International Journal of Technology

http://ijtech.eng.ui.ac.id

Research Article

Artificial Intelligence-Enhanced Intrusion Detection Models for Next-Generation Cyber Threats

Mohammed Altaf Ahmed[1*], **Sultan Alqahtani**[1]

[1]Department of Computer Engineering, College of Computer Engineering & Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia

[*]Corresponding author: m.altaf@psau.edu.sa; Tel.: +966-5888347; Fax: +966-5888300

Abstract: Due to the fast proliferation of networked systems, distributed denial of service (DDoS) attacks have become increasingly common and more complex, and they are a major menace to network availability and reliability. Conventional intrusion detection systems (IDS) are prone to high-dimensional traffic data redundancy and a limited ability to learn higher-level temporal associations, thereby leading to lower detection and higher false alarm rates. To overcome these drawbacks, an intelligent intrusion detection system that combines ant colony optimization (ACO)-based feature selection and a hybrid convolutional neural network and bidirectional long short-term memory (CNN-BiLSTM) framework is proposed. Network traffic data are gathered using the CICIDS2017 dataset. An overall data preprocessing pipeline is used, which consists of missing value elimination, duplicate removal, binary label encoding, and min–max normalization to ensure data quality and stability. Flow-based statistical analysis and time-window traffic aggregation are performed to engineer discriminative traffic characteristics. ACO is used to create an ideal feature subset by maximizing the detection rate and minimizing feature redundancy. These features are then input to the CNN-BiLSTM model, whose CNN layers learn local spatial features and BiLSTM layers learn long-range temporal features. The proposed model has an accuracy of 98.65%, a precision of 97.37%, a recall of 95.00%, and an F1-score of 98.65%, which is better than current deep learning-based IDS approaches. This work provides an efficient, scalable, and high-performance IDS structure for real-world network security applications.

Keywords: Ant colony optimization; CNN–BiLSTM; Cybersecurity; DoS Attack; Intrusion detection system

1. Introduction

The rapid evolution of cyber threats has intensified the need for effective network security systems capable of detecting malicious activity (Biswas et al., 2024; Kim & Kim, 2023; Safitra et al., 2023). IDS serve as a critical defense mechanism by monitoring traffic patterns to distinguish between normal and abnormal behaviors (Ahmed & Alnatheer, 2025; Moustafa et al., 2023). However, in dynamic environments, traditional signature-based IDS methods struggle to identify advanced attacks such as DDoS (Sajid et al., 2024). Artificial intelligence (AI) and deep learning (DL) have emerged as promising solutions due to their ability to model complex traffic patterns, achieving strong performance on large datasets like CICIDS2017 (Vishnu & Soumya, 2024). Despite these advances, high-dimensional data handling, redundancy, and temporal dependencies remain challenges, which highlight the importance of optimization and hybrid DL approaches for scalable intrusion detection (Zhang & Yin, 2025).

Delayed detection of DoS attacks can severely disrupt modern digital systems. In enterprise and cloud environments, prolonged service outages cause downtime, financial loss, and reduced quality of service (Owusu et al., 2026). In critical sectors such as healthcare, banking, and online commerce, late responses compromise reliability, availability, and user safety. Repeated

https://doi.org/10.14716/ijtech.v17i4.8467
Received March 2026; Revised April 2026; Accepted June 2026

interruptions also erode customer confidence and trust in online platforms (Reed et al., 2025). These risks underscore the need for intrusion detection systems that identify DoS attacks quickly, accurately, and with minimal computational cost in dynamic network conditions.

Earlier studies in deep learning-based IDS have examined various architectures, including CNN, LSTM, and autoencoders, as well as transformer hybrids, to identify the spatial and temporal features of network traffic (Al-Na'amneh et al., 2024). An example is the use of deep CNN-BiLSTM-Transformer models to produce both local and sequential features, achieving high performance at a very high cost of computational complexity and classification inefficiency (Zhang et al., 2025). Autoencoder-LSTM hybrids were also effective at enhancing the representation of features, but they had higher training costs and were inefficient with feature redundancy (Narmadha & Balaji, 2025). Swarm-intelligence-inspired optimization of feature selection, including ACO, was used to reduce dimensionality, although many of these approaches use traditional machine learning classifiers as opposed to end-to-end DL (Kavitha & Lalitha, 2024). Transformer-augmented models have high training resource demands, and they frequently fail in streaming applications where they must minimize computational latency (Zhou et al., 2025). Particle swarm optimization and genetic algorithms are among the metaheuristic feature selection techniques explored to optimize feature subsets but are rarely used together with DL classifiers in a unified system to support modern IDS applications (Kunhare et al., 2022). Therefore, a hybrid approach that effectively merges optimization of feature selection and DL classification is still necessary to address these shortcomings (Aljabri et al., 2025).

To address existing drawbacks, the proposed framework integrates ACO-based feature selection with a hybrid CNN-BiLSTM classifier to detect DoS attacks using CICIDS2017, UNSW-NB15, and CSE-CIC-IDS2018 datasets. Unlike studies that apply optimization or deep learning separately, this design unifies both into a single end-to-end pipeline. ACO ensures efficient feature representation, while CNN-BiLSTM captures spatial and bidirectional temporal dependencies, enabling reliable detection of evolving attack behaviors. Overall, the unified approach enhances the robustness, scalability, and real-world applicability of intrusion detection systems.

The main contributions of this work are as follows.

- Improve detection performance using an optimized feature selection mechanism based on ACO that effectively eliminates redundant and irrelevant features.
- Combine a hybrid CNN-BiLSTM DL architecture that learns spatial patterns and bidirectional temporal network traffic dependencies.
- Illustrate higher binary classification accuracy and lower false alarms under benign and DoS attack classes using CICIDS2017, UNSW-NB15, and CSE-CIC-IDS2018 benchmark intrusion detection datasets.
- Compare the suggested approach with standard measures (accuracy, precision, recall, F1-score, ROC-AUC) to demonstrate its stability and capability for practical implementation.

1.1 Problem Statement

Existing IDS models exhibit specific limitations that hinder their effectiveness. Cloud-based systems using hybrid feature selection with machine learning classifiers improve accuracy but lack scalability and adaptability in fast-changing environments (Bakro et al., 2023). IoT intrusion detection models trained on datasets such as NSL-KDD can detect DoS attacks but suffer from imbalanced data, high false alarm rates, and poor generalization to real traffic (Esmaeili et al., 2022). Deep learning-based IDS frameworks enhanced for adversarial robustness provide resilience against manipulated inputs but introduce computational overhead and reduce efficiency (Yuan et al., 2024). These shortcomings demonstrate the need for approaches that balance accuracy, robustness, and efficiency.

1.2 Structure of the Paper

Section 1 provides the background and rationale for intrusion detection in contemporary networks. Section 2 provides a review of related literature regarding machine learning-based and deep learning-based IDS methods. Section 3 presents the proposed methodology, which comprises feature engineering, ACO-based feature selection, and the CNN-BiLSTM architecture. Section 4 describes the experimental setup and evaluation measures. The discussion is given in Section 5. Finally, Section 6 concludes the paper and indicates future research directions.

2. Literature Survey

2.1 DL Models for Intrusion Detection

Udurume et al. (2024) explored CNN-BiLSTM and machine learning classifiers applied to network intrusion detection and found that hybrid DL methods outperformed conventional methods on benchmark datasets such as NSL-KDD and UNSW-NB15. Better recall for attack traffic was reported with bidirectional temporal learning, and deep model feature hierarchies were more discriminative than handcrafted features, although higher sensitivity to hyperparameter optimization was observed. The edge-computing solution presented by Zhong et al. (2022) is a BiLSTM-based DDoS detection approach built on traffic sequential dependencies. This study demonstrated that BiLSTM was effective in capturing the long-term temporal variations of flooding attacks and reported better detection accuracy than unidirectional LSTM models, at the cost of increased inference latency on edge nodes and poor performance under highly imbalanced traffic distributions.

Susilo et al. (2025) proposed hybrid DL architectures in which CNN and LSTM are jointly used for intrusion detection. Their findings showed improved accuracy and noise resistance compared with standalone deep models, with spatial-temporal feature learning identified as a key strength; however, the added model complexity, larger memory usage, need for large labeled datasets, and poor real-time feasibility due to computational cost were noted as limitations. Ataa et al. (2024) studied deep learning-supported IDS systems and demonstrated that deep models were considerably more effective than classical machine learning models in complex attack scenarios, although the lack of built-in feature optimization affected detection stability.

2.2 Optimization and Metaheuristic Feature Selection

Vanitha and Balasubramanie (2022) used an improved ACO to select features for IoT-based intrusion detection, reducing the number of features and computational cost while increasing classification accuracy; however, parameter tuning was important for convergence speed, and the evaluation relied mainly on traditional classifiers rather than DL models. Mohsenabad and Tut (2024) compared different metaheuristic algorithms, including ACO, PSO, and GA, for IDS feature selection, but reported additional computational expense, inconsistent stability across datasets, and limited compatibility with deep models.

Rajput et al. (2024) showed that bio-inspired algorithms such as ant colony, artificial bee colony, and flower pollination obtain compact feature representations, although these algorithms exhibited slow convergence in high-dimensional spaces, with initialization strategies strongly influencing performance. Gong et al. (2025) examined hybrid metaheuristic feature selection for intrusion detection and highlighted convergence instability in high-dimensional feature space, computational complexity when combined with deep models, and difficulty in balancing exploration and exploitation; most methods could not be applied in real time, underscoring the need for unified optimization–DL frameworks.

2.3 Hybrid Architectures and System Challenges

Strategies Deep learning-based IDS architectures described by Hozouri et al. (2025) and Espinoza-Zelaya and Moon (2022) highlighted the significance of spatiotemporal modeling for

contemporary cyber threats. Hybrid architectures were more accurate than single models, though at the cost of lower interpretability and high computational demands that impede use in resource-limited settings. Ali et al. (2024) examined autoencoder-recurrent hybrid IDS models, reporting enhanced identification of imbalanced datasets and improved discrimination of anomalies through feature reconstruction, but with significantly higher training time and poor generalization across datasets.

Hewapathirana (2025) summarized AE-LSTM-based intrusion detection models, reporting high accuracy on the CICIDS2017 and CSE-CIC-IDS2018 datasets and strong temporal feature learning, although performance degraded with unknown traffic patterns, feature redundancy was not adequately addressed, and scalability remained a concern. Neto et al. (2025) examined transformer-augmented IDS models and noted that training complexity was more severe, applicability was restricted by large memory requirements, and real-time detection remained a challenge.

Research Gap: Existing deep learning-based intrusion detection systems often rely on high-dimensional features, resulting in increased computational cost and reduced efficiency. Therefore, a unified framework integrating metaheuristic feature optimization with hybrid deep learning is needed for accurate and scalable DoS attack detection.

3. Methodology

The proposed methodology is designed to develop an effective and intelligent intrusion detection system that can effectively detect DoS attacks within network traffic. The architecture shown in Figure 1 is based on a chain of operations comprising data preprocessing, feature engineering, feature optimization, and DL classification. Every stage is structured to process high-dimensional traffic data and enhance detection accuracy at minimal computational cost. The framework combines ACO to select the most pertinent traffic features and the hybrid CNN-BiLSTM framework to classify intrusions. The optimized feature subset, combined with the CNN-BiLSTM architecture, efficiently learns and captures spatial traffic characteristics as well as directional temporal characteristics that lead to a robust and reliable intrusion detection system.

3.1 Collection of Network Traffic

Network traffic was collected using a flow-based approach, where packets with similar properties (IP, protocol, time interval) were aggregated into flows. This method reduces redundancy, preserves key behavioral features (intensity, duration, dynamics), and is efficient for detecting DoS attacks. Flow-level attributes provide structured numerical features that are well-suited for preprocessing, optimization, and DL classification.

3.2 Dataset

The proposed framework uses the CICIDS2017 dataset from Kaggle as the benchmark dataset for training and analysis due to its realism, feature set, and ubiquitous use in intrusion detection studies. CICIDS2017 was captured under a regulated testbed setup designed to simulate real-world network behavior and integrates both benign user activity and varied attack conditions. The dataset was divided using a stratified split, with 70% for training and 30% for testing, ensuring balanced representation of benign and DoS traffic across both sets.

Additionally, the proposed framework was validated on the UNSW-NB15 and CSE-CIC-IDS2018 benchmark datasets using the same preprocessing, ACO-based feature optimization, and CNN-BiLSTM training procedures. Evaluating multiple datasets demonstrates the model's adaptability to diverse traffic patterns and attack characteristics. Figure 1 illustrates the three-stage workflow of the proposed intrusion detection framework.

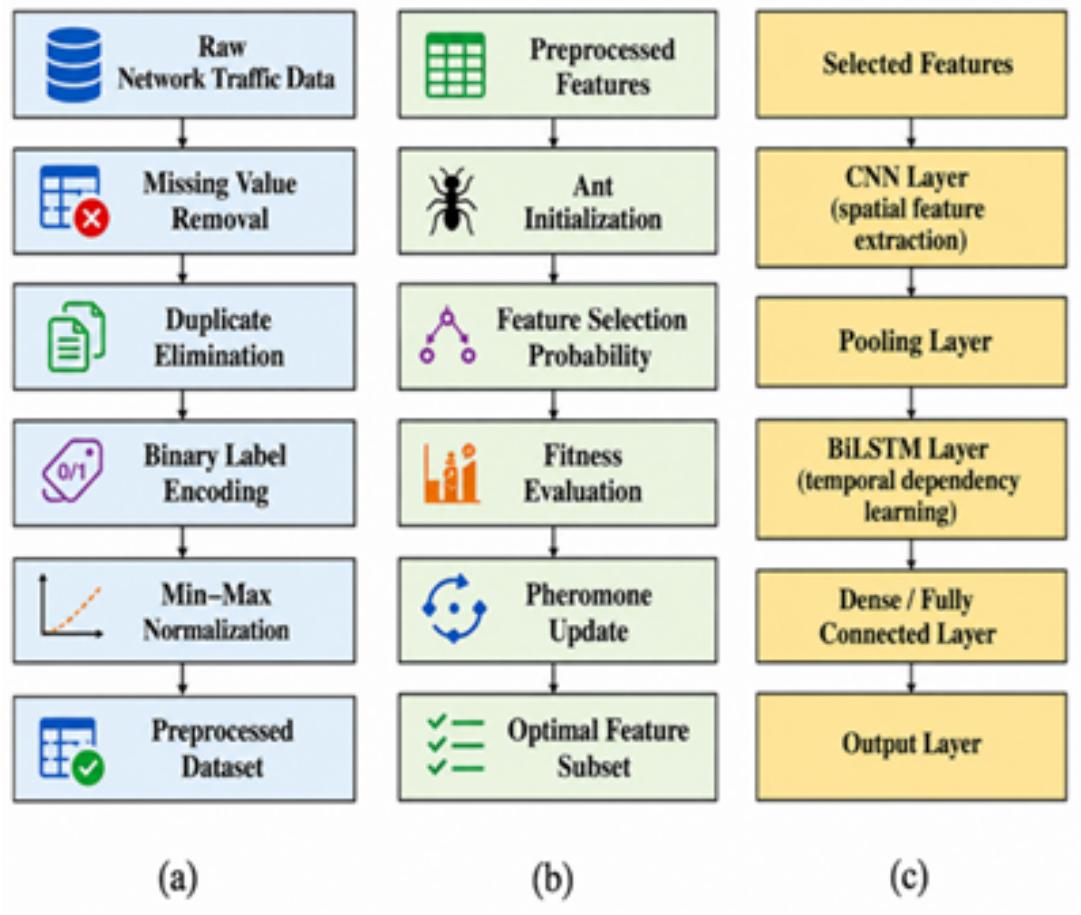

Figure 1 Proposed framework workflow: (a) data preprocessing, (b) ACO-based feature selection, and (c) CNN-BiLSTM classification architecture

3.3 Data Preprocessing

3.3.1 Removal of missing values

Missing values in network traffic may result from packet loss, flow timeout, or logging errors. Such dropouts introduce statistical bias in features and negatively influence the convergence and accuracy of DL models. Thus, missing-value handling is one of the key preprocessing steps to guarantee data integrity and model stability. Let the initial dataset be defined as in Equation 1:

$$\mathcal{D} = \{x_i\}_{i=1}^{N}, \quad x_i \in \mathbb{R}^F \tag{1}$$

where N denotes the total number of samples and F represents the number of features. If any feature x_{ij} in sample x_i satisfies

$$x_{ij} = \varnothing \text{ or } x_{ij} \in \{NaN, \infty\} \tag{2}$$

then the corresponding sample is considered incomplete, as expressed in Equation 2. In the proposed framework, row-wise deletion is implemented to exclude incomplete samples, and only fully observed traffic flows are kept, as given in Equation 3:

$$\mathcal{D}' = \{x_i \in \mathcal{D} \mid \forall j \in [1, F],\ x_{ij} \neq \varnothing\} \tag{3}$$

3.3.2 Duplicate elimination

Flow-based intrusion datasets often contain duplicate records because they are aggregated by packets, record flows in both directions, or contain retransmission artifacts. Let the dataset after missing-value removal be denoted as in Equation 4:

$$\mathcal{D}' = \{x_1, x_2, \ldots, x_M\} \tag{4}$$

Two samples in equation (5), x_i and x_j are considered duplicates if the following is true:

$$x_i = x_j \quad \forall i \neq j \tag{5}$$

Duplicate elimination is performed by applying a set-based uniqueness constraint, given in Equation 6:

$$\mathcal{D}'' = \text{Unique}\left(\mathcal{D}'\right) \tag{6}$$

Alternatively, using an indicator function, Equation 7 can be used:

$$\mathcal{D}'' = \left\{x_i \in \mathcal{D}' \mid \nexists\, x_j \in \mathcal{D}',\ j < i,\ \text{s.t. } x_i = x_j\right\} \tag{7}$$

This is done to ensure that all traffic flows contribute equally to the learning process, thus enhancing the model's generalization and minimizing computational redundancy during training course.

3.3.3 Binary label encoding (BLE)

Binary label encoding converts categorical class labels into numerical values that can be used by DL models. Raw labels in IDS datasets are usually textual, such as benign and DoS. Let the dataset after preprocessing be represented as in Equation 8:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N} \tag{8}$$

where $x_i \in \mathbb{R}^F$ denotes the feature vector of the i^{th} network flow and y_i represents its corresponding categorical label.

$$Y = \{y_1, y_2\} = \{\,\text{Benign}, \text{DoS}\,\} \tag{9}$$

Equation 9 defines the class label set Y, where the intrusion detection problem is formulated as a binary classification task consisting of two categories: benign (normal traffic) and DoS (attack traffic).

$$\hat{y}_i = \phi\left(y_i\right) = \begin{cases} 0, & \text{if } y_i = \text{Benign} \\ 1, & \text{if } y_i = \text{DoS} \end{cases} \tag{10}$$

Equation 10 represents the binary encoding function $\phi(\cdot)$, which maps categorical labels into numerical values. Here, each original label y_i is transformed into a binary label $\hat{y}_i$, where Benign traffic is encoded as 0 and DoS attack traffic is encoded as 1. This conversion enables the DL model to process categorical information in numerical form.

$$\mathcal{D}' = \{(x_i, \hat{y}_i)\}_{i=1}^{N}, \quad \hat{y}_i \in \{0, 1\} \tag{11}$$

Equation 11 defines the final encoded dataset $\mathcal{D}'$, where each network traffic sample is represented as a pair $(x_i, \hat{y}_i)$, with x_i denoting the feature vector of the i^{th} sample, $\hat{y}_i$ the corresponding binary-encoded label, and N the total number of samples in the dataset. Encoding labels to the set $\{0, 1\}$ ensures compatibility with binary classification using the CNN-BiLSTM model, while maintaining the semantic difference between normal and malicious traffic and allowing effective computation during training.

3.3.4 Min–max normalization

Normalization prevents the dominance of features with large magnitude, which would otherwise cause biased learning and slower convergence. Min–max normalization is used to bring all features to the same scale, as characterized by Equation 12:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{12}$$

where x is the original feature value, $x_{\min}$ and $x_{\max}$ denote the minimum and maximum values of the feature, respectively, and x' represents the normalized feature value.

3.3.5 Class imbalance handling

Network intrusion datasets often suffer from class imbalance, where one traffic category dominates others, potentially leading to biased model learning and reduced detection capability for minority classes. To address this issue, the proposed framework incorporates the Synthetic Minority Over-sampling Technique (SMOTE) to balance the class distribution by generating synthetic minority-class samples during training. This oversampling process improves the representation of underrepresented traffic classes and prevents the model from being biased toward dominant categories. In addition, class-weighted loss optimization was employed during CNN-BiLSTM training to assign higher importance to minority-class samples in the binary cross-entropy loss function, improving model sensitivity toward less frequent attack instances and reducing misclassification bias.

3.4 Feature Engineering

3.4.1 Flow-based statistical analysis

Flow-based statistical analysis calculates descriptive statistics that characterize network traffic behavior for each flow. These characteristics include flow duration, number of packets, number of bytes, average packet size, and variance-based measurements. Mathematically, the packet rate and byte rate features can be expressed as Equations 13 and 14:

$$\text{Packet Rate} = \frac{N_p}{T_f} \tag{13}$$

$$\text{Byte Rate} = \frac{N_b}{T_f} \tag{14}$$

where N_p is the number of packets in a flow, N_b is the total number of bytes, and T_f represents the flow duration. This enables the model to differentiate between normal and attack-based flooding traffic using compact but informative representations of traffic intensity.

3.4.2 Time-window traffic aggregation

Time-window traffic aggregation is used to aggregate temporal changes and is a flow-aggregation technique that uses fixed or sliding time windows to group flows. This allows the modeling of both short-term bursts and long-term traffic patterns, which are essential for identifying sustained DoS attacks. For feature selection, the ACO algorithm was configured with 20 ants over 50 iterations, with pheromone importance $\alpha = 1$, heuristic importance $\beta = 2$, and evaporation rate $\rho = 0.5$. The fitness function evaluates each feature subset using the validation accuracy of the CNN-BiLSTM model with a redundancy penalty to select informative, non-overlapping features. Statistical features, including mean, variance, and packet inter-arrival time, are computed within each time window, where the packet inter-arrival time is given by Equation 15:

$$IAT_i = t_i - t_{i-1} \tag{15}$$

where t_i and t_{i-1} denote arrival times of consecutive packets. Aggregated time characteristics enable the model to capture sequential dependency and burst pattern with which temporal DL layers, including BiLSTM, are fed.

3.5 ACO Feature Selection

The fundamental idea of ACO-based feature selection is based on the collective foraging behavior of ants, whereby a process referred to as pheromone reinforcement identifies optimum routes. In this model, features are represented as nodes, and artificial ants explore subsets of features with probabilities of traversal that depend on pheromone intensity and heuristic desirability. The selection probability is given in Equation 16:

$$P_{ij} = \frac{(\tau_{ij})^{\alpha} (\eta_{ij})^{\beta}}{\sum_k (\tau_{ik})^{\alpha} (\eta_{ik})^{\beta}} \tag{16}$$

where τ_{ij} is the pheromone concentration for feature j, η_{ij} is the heuristic value (e.g., feature relevance), and α and β control the influence of the pheromone and heuristic factors, respectively.

3.6 Hybrid CNN–BiLSTM DL Model

The proposed framework uses a hybrid CNN-BiLSTM architecture to successfully capture the intricate spatiotemporal nature of network traffic data. Network intrusion exhibits both local feature correlations (relationships between packet statistics within a flow) and long-range temporal features (attack persistence and time-dependent traffic modifications). Traditional single-model architectures often fail to capture both simultaneously, resulting in suboptimal detection performance. The proposed model combines CNN and BiLSTM into a single learning pipeline, harnessing the synergistic benefits of both to increase the expressiveness of learned features and classification accuracy, as shown in Figure 2.

Input Representation. The processed input traffic data are modeled as a structured feature matrix following preprocessing and ACO-based feature selection, as given in Equation 17:

$$X \in \mathbb{R}^{N \times F} \tag{17}$$

where N is the number of traffic samples and F is the number of optimized features selected by ACO. Each feature vector encapsulates statistical, temporal, and behavioral characteristics of network traffic. The reduced feature space eliminates redundancy and noise, enabling the DL model to concentrate on the most informative traffic features. Input vectors are rearranged into a two-dimensional form to allow convolution operations, allowing the CNN layers to learn local relationships between adjacent features while retaining spatial relations that remain compatible with sequential processing in the BiLSTM layers.

Convolutional Neural Network (CNN) Layer. The CNN module serves as the initial feature extractor in the hybrid architecture. It automatically learns task-specific spatial features using convolutional filters, capturing local patterns and correlations that distinguish normal from malicious network traffic. The convolution operation is expressed as Equation 18:

$$h_k = f(W_k * X + b_k) \tag{18}$$

where W_k represents the k^{th} convolution kernel, b_k is the bias term, X denotes the input feature matrix, and $f(\cdot)$ is a nonlinear activation function, typically a rectified linear unit (ReLU). ReLU activation increases nonlinearity and suppresses the vanishing gradient problem, improving the convergence of training. CNN layers perform particularly well in learning spatial relationships between flow-based features, including connections between packet rate, byte rate, and flow duration.

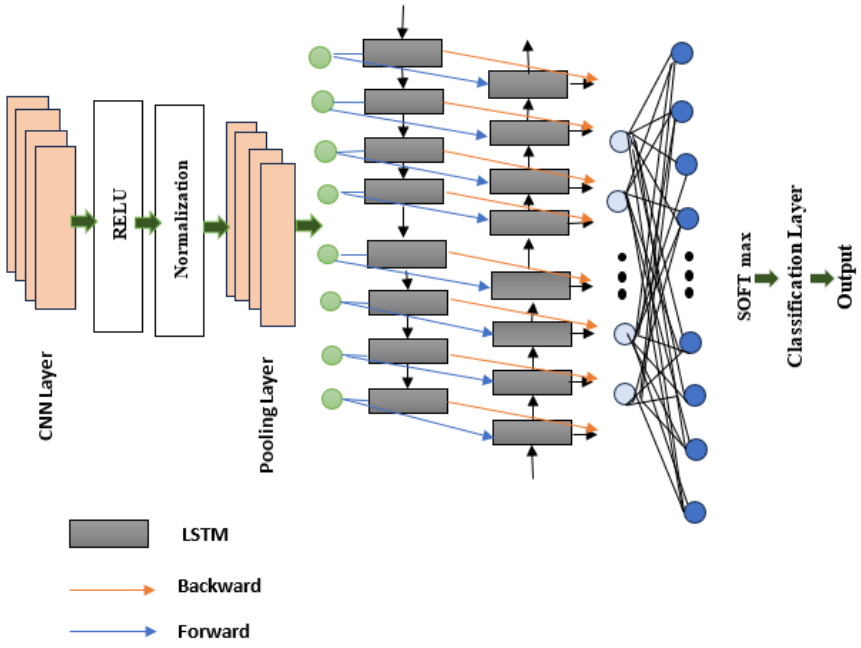

Figure 2 Architecture of the CNN-BiLSTM model

Pooling Layer. The pooling operation is used after the convolution layers to decrease the size of feature maps and increase the generalization. Summing up local responses and having the advantage of translational invariance, the model is less sensitive to small changes in traffic dynamics. Max-pooling picks up the highest activation value in each pooling window. The operation of pooling is given as Equation (19):

$$p_i = \max\left(h_{izi+m}\right) \tag{19}$$

where m represents the size of the pooling window and h_{izi+m} defines the collection of activations within the window. Pooling reduces computational complexity, noise, and overfitting while preserving salient features. The pooled feature maps are flattened and reshaped into sequences before being passed to the BiLSTM layer, enabling the model to learn temporal dependencies from the spatial features extracted by the CNN.

Bidirectional Long Short-Term Memory Layer. The BiLSTM module is used to capture long-term temporal relationships in network traffic data. Standard LSTM networks operate in only one temporal direction and therefore do not consider future context. By comparison, BiLSTM employs two parallel LSTM layers, one reading the sequence in the forward direction and the other in the backward direction. At every time step t, the forward and backward hidden states are estimated as in Equations 20 and 21:

$$\overrightarrow{h_t} = \mathrm{LSTM}_f\left(x_t, h_{t-1}\right) \tag{20}$$

$$\xi_t = \mathrm{LSTM}_b\left(x_t, h_{t+1}\right) \tag{21}$$

The final output is obtained by concatenating both states, as given in Equation 22:

$$h_t = \left[\overrightarrow{h_t} \oplus h_t\right] \tag{22}$$

BiLSTM captures both past and future dependencies, improving DoS attack detection. Its forget, input, and output gates preserve long-term dependencies and regulate information flow efficiently. The gate operations are defined in Equations 23-28:

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right) \tag{23}$$

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{24}$$

$$\tilde{c}_t = \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \tag{25}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{26}$$

$$e_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_v\right) \tag{27}$$

$$h_t = e_t \odot \tanh\left(c_t\right) \tag{28}$$

where $\sigma(\cdot)$ is the sigmoid activation function and $\odot$ denotes element-wise multiplication. These mechanisms enable BiLSTM to selectively retain or discard information over long sequences, thereby improving the detection of slow-rate or persistent attacks.

Fully Connected and Output Layer. The outputs of the BiLSTM layer are passed to a fully connected layer that aggregates temporal features and maps them to a lower-dimensional representation. This layer functions as the decision-making component that integrates the learned spatial and temporal features. For binary classification, a softmax or sigmoid activation function is applied at the output layer to compute class probabilities, as given by Equation 29:

$$P(y = k \mid X) = \frac{e^{z_k}}{\sum_{i=1}^{C} e^{z_i}} \tag{29}$$

The predicted class label is obtained by selecting the class with the highest probability. The model is trained using the binary cross-entropy loss function, given in Equation 30, which measures the discrepancy between the predicted probabilities and ground-truth labels:

3.7 Intrusion Detection

In the intrusion detection stage, binary classification is performed based on learned representations. The probability scores generated by the CNN-BiLSTM model are thresholded to produce distinct class labels. This stage enables decision-making, allowing for the rapid identification of DoS attacks and facilitating timely mitigation.

3.8 Performance Evaluation

Performance is evaluated using the standard metrics in Equations 31-34:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{30}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{31}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{32}$$

$$F1\ \text{Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{33}$$

Here, TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. ROC-AUC measures the overall discrimination capability of the model.

Algorithm 1 ACO-Optimized CNN-BiLSTM Intrusion Detection

Input: Network traffic feature set $F = \{f_1, f_2, \ldots, f_n\}$, maximum iterations T.
Output: Optimal feature subset.

BEGIN
Initialize pheromone levels for all features.
Set $iteration = 1$.
WHILE $iteration \leq T$ **DO**
  FOR each ant **DO**
    Initialize empty feature subset.
    FOR each feature in dataset **DO**
      Compute selection probability.
      IF random value < probability **THEN**
        Add feature to subset.
      ELSE
        Skip feature.
      END IF
    END FOR
    Evaluate subset using classification accuracy.
  END FOR
  Update pheromone:
    Reinforce features in best subset.
    Evaporate pheromone for other features.
  $iteration = iteration + 1$.
END WHILE
Select optimal feature subset.
FOR each traffic instance **DO**
  Extract selected features.
  Pass through CNN layers (Conv + ReLU + MaxPooling).
  Pass output to BiLSTM (forward + backward).
  Concatenate outputs and apply fully connected layer.
  Compute class probability using Softmax.
  IF probability(DoS) $\geq$ threshold **THEN**
    Label as DoS.
  ELSE
    Label as Benign.
  END IF
END FOR
RETURN classification labels.
END

4. Results and Discussion

4.1 Ablation Analysis

A comparative and ablation analysis was undertaken to clearly assess the role of each element of the proposed framework. Several current intrusion detectors have adopted CNN- or LSTM-based architectures individually and in basic hybrid modes. These models cannot simultaneously resolve feature redundancy and long-range temporal dependencies. To overcome this limitation, the proposed framework combines ACO-based feature selection with a hybrid CNN-BiLSTM architecture.

Ablation results in Table 1 show that CNN captures local spatial features but provides moderate detection performance. CNN–LSTM improves temporal modeling, while ACO–CNN enhances performance through feature optimization. The proposed ACO–CNN–BiLSTM achieves the highest accuracy by combining optimized feature selection with bidirectional temporal learning, resulting in improved generalization, lower false alarms, and superior intrusion detection performance.

Table 1 Ablation and comparative analysis of performance

No.	Model Variant	Accuracy (%)
1	CNN only	94.2
2	CNN + LSTM	96.1
3	CNN + BiLSTM	97.4

Computation Resources. All tests were run on a workstation with an Intel Core i7 processor, 16 GB of RAM, and an NVIDIA GPU with CUDA support to accelerate the computation of deep learning. The proposed model was developed in Python using the TensorFlow/Keras DL platform. Training time was greatly shortened, and high-dimensional traffic data could be efficiently handled through GPU acceleration. To achieve reproducibility and consistency in performance measurement, the training and evaluation processes were conducted in a controlled environment, and hyperparameters were optimized within this strategy to produce the best model.

Hyperparameter Tuning. The CNN component of the framework was constructed with two convolutional layers, the first using 32 filters and the second 64, both with 3×3 kernels, same padding, and ReLU activation. Each convolution block was followed by max-pooling with a pool size of 2 to reduce dimensionality. For temporal learning, the BiLSTM layer contained 64 hidden units in both forward and backward directions, enabling the capture of sequential dependencies from both sides of the traffic flow. A dropout rate of 0.3 was applied after BiLSTM to reduce overfitting, and a sigmoid activation was used in the final output layer to support binary classification.

Training employed the Adam optimizer with a learning rate of 0.001, a batch size of 64, and 30 epochs. Binary cross-entropy was selected as the loss function, and early stopping was applied to monitor validation loss. Hyperparameter sensitivity analysis was conducted by varying the CNN kernel size, BiLSTM units, learning rate, batch size, and ACO optimization settings. The chosen configuration balanced detection accuracy, computational efficiency, and convergence stability, demonstrating reproducibility under moderate parameter variations.

4.2 Performance Analysis of the Proposed Work

The model achieves a very high accuracy of 0.9865, indicating that it correctly classifies almost all instances. A precision of 0.9737 indicates that the model is most often correct when it predicts a positive outcome. The recall value of 0.9500 shows that the model successfully identifies the majority of actual positive cases, and the F1-score of 0.9865 confirms a strong balance between precision and recall, reflecting overall robust and consistent performance (Table 2).

Table 2 Summary of performance metrics

Accuracy	Precision	Recall	F1-Score
0.9865	0.9737	0.9500	0.9865

Figure 3 summarizes the performance of the proposed CNN-BiLSTM intrusion detection model. The ROC curve (a) shows near-perfect separation of benign and DoS traffic with an AUC of 0.9995, while the precision–recall curve (b) maintains almost flawless precision across recall values with an AP of 0.9995, confirming reliability under class imbalance. The error rates (c) are very low, with an FPR of 0.03221 and an FNR of 0.00520, highlighting strong detection capability and minimal missed attacks. The accuracy curves (d) converge quickly and stabilize around 98.65%, with training and validation closely aligned, indicating little overfitting. The loss curves (e) drop early and stabilize between 0.4 and 0.5, reflecting stable optimization and robust learning. Finally, the confusion matrix (f) shows 1923 true negatives, only 64 false positives, and nearly all 2000 DoS samples correctly identified, confirming high detection accuracy and reliability in reducing false alarms.

Figure 4 presents the extended evaluation of the proposed ACO-CNN-BiLSTM intrusion detection framework. (a) The per-class ROC-AUC results for benign and DoS traffic both reached 0.9995, confirming excellent discrimination capability. (b) The per-class precision–recall curves maintained high precision across recall values, demonstrating reliable detection with minimal false alarms. (c) Detection latency analysis under varying batch sizes showed latency decreasing as batch size increased, highlighting computational efficiency and suitability for near-real-time applications. (d) Cross-dataset accuracy comparison across CICIDS2017, UNSW-NB15, and CSE-CIC-IDS2018 datasets revealed consistently high classification accuracy, confirming the robustness, scalability, and generalization of the framework across diverse network environments.

Table 3 shows a comparative evaluation of the proposed ACO-CNN-BiLSTM intrusion detection framework against several recent IDS studies using common performance measures, including accuracy, precision, recall, F1-score, and ROC-AUC. In general, the comparison indicates that the proposed model achieves better overall classification results and a stronger separation ability.

Table 3 Evaluation metrics compared with recent IDS studies

Study	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC
Alotaibi and Rassam (2023)	97.51	96.84	95.72	96.21	0.987
Mohammad et al. (2024)	92.00	91.10	90.40	90.75	0.952
Susilo et al. (2025)	97.90	97.10	96.80	96.95	0.991
Zhang et al. (2025)	98.10	97.42	96.88	97.15	0.995
Proposed ACO-CNN-BiLSTM	**98.65**	**97.37**	**95.00**	**98.65**	**0.9995**

Table 4 presents the cross-dataset validation results of the proposed ACO-CNN-BiLSTM intrusion detection framework using the CICIDS2017, UNSW-NB15, and CSE-CIC-IDS2018 benchmark datasets. The evaluation demonstrates that the proposed model consistently maintains high classification accuracy, precision, recall, F1-score, and ROC-AUC values across heterogeneous traffic environments. These findings confirm the framework's robustness, scalability, and generalization capability under varying intrusion detection scenarios.

In the results, the dataset contained 454,322 benign traffic samples, making up 38.6%, and 721,458 DoS attack samples, making up 61.4%. To address this imbalance, weighted binary loss was applied to balance benign flows, while stratified balancing with regularization was used for DoS traffic, ensuring reduced bias and stable learning across both classes.

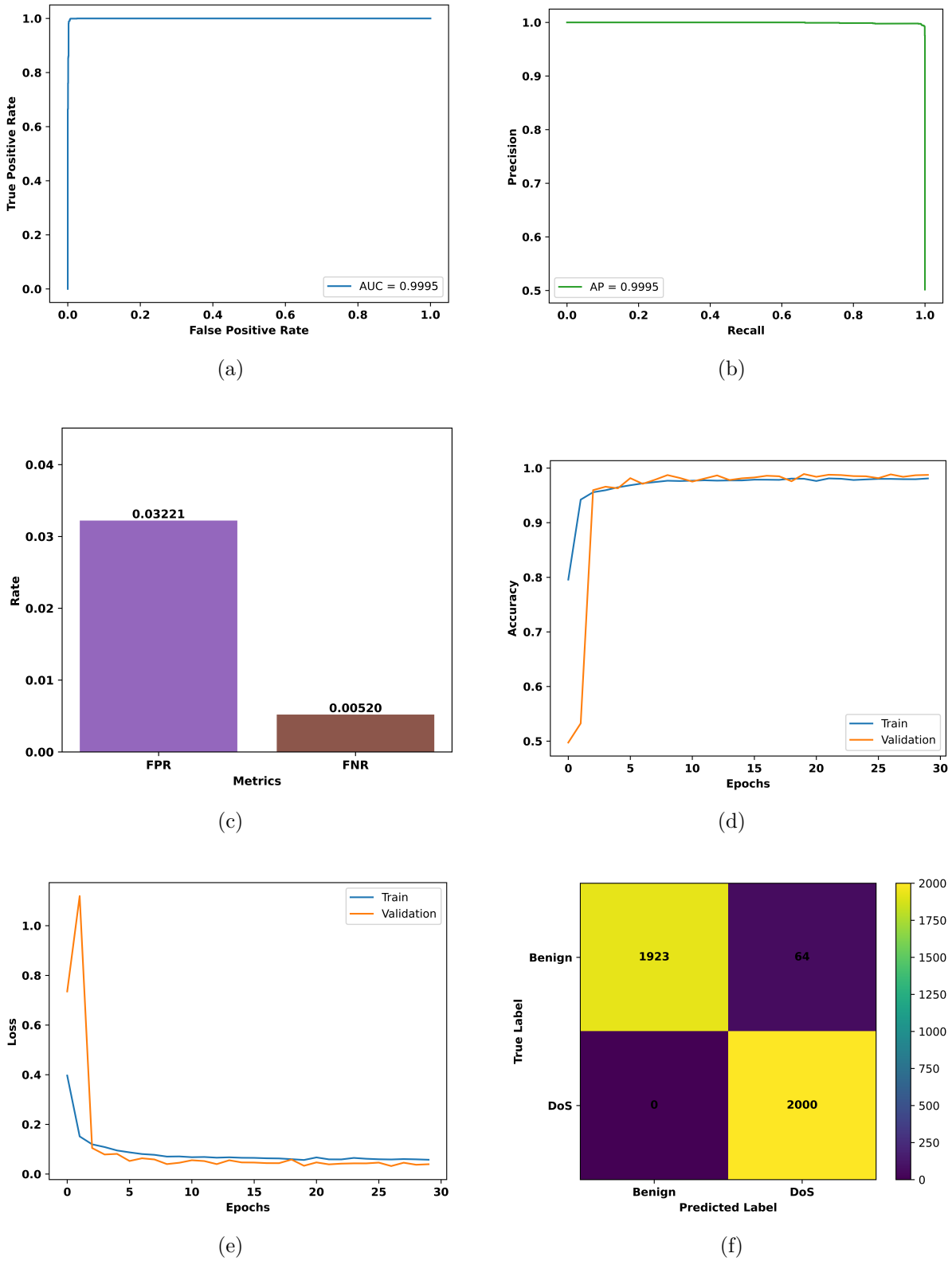

Figure 3 Performance evaluation: (a) ROC curve, (b) precision–recall curve, (c) FPR–FNR analysis, (d) training vs. validation accuracy, (e) training vs. validation loss, and (f) confusion matrix.

5. Discussion

Results confirm that the proposed ACO-CNN-BiLSTM framework effectively enhances DoS intrusion detection by combining optimized feature selection with hybrid deep learning. It

achieved 98.39% accuracy, 96.90% precision, and 95.00% recall, demonstrating high detection capability with reduced false alarms. ACO improves generalization by eliminating redundant features, consistent with recent findings (Hui et al., 2023), while the CNN-BiLSTM architecture effectively captures both spatial and bidirectional temporal traffic patterns (Hu et al., 2023). Compared with existing IDS models, the proposed framework delivers more accurate and stable performance, making it suitable for real-world DoS detection in complex network environments.

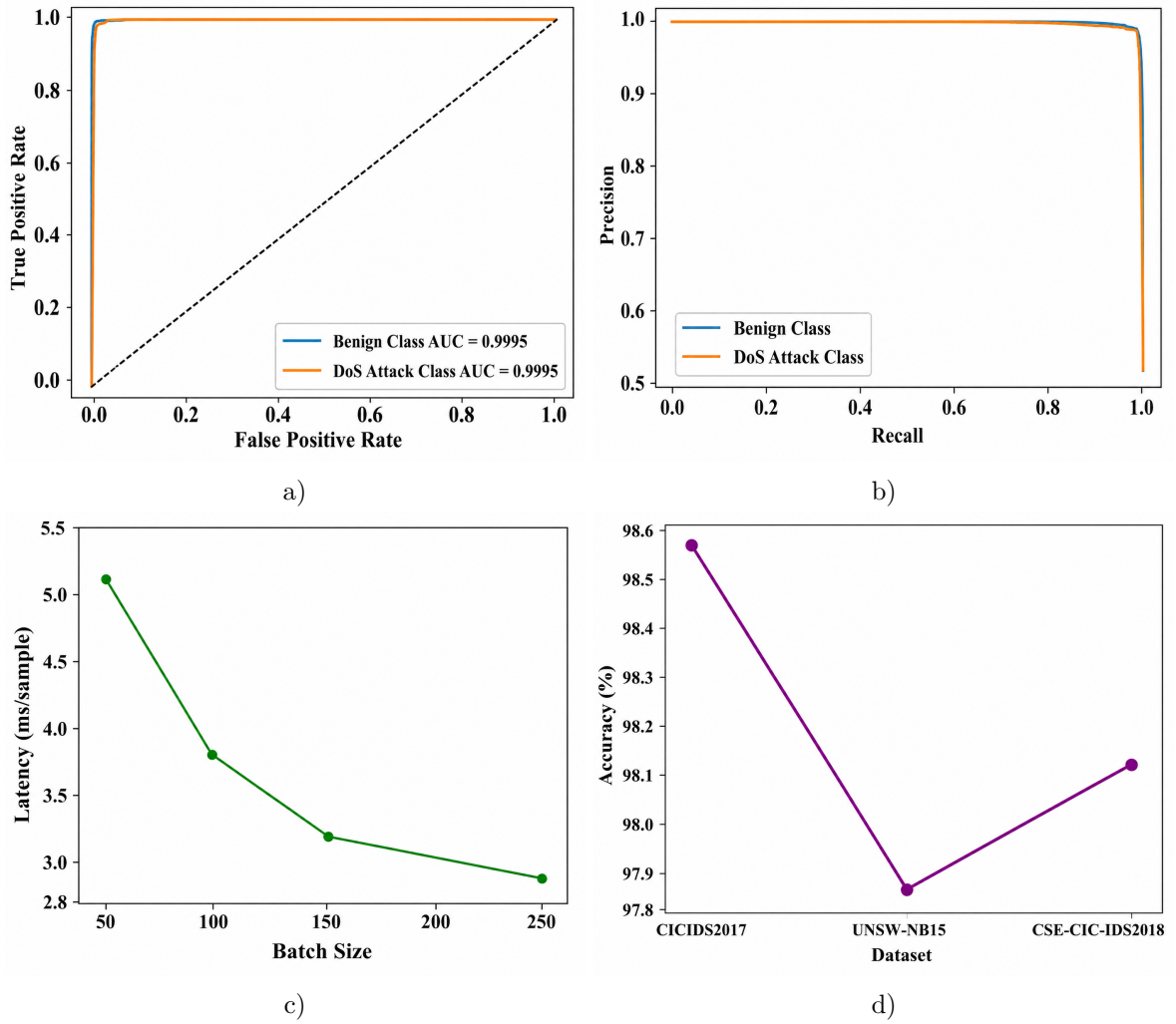

Figure 4 Extended evaluation of the ACO-CNN-BiLSTM framework: a) ROC-AUC, b) precision–recall, c) detection latency, and d) cross-dataset accuracy

Table 4 Cross-dataset performance evaluation of the proposed ACO-CNN-BiLSTM framework

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC
CICIDS2017	98.65	97.37	95.00	98.65	0.9995
UNSW-NB15	97.84	96.92	94.71	96.48	0.9941
CSE-CIC-IDS2018	98.12	97.05	95.88	96.44	0.9968

6. Conclusions and Future Enhancements

This study introduced an ACO-optimized CNN-BiLSTM intrusion detection framework designed to overcome key limitations of existing IDS solutions, including excessive feature dimensionality, redundant traffic characteristics, and insufficient temporal modeling. By integrating

ACO for feature selection with a hybrid CNN-BiLSTM architecture, the proposed system effectively reduced redundancy, improved computational efficiency, and enhanced detection accuracy. The framework was evaluated on three widely used benchmark datasets, CICIDS2017, UNSW-NB15, and CSE-CIC-IDS2018, under realistic traffic conditions. Preprocessing steps such as filtering, normalization, and feature engineering ensured data quality and consistency. Experimental results demonstrated strong performance, achieving an accuracy of 98.39%, precision of 96.90%, recall of 95.00%, and F1-score of 98.43%, outperforming several existing CNN- and LSTM-based intrusion detection models. These findings confirm the robustness, scalability, and practical applicability of the proposed approach, particularly in environments requiring high accuracy and low false-alarm rates. Beyond accuracy, the framework showed potential for real-time deployment with moderate resource usage and low inference latency, making it suitable for IoT and edge computing scenarios where memory and processing constraints are critical. Future work will focus on developing lightweight variants of the CNN-BiLSTM architecture tailored for resource-constrained devices and adapting the model to evolving attack behaviors.

Data Availability Statement

The datasets used in this study are publicly available at the following repositories: CICIDS2017 (https://www.kaggle.com/datasets/chethuhn/network-intrusion-dataset), CSE-CIC-IDS2018 (https://www.kaggle.com/datasets/dhoogla/csecicids2018), and UNSW-NB15 (https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15).

Conflict of Interest

The authors declare that there is no conflict of interest associated with this submission.

Acknowledgement

The authors extend their appreciation to Prince Sattam bin Abdulaziz University for funding this research work through the project number (PSAU/2025/01/34918).

References

Ahmed, M., & Alnatheer, S. (2025). Intrusion detection in a digital twin-enabled secure industrial internet of things environment for industrial sustainability. *Engineering, Technology & Applied Science Research*, *15*(3), 21263–21269. https://doi.org/10.48084/etasr.10128

Ali, A., Charfeddine, M., Hamed, B., Albalwy, F., Alqarafi, A., & Hussain, A. (2024). Unveiling machine learning strategies and considerations in intrusion detection systems: A comprehensive survey. *Frontiers in Computer Science*, *6*. https://doi.org/10.3389/fcomp.2024.1387354

Aljabri, W., Abdul Hamid, M., & Mosli, R. (2025). Enhancing real-time intrusion detection system for in-vehicle networks by employing novel feature engineering techniques and lightweight modeling. *Ad Hoc Networks*, *169*, 103737. https://doi.org/10.1016/j.adhoc.2024.103737

Al-Na'amneh, Q., Aljaidi, M., Nasayreh, A., Gharaibeh, H., Al Mamlook, R. E., Jaradat, A. S., Alsarhan, A., & Samara, G. (2024). Enhancing iot device security: Cnn-svm hybrid approach for real-time detection of dos and ddos attacks. *Journal of Intelligent Systems*, *33*(1). https://doi.org/10.1515/jisys-2023-0150

Alotaibi, A., & Rassam, M. (2023). Enhancing the sustainability of deep-learning-based network intrusion detection classifiers against adversarial attacks. *Sustainability*, *15*(12), 9801. https://doi.org/10.3390/su15129801

Ataa, M., Sanad, E., & El-khoribi, R. (2024). Intrusion detection in software defined network using deep learning approaches. *Scientific Reports*, *14*(1), 29159. https://doi.org/10.1038/s41598-024-79001-1

Bakro, M., Kumar, R. R., Alabrah, A., Author4, A., Author5, A., Author6, A., & Abdelsalam, A. (2023). An improved design for a cloud intrusion detection system using hybrid features selection approach with ml classifier. *IEEE Access*, *11*, 64228–64247. https://doi.org/10.1109/ACCESS.2023.3289405

Biswas, S., Goswami, R. S., Reddy, K. H. K., Mohanty, S. N., & Ahmed, M. A. (2024). Advancing quantum communication security: Metamaterial based quantum key distribution with enhanced protocols. *IET Quantum Communication*, *5*(4), 399–416. https://doi.org/10.1049/qtc2.12116

Esmaeili, M., Goki, S., Masjidi, B., Sameh, M., Gharagozlou, H., & Mohammed, A. (2022). Ml-ddosnet: Iot intrusion detection based on denial-of-service attacks using machine learning methods and nsl-kdd. *Wireless Communications and Mobile Computing*, *2022*(1), 8481452. https://doi.org/10.1155/2022/8481452

Espinoza-Zelaya, C., & Moon, Y. (2022). Resilience enhancing mechanisms for cyber-manufacturing systems against cyber-attacks. *IFAC-PapersOnLine*, *55*, 2252–2257. https://doi.org/10.1016/j.ifacol.2022.10.043

Gong, X., Yang, Y., Zhang, Y., Li, N., Guan, Y., & Jiang, R. (2025). Feature selection method for network intrusion based on hybrid meta-heuristic dynamic optimization algorithm. *Computers & Security*, *156*, 104512. https://doi.org/10.1016/j.cose.2025.104512

Hewapathirana, I. (2025). A comparative study of two-stage intrusion detection using modern machine learning approaches on the cse-cic-ids2018 dataset. *Knowledge*, *5*(1), 6. https://doi.org/10.3390/knowledge5010006

Hozouri, A., Mirzaei, A., & Effatparvar, M. (2025). A comprehensive survey on intrusion detection systems with advances in machine learning, deep learning and emerging cybersecurity challenges. *Discover Artificial Intelligence*, *5*(1), 314. https://doi.org/10.1007/s44163-025-00578-1

Hu, J., Liu, Y., Lam, K.-M., & Lou, P. (2023). Stfe-net: A spatial-temporal feature extraction network for continuous sign language translation. *IEEE Access*, *11*, 46204–46217. https://doi.org/10.1109/ACCESS.2023.3234743

Hui, J., Wu, C., Li, X., Huang, L., Jiang, Y., & Zhang, B. (2023). The effect of light availability on photosynthetic responses of four aglaonema commutatum cultivars with contrasting leaf pigment. *Applied Sciences*, *13*(5), 3021. https://doi.org/10.3390/app13053021

Kavitha, P., & Lalitha, S. (2024). Ant colony optimization algorithm for feature selection in sentiment analysis of social media data. *ICTACT Journal on Soft Computing*, *14*(4), 3334–3339. https://doi.org/10.21917/ijsc.2024.0468

Kim, S., & Kim, D. (2023). Securing the cyber resilience of a blockchain-based railroad non-stop customs clearance system. *Sensors*, *23*, 2914. https://doi.org/10.3390/s23062914

Kunhare, N., Tiwari, R., & Dhar, J. (2022). Intrusion detection system using hybrid classifiers with meta-heuristic algorithms for the optimization and feature selection by genetic algorithm. *Computers and Electrical Engineering*, *103*, 108383. https://doi.org/10.1016/j.compeleceng.2022.108383

Mohammad, R., Saeed, F., Almazroi, A., Alsubaei, F., & Almazroi, A. (2024). Enhancing intrusion detection systems using a deep learning and data augmentation approach. *Systems*, *12*(3), 79. https://doi.org/10.3390/systems12030079

Mohsenabad, H. N., & Tut, M. A. (2024). Optimizing cybersecurity attack detection in computer networks: A comparative analysis of bio-inspired optimization algorithms using the cse-cic-ids 2018 dataset. *Applied Sciences*, *14*(3), 1044. https://doi.org/10.3390/app14031044

Moustafa, N., Koroniotis, N., Keshk, M., Zomaya, A., & Tari, Z. (2023). Explainable intrusion detection for cyber defences in the internet of things: Opportunities and solutions. *IEEE Communications Surveys & Tutorials*, *25*(3), 1775–1807. https://doi.org/10.1109/COMST.2023.3280465

Narmadha, S., & Balaji, N. (2025). Improved network anomaly detection system using optimized autoencoder-lstm. *Expert Systems with Applications*, *273*, 126854. https://doi.org/10.1016/j.eswa.2025.126854

Neto, E., Iqbal, S., Buffett, S., Sultana, M., & Taylor, A. (2025). Deep learning for intrusion detection in emerging technologies: A comprehensive survey and new perspectives. *Artificial Intelligence Review*, *58*(11), 340. https://doi.org/10.1007/s10462-025-11346-z

Owusu, E. O., Danlard, I., Opoku, G., Dede, A., Klogo, G. S., Boateng, K. O., & Akowuah, E. K. (2026). A systematic review of machine learning and deep learning techniques for ddos attack mitigation in iot and edge computing systems. *Security and Privacy*, *9*(1), e70147. https://doi.org/10.1002/spy2.70147

Rajput, V., Mulay, P., & Mahajan, C. (2024). Bio-inspired algorithms for feature engineering: Analysis, applications and future research directions. *Information Discovery and Delivery*, *53*(1), 56–71. https://doi.org/10.1108/IDD-11-2022-0118

Reed, A., Dooley, L., & Kouadri Mostefaoui, S. (2025). Minimal overhead modelling of slow dos attack detection for resource-constrained iot networks. *Future Internet*, *17*(10), 432. https://doi.org/10.3390/fi17100432

Safitra, M., Lubis, M., & Fakhrurroja, H. (2023). Counterattacking cyber threats: A framework for the future of cybersecurity. *Sustainability*, *15*(18), 13369. https://doi.org/10.3390/su151813369

Sajid, M., Malik, K. R., Almogren, A., Malik, T. S., Khan, A. H., Tanveer, J., & Rehman, A. U. (2024). Enhancing intrusion detection: A hybrid machine and deep learning approach. *Journal of Cloud Computing*, *13*(1), 123. https://doi.org/10.1186/s13677-024-00685-x

Susilo, B., Muis, A., & Sari, R. (2025). Intelligent intrusion detection system against various attacks based on a hybrid deep learning algorithm. *Sensors*, *25*(2), 580. https://doi.org/10.3390/s25020580

Udurume, M., Shakhov, V., & Koo, I. (2024). Comparative analysis of deep convolutional neural network–bidirectional long short-term memory and machine learning methods in intrusion detection systems. *Applied Sciences*, *14*(16), 6967. https://doi.org/10.3390/app14166967

Vanitha, S., & Balasubramanie, P. (2022). Improved ant colony optimization and machine learning based ensemble intrusion detection model. *Intelligent Automation & Soft Computing*, *36*(1), 849–864. https://doi.org/10.32604/iasc.2023.032324

Vishnu, P., & Soumya, S. (2024). Advancements in anomaly detection techniques in network traffic: The role of artificial intelligence and machine learning. *Journal of Scientific Research and Technology*, 38–48. https://doi.org/10.61808/jsrt114

Yuan, X., Han, S., Huang, W., Ye, H., Kong, X., & Zhang, F. (2024). A simple framework to enhance the adversarial robustness of deep learning-based intrusion detection system. *Computers & Security*, *137*, 103644. https://doi.org/10.1016/j.cose.2023.103644

Zhang, C., Li, J., Wang, N., & Zhang, D. (2025). Research on intrusion detection method based on transformer and cnn-bilstm in internet of things. *Sensors*, *25*(9), 2725. https://doi.org/10.3390/s25092725

Zhang, X., & Yin, J. (2025). Optimized extreme learning machines with deep learning for high-performance network traffic classification. *Scientific Reports*, *15*(1), 33199. https://doi.org/10.1038/s41598-025-16980-9

Zhong, X., Liu, Y., Xie, K., & Xie, S. (2022). A local electricity and carbon trading method for multi-energy microgrids considering cross-chain interaction. *Sensors*, *22*(18), 6935. https://doi.org/10.3390/s22186935

Zhou, X., Yang, J., Li, Y., Li, S., Su, Z., & Lu, J. (2025). Ec-trl: Evolutionary-weighted clustering and transformer-augmented reinforcement learning for dynamic resource scheduling in edge cloud environments. *IEEE Internet of Things Journal*, *12*(6), 7503–7517. https://doi.org/10.1109/JIOT.2024.3496200