) to improve cyberattack detection, particularly Distributed Denial of Service (DDoS) attacks, in imbalanced penetration testing data. The experimental results demonstrate that while the CNN-LSTM model performs well on benign traffic, its performance in DDoS detection significantly improves after applying K-means SMOTE, with the recall rising from 64.3% to 94.6% and the F1-score from 69.5% to 89.45%. Comparative analysis with baseline models, such as Support Vector Machine (SVM), Random Forest (RF), CNN-only, and LSTM-only, further confirms the superiority of the proposed method across all evaluation metrics, including Area Under the ROC Curve (AUC-ROC) and Precision-Recall (PR)-AUC. These findings highlight the effectiveness of integrating spatial-temporal deep learning with clustering-based oversampling techniques for enhancing intrusion detection in real-world, imbalanced network traffic scenarios.

ACKNOWLEDGMENT

This research was funded by the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia, Directorate General of Higher Education, Research, and Technology, under Contract No. 001/LL6/PL/AL.04/2025.

DATA AVAILABILITY STATEMENT

The dataset used in this study is available from the corresponding author upon reasonable request.

REFERENCES

[1] D. M. A. A. Afraji, J. Lloret, and L. Peñalver, "Deep Learning-Driven Defense Strategies for Mitigating DDoS Attacks in Cloud Computing Environments," *Cyber Security and Applications*, vol. 3, Dec. 2025, Art. no. 100085, https://doi.org/10.1016/j.csa.2025.100085.

[2] A. Abdelkhalek and M. Mashaly, "Addressing the Class Imbalance Problem in Network Intrusion Detection Systems Using Data Resampling and Deep Learning," *The Journal of Supercomputing*, vol. 79, no. 10, pp. 10611–10644, Jul. 2023, https://doi.org/10.1007/s11227-023-05073-x.

[3] M. A. Talukder *et al.*, "Machine Learning-Based Network Intrusion Detection for Big and Imbalanced Data Using Oversampling, Stacking Feature Embedding and Feature Extraction," *Journal of Big Data*, vol. 11, no. 1, Feb. 2024, Art. no. 33, https://doi.org/10.1186/s40537-024-00886-w.

[4] S. Aktar and A. Yasin Nur, "Towards DDoS Attack Detection Using Deep Learning Approach," *Computers & Security*, vol. 129, Jun. 2023, Art. no. 103251, https://doi.org/10.1016/j.cose.2023.103251.

[5] N. Mandela and F. Etyang, "Comparative Analysis of Deep Learning Models for Effective Denial of Service (DoS) Attack Detection in Network Security," *Journal of Electrical Systems and Information Technology*, vol. 12, no. 1, Sep. 2025, Art. no. 73, https://doi.org/10.1186/s43067-025-00267-0.

[6] M. Mbow, H. Koide, and K. Sakurai, "Handling Class Imbalance Problem in Intrusion Detection System Based on Deep Learning," *International Journal of Networking and Computing*, vol. 12, no. 2, pp. 467–492, 2022, https://doi.org/10.15803/ijnc.12.2_467.

[7] M. S. Milosevic and V. M. Ciric, "Extreme Minority Class Detection in Imbalanced Data for Network Intrusion," *Computers & Security*, vol. 123, Dec. 2022, Art. no. 102940, https://doi.org/10.1016/j.cose.2022.102940.

[8] R. Barona and E. Baburaj, "An Efficient DDoS Attack Detection and Categorization Using Adolescent Identity Search-Based Weighted SVM Model," *Peer-to-Peer Networking and Applications*, vol. 16, no. 2, pp. 1227–1241, Mar. 2023, https://doi.org/10.1007/s12083-023-01460-6.

[9] T. Wu, H. Fan, H. Zhu, C. You, H. Zhou, and X. Huang, "Intrusion Detection System Combined Enhanced Random Forest with SMOTE Algorithm," *EURASIP Journal on Advances in Signal Processing*, vol. 2022, no. 1, Dec. 2022, Art. no. 39, https://doi.org/10.1186/s13634-022-00871-6.

[10] M. S. Christo, J. J. Menandas, M. George, and S. V. Nuna, "DDoS Detection using Multilayer Perceptron," in *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, Jul. 2023, pp. 688–693, https://doi.org/10.1109/ICESC57686.2023.10193406.

[11] J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, "CNN-Based Network Intrusion Detection against Denial-of-Service Attacks," *Electronics*, vol. 9, no. 6, Jun. 2020, Art. no. 916, https://doi.org/10.3390/electronics9060916.

[12] Y. Yang, S. Tu, R. H. Ali, H. Alasmary, M. Waqas, and M. N. Amjad, "Intrusion Detection Based on Bidirectional Long Short-Term Memory with Attention Mechanism," *Computers, Materials & Continua*, vol. 74, no. 1, pp. 801–815, 2023, https://doi.org/10.32604/cmc.2023.031907.

[13] V. Hnamte and J. Hussain, "DCNNBiLSTM: An Efficient Hybrid Deep Learning-Based Intrusion Detection System," *Telematics and Informatics Reports*, vol. 10, Jun. 2023, Art. no. 100053, https://doi.org/10.1016/j.teler.2023.100053.

[14] Z. S. Dhahir, "A Hybrid Approach for Efficient DDoS Detection in Network Traffic Using CBLOF-Based Feature Engineering and XGBoost," *Journal of Future Artificial Intelligence and Technologies*, vol. 1, no. 2, pp. 174–190, Sep. 2024, https://doi.org/10.62411/faith.2024-33.

[15] S. S. Bamber, A. V. R. Katkuri, S. Sharma, and M. Angurala, "A Hybrid CNN-LSTM Approach for Intelligent Cyber Intrusion Detection System," *Computers & Security*, vol. 148, Jan. 2025, Art. no. 104146, https://doi.org/10.1016/j.cose.2024.104146.

[16] A. A. Najar and S. Manohar Naik, "DDoS Attack Detection Using CNN-BiLSTM with Attention Mechanism," *Telematics and Informatics Reports*, vol. 18, Jun. 2025, Art. no. 100211, https://doi.org/10.1016/j.teler.2025.100211.

[17] R. Efendi, "Optimizing Neural Network Architecture for Detecting DDOS Attacks Using ANN and XGBoost in Imbalanced Networks," *Engineering, Technology & Applied Science Research*, vol. 15, no. 3, pp. 22518–22526, Jun. 2025, https://doi.org/10.48084/etasr.9909.

[18] Y. Xue, C. Kang, and H. Yu, "A Network Intrusion Detection System Utilizing a Novel Autoencoder and a Hybrid Enhanced LSTM-CNN-Based Residual Network," *Computers & Security*, vol. 151, Apr. 2025, Art. no. 104328, https://doi.org/10.1016/j.cose.2025.104328.

[19] H. C. Altunay and Z. Albayrak, "A Hybrid CNN+LSTM-Based Intrusion Detection System for Industrial IoT Networks," *Engineering Science and Technology, an International Journal*, vol. 38, Feb. 2023, Art. no. 101322, https://doi.org/10.1016/j.jestch.2022.101322.

[20] M. Abdallah, N. An Le Khac, H. Jahromi, and A. Delia Jurcut, "A Hybrid CNN-LSTM Based Approach for Anomaly Detection Systems in SDNs," in *Proceedings of the 16th International Conference on Availability, Reliability and Security*, Vienna, Austria, Aug. 2021, pp. 1–7, https://doi.org/10.1145/3465481.3469190.

[21] A. O. Widodo, B. Setiawan, and R. Indraswari, "Machine Learning-Based Intrusion Detection on Multi-Class Imbalanced Dataset Using SMOTE," *Procedia Computer Science*, vol. 234, pp. 578–583, 2024, https://doi.org/10.1016/j.procs.2024.03.042.

[22] A. Hozouri, A. Mirzaei, and M. Effatparvar, "A Comprehensive Survey on Intrusion Detection Systems with Advances in Machine Learning, Deep Learning and Emerging Cybersecurity Challenges," *Discover Artificial Intelligence*, vol. 5, no. 1, Nov. 2025, Art. no. 314, https://doi.org/10.1007/s44163-025-00578-1.

[23] J. C. Mondragon, P. Branco, G.-V. Jourdan, A. E. Gutierrez-Rodriguez, and R. R. Biswal, "Advanced IDS: A Comparative Study of Datasets and Machine Learning Algorithms for Network Flow-Based Intrusion Detection Systems," *Applied Intelligence*, vol. 55, no. 7, May 2025, Art. no. 608, https://doi.org/10.1007/s10489-025-06422-4.

[24] Y. Yang, H. Akbarzadeh Khorshidi, and U. Aickelin, "A Diversity-Based Synthetic Oversampling Using Clustering for Handling Extreme Imbalance," *SN Computer Science*, vol. 4, no. 6, Nov. 2023, Art. no. 848, https://doi.org/10.1007/s42979-023-02249-3.

[25] D. Akgun, S. Hizal, and U. Cavusoglu, "A New DDoS Attacks Intrusion Detection Model Based on Deep Learning for Cybersecurity," *Computers & Security*, vol. 118, Jul. 2022, Art. no. 102748, https://doi.org/10.1016/j.cose.2022.102748.

AUTHORS PROFILE

Rissal Efendi earned his Master of Information Systems from Diponegoro University. He has been working as a lecturer in Pedagogy on Informatics Engineering and Computer Science Study Program, Satya Wacana Christian University since 2018. He has worked as a computer network engineer in the private sector. His research interests include computer networks, cybersecurity, machine learning, deep learning, and pervasive computing. He is currently focused on conducting IoT-based flood disaster prediction research in Semarang, Central Java, Indonesia.

Indrastanti Ratna Widiasari earned her Doctoral degree in the Electrical Engineering study program at Universitas Gadjah Mada, Yogyakarta, in 2018. She is a lecturer in the Department of Informatics Engineering, Satya Wacana Christian University. Her research interests include computer networks, wireless sensor networks, smart technology, deep learning, and pervasive computing.

Erwien Christianto earned his Master's Degree in computer science at Satya Wacana Christian University. He currently serves as a lecturer in the Diploma Program of Informatics Engineering at Satya Wacana Christian University. His research interests include computer science, information technology, information systems, and networking.