PENINSULA PUBLISHING LLC

SHIFRA

Vol. **(2026)**, 2026, **pp**. 158–176
ISSN: 3078-3186

Research Article

Deep Learning-Based Cybersecurity Framework for Proactive Intrusion Detection on the CSE-CIC-IDS2018 Dataset

Ali Rachini [1, *], Musaria Karim Mahmood[2],

[1] *Faculty of Arts and Sciences, Holy Spirit University of Kaslik (USEK), Jounieh, Lebanon*

[2] *Ankara Yildirim Beyazit University, Energy systems engineering department, Türkiye*

ARTICLEINFO

Article History

Received 3 Apr 2026

Revised: 24 May 2026
Accepted 21 Jun 2026
Published 12 Jul 2026

Keywords

Deep Learning,

Intrusion Detection System (IDS),

CNN-LSTM,

CSE-CIC-IDS2018,

Cybersecurity.

ABSTRACT

Network infrastructure has become more complex and the amount of cybersecurity challenges has grown with the introduction of cloud computing, the Internet of Things (IoT) and next-generation communication technologies. The known attack and known signatures are the reasons why traditional IDSs fail to detect sophisticated and new attacks; they are signature-based and programmed by hand. To overcome these drawbacks, this work presents a deep learning-based cybersecurity framework for proactively detecting the intrusion in CSE-CIC-IDS2018 benchmark dataset. The proposed system combines spatial and temporal features of network traffic using a hybrid Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) network architecture with a systematic data preprocessing pipeline. In the preprocessing stage, data quality and learning efficiency are enhanced by data cleaning, handling missing data, removing duplicate data, encoding labels, normalizing features, and splitting the data into training and testing sets. The CNN part is used to automatically learn discriminative spatial features, and the LSTM was used to learn the sequential traffic dependencies to see the relation between the sequential sequence and attack detection capability was improved. Experiments were carried out on Python, TensorFlow, Keras and Google Colab with GPU support. The proposed framework was evaluated according to the accuracy, precision, recall, F1-score, and ROC-AUC and the results were found to be 98.64%, 98.41%, 98.18%, 98.29%, and 99.12%, respectively, which are considered to be excellent classification results and strong discrimination ability. Moreover, the framework demonstrated good computational efficiency with an average inference time of 2.8ms per network flow and a moderate use of GPU memory. Results of comparative analysis with the latest deep learning-based intrusion detection methods also validate the competitiveness and applicability of the proposed framework. The results highlight the efficiency, reliability, and effectiveness of the hybrid CNN-LSTM framework in proactively detecting network intrusions in contemporary cybersecurity settings.

1. INTRODUCTION

Today's network infrastructure is so complex and large, thanks to the growth of digital transformation, cloud computing, Internet of Things (IoT), industrial automation and next-generation communication networks. The technologies have opened the door for smooth integration and smart services across various application segments including healthcare, finance, transport, smart cities, critical industrial systems and more. But as the number of connected devices continues to grow, the cybersecurity attack surface has also increased, making networks more vulnerable to an array of new sophisticated cyber threats. With the increasing sophistication, automation and persistence of cyberattacks, network proactivity protection has become one of the most critical challenges for contemporary cybersecurity systems. Hence, Intrusion Detection Systems (IDSs) are important that are able to continuously monitor network activities, detect malicious activities and prevent attacks to compromise the system integrity and availability [1]. The conceptual architecture of deep learning-based cybersecurity framework for proactive intrusion detection is shown in Figure 1.

*Corresponding author email: AliRachini@usek.edu.lb
DOI: https://doi.org/10.70470/SHIFRA/2026/009

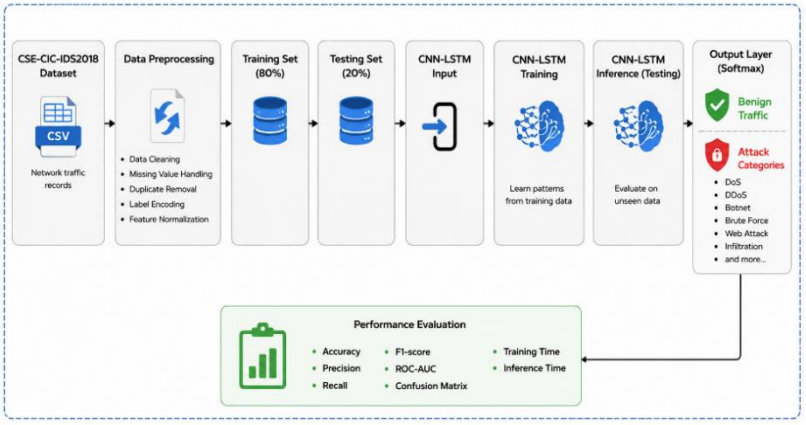

Fig. 1. Conceptual overview of deep learning-based cybersecurity framework for proactive intrusion detection using the CSE-CIC-IDS2018 dataset.

The most common type of IDS detector is one that uses signature-based detection, statistical analysis or manually designed detection rules to recognize malicious activity. While these traditional methods still work for attacks they've seen before, they have drawbacks when dealing with polymorphic malware, advanced persistent threats (APTs), and intrusions that change quickly, or zero-day attacks. Signature-based systems are also prone to false-positive alerts, as network traffic is constantly evolving, while anomaly-based systems are time-consuming and costly to update to ensure that they stay current with the latest attacks. In addition, cloud platforms, IoT devices, and edge computing environments have created a massive amount, variety, and speed of network data that have significantly diminished the scalability and adaptability of traditional intrusion detection systems [1,2]. These constraints have spurred an effort to explore intelligent data-driven solutions that can be used to automatically discover complex network behaviors by learning without manual design of detection rules.

With recent developments in AI, specifically deep learning, cybersecurity research has been dramatically changed, allowing researchers to automatically extract features from massive amounts of network traffic and adaptively detect threats. Hierarchical representation learning approaches have been successfully used to detect known and unknown cyberattacks using deep learning architectures like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, DenseNet, Bidirectional LSTM (BiLSTM) and hybrid CNN-LSTM models [3-8]. Conventional machine learning algorithms require a lot of hand-crafted features; in contrast, deep learning models are able to automatically derive discriminative spatial and temporal features directly from raw or preprocessed network traffic, leading to higher detection accuracy and better generalization ability when adapting to different classes of attacks. Recently, hybrid deep learning architectures, such as CNN-LSTMs, which use CNN to cope with local spatial relations and LSTM to capture long-term sequential trends in network traffic received much attention due to their superior performance in detecting intrusions [9,10].

However, a number of research challenges are still to be addressed, and despite these promising strides, the scientific community continues to grapple with a number of challenges. Most of the existing intrusion detection frameworks still depend on the relatively old benchmark datasets, like the NSL-KDD and KDD Cup 1999, which don't reflect the complexity and diversity of today's network environment. Furthermore, many published works focus mainly on classification accuracy and fail to analyse computational efficiency, inference speed, scalability and feasibility of real-world deployment. Multiple deep learning models also have overly complex architectures that need significant computing power, which is not suitable for resource-limited cybersecurity use cases or real-time intrusion detection systems. Besides, few studies have conducted a thorough investigation on the hybrid deep learning models by exploiting the more recent and realistic CSE-CIC-IDS2018 dataset that has been created with different attack scenarios to best capture the current cyber threats [11-14].

Driven by these challenges, this study presents an innovative cybersecurity framework for proactive intrusion detection system based on deep learning technique and CSE-CIC-IDS2018 benchmark dataset. The proposed framework is a hybrid CNN-LSTM structure to learn automatically the spatial and temporal characteristics of the network traffic and detect and classify attacks with an accuracy. It involves systematic approaches for data preprocessing like data cleaning, removing duplicate data, filling in missing values, label encoding, normalizing features, and splitting data sets prior to training machine learning models. A comprehensive evaluation is performed on the proposed framework by various metrics including Accuracy, Precision, Recall, F1 score, Receiver Operating Characteristic Area Under the Curve (ROC-AUC), Inference time and Training time under realistic network intrusion scenarios to check the effectiveness, robustness, and computational efficiency of the proposed framework [15].

The major contributions of this work are: A detailed deep learning-based cybersecurity framework is proposed in first part, for the proactive intrusion detection based on the recent CSE-CIC-IDS2018 benchmark dataset. Second, to improve the accuracy of ID and classification, a hybrid CNN-LSTM network is designed to learn the spatial and temporal features of

network traffic at the same time. Third, a systematic data preprocessing pipeline is created to enhance the quality of the data and further increase the efficiency of learning before feeding the data into the model. Fourth, the proposed framework is extensively evaluated basing on the different performance metrics including accuracy, precision, recall, f1 score, roc-auc, training time and inference time to ensure robustness and the computational efficiency of the proposed framework. Finally, the proposed approach is compared with recent state-of-the-art deep learning intrusion detection models to show that the proposed approach is capable of solving the modern challenges in cybersecurity and still remains applicable to real world applications.

2. PROPOSED METHODOLOGY

The proposed deep learning-based cybersecurity framework is meant to offer an intelligent and proactive intrusion detection mechanism which could accurately detect not only the benign network operations but also the malicious ones and also with high computational efficiency. In contrast to signature-based or pre-defined rule-based IDS, the proposed framework automatically learns complex network traffic characteristics using a hybrid Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) network. The proposed model combines both the spatiotemporal feature extraction and temporal sequence learning, leading to the effective detection of sophisticated and unseen cyberattacks.

The proposed framework is represented by seven successive stages as shown in Figure 2. The first phase is obtaining the realistic network traffic data set CSE-CIC-IDS2018, which consists of various types of cyberattacks and normal network traffic. The second step consists of a detailed pre-processing pipeline involving data cleaning, missing value imputation, removal of duplicate values, encoding of the labels, normalisation of the features, and finally splitting the data into training and test sets. This pre-processing will ensure consistency in data, eliminate redundant data, and provide high quality representations for features that will be utilized in deep learning model training.

The preprocessed feature vectors are then fed into the proposed hybrid CNN-LSTM model as normalized feature vectors. The CNN is used to automatically extract spatial features from the traffic flow and recognize discriminative patterns of traffic behavior related to various intrusion types, by performing 1D convolution operations. Specifically, the convolutional network is composed of a Conv1D layer with 64 filters and kernel size 3, followed by a Rectified Linear Unit (ReLU) activation function, which is used to introduce the non-linearity, and a MaxPooling1D layer with a pool size of 2, which reduces the dimensionality of the features while keeping the most informative traffic characteristics. These operations produce smaller feature maps that are able to accurately represent the spatial relationships between network traffic records.

The extracted feature maps are then passed to the LSTM layer with 128 memory units where temporal dependencies and sequential relationships between network traffic features are learned. The LSTM architecture incorporates gating mechanisms and memory cells that enable the proposed framework to capture long-term contextual information, which is crucial for accurately identifying changing intrusion patterns and sophisticated cyberattacks, unlike conventional feedforward networks. A Dropout layer is used prior to sending the learned representations to a fully connected Dense layer with 64 neurons, and with a Dropout rate of 0.5 to reduce overfitting and improve the model's generalization ability.

The output layer used in the last classification step is a softmax layer, which is used for multi-class intrusion classification. The Softmax classifier calculates the probability distribution over all the pre-defined traffic classes in the CSE-CIC-IDS2018 and then uses the maximum probability class for predicting the final class. Consequently, each connection that enters the network is marked as “benign traffic” or one of the various categories of malicious traffic in the benchmark data set (DoS, DDoS, Brute Force, Botnet, Web Attacks, Infiltration and other types of intrusion).

Finally, a comprehensive set of classification and computational performance measures are presented to assess the capabilities of the proposed framework. It includes several metrics such as Accuracy, Precision, Recall, F1-score, Receiver Operating Characteristic Area Under the Curve (ROC-AUC), confusion matrix analysis, training time, and inference time, providing a comprehensive analysis of the predictive performance and computational efficiency in realistic cybersecurity scenarios.

In summary, this framework uses systematic data preprocessing, efficient spatial feature extraction, sequential behavior learning and powerful multi-class classification into one solution to cybersecurity for proactive intrusion detection. The modular design enables easy integration of new deep learning algorithms, feature engineering techniques, or explainable artificial intelligence (XAI) in the future, minimizing the need for extensive changes to the main structure. Techniques, or explainable artificial intelligence (XAI) in the future, minimizing the need for extensive changes to the main structure.

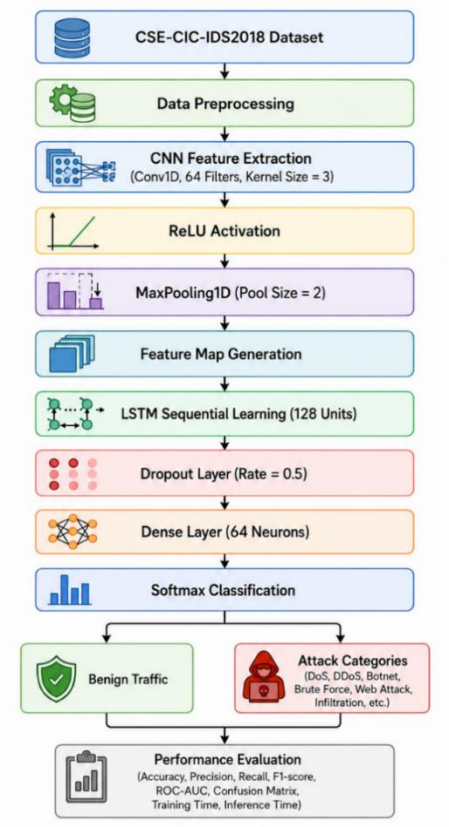

Fig 2. Overall architecture of the proposed deep learning-based cybersecurity framework for proactive intrusion detection using the CSE-CIC-IDS2018 dataset.

2.1 CSE-CIC-IDS2018 Dataset

An intrusion detection system that is based on deep learning strongly relies on the quality, variety and realism of the dataset used for training and testing. The benchmark dataset used in this study was CSE-CIC-IDS2018 as it is a comprehensive dataset of modern network traffic and covers a variety of legitimate network traffic and different types of cyberattacks. The dataset was created together by the Canadian Institute for Cybersecurity (CIC) and the Communications Security Establishment (CSE) with the goal of producing a realistic enterprise network environment populated with objects that would mimic modern-day cybersecurity situations found in real-life applications.

CSE-CIC-IDS2018 is a new benchmark dataset with realistic bidirectional network flow features extracted with CICFlowMeter-V3, which avoids the shortcomings of the previous benchmark datasets (such as KDD Cup 1999 and NSL-KDD) that have been outdated by repetitive records and attack behaviors. These flow-based features describe some of the key attributes of communication, such as packet statistics, flow duration, transmission rates, protocol information, and temporal/behavioral features, which can be used to more effectively differentiate between normal and malicious network activities in the context of deep learning models.

It contains a large number of different types of cyber-attacks to cover both common and newer forms of intrusions. These attacks encompass Distributed Denial of Service (DDoS), Denial of Service (DoS), Brute Force attacks, Web-based attacks, Botnet activities, Infiltration attacks and other malicious activities commonly observed in today's enterprise networks. The proposed framework supports both binary intrusion detection and multi-class attack classification, resulting from the diversity of attack categories, enhancing its applicability to real-world cybersecurity environments.

The data set was split into two classes, benign and malicious, when developing the model to make the process easier. The dataset was divided into the classes, benign and malicious, before applying the preprocessing pipeline described in the next section, to make the process of model development easier. In the preprocessing phase, categorical data was transformed into numerical format, numerical data was scaled, and inconsistent or incomplete information was eliminated to enhance the consistency and efficiency of the learning process. It enables the development of a solid dataset for training of the proposed framework of hybrid CNN-LSTM and its evaluation in detecting various types of network intrusions in realistic scenarios. The general characteristics of the CSE-CIC-IDS2018 data set used in this study is summarized as shown in Table I below.

TABLE I. GENERAL CHARACTERISTICS OF THE CSE-CIC-IDS2018 DATASET

Attribute	Description
Dataset Name	CSE-CIC-IDS2018
Source	Canadian Institute for Cybersecurity (CIC) & Communications Security Establishment (CSE)
Data Type	Bidirectional Network Flow Records
Feature Extraction Tool	CICFlowMeter-V3
Number of Features	80 Flow-Based Features
Traffic Classes	Benign and Malicious
Classification Type	Binary and Multi-Class Intrusion Detection
Attack Categories	DDoS, DoS, Brute Force, Botnet, Web Attacks, Infiltration, Heartbleed, SQL Injection, SSH/FTP Brute Force
Network Protocols	TCP, UDP, HTTP, HTTPS and others
Application Domain	Enterprise Network Cybersecurity
Dataset Purpose	Training and Evaluation of Intrusion Detection Systems

The CSE-CIC-IDS2018 dataset is diverse with a wide range of normal network traffic and multiple types of intrusion categories to allow for comprehensive evaluation of intrusion detection systems where realistic cybersecurity scenarios can be evaluated. Depending on their behavioural features, different types of attacks are added to the proposed hybrid CNN-LSTM model to learn representative patterns of both benign and malicious network activities. This diversity can enhance the model robustness and generalization capabilities to other classes of intrusions and deployment environments. The key categories of traffic that are represented in the data set are listed in Table II.

TABLE II. DISTRIBUTION OF NETWORK TRAFFIC CLASSES IN THE CSE-CIC-IDS2018 DATASET

Traffic Category	Description
Benign	Legitimate network traffic generated by normal user activities
DDoS	Distributed Denial of Service attacks
DoS	Denial of Service attacks
Brute Force	Password guessing attacks including FTP and SSH
Botnet	Malware-controlled network communications
Web Attacks	SQL Injection, XSS, and Brute Force Web attacks
Infiltration	Unauthorized network penetration attempts
Heartbleed	SSL/TLS Heartbleed vulnerability exploitation

2.2 Data Preprocessing

Data preprocessing is one of the most important stages of designing a Deep Learning based Intrusion detection system since the quality and consistency of data directly influences the learning ability and the capacity of the model to forecast. Raw CSE-CIC-IDS2018 data includes a combination of network traffic data (which may contain missing values and identical records) as well as categorical attributes and features with differing numerical scales. If not properly preprocessed, these inconsistencies can lead to higher computational complexity, slower convergence of the learning algorithm and a decrease in classification accuracy. Therefore, a complete preprocessing pipeline was designed prior to training the model to ensure the quality and correct representation of features from the data.

The first stage of data preprocessing is to identify and remove irrelevant attributes and corrupted records, which is known as data cleaning. This step will filter out samples with noise that may cause excess noise in the model training process. To If records contain invalid values or feature representations that are inconsistent, have the learning process more reliable, the records will be excluded from the network flow.

The second part concerns the treatment of missing values. Any feature with values of null or undefined or not a number (NaN) is carefully treated to prevent the loss of information when training the model. If numbers are missing, they are replaced in a suitable way using statistical methods and the data sets that have too many numbers missing are not taken for their integrity. In this way, all the input vectors given to the deep learning model are complete and informative.

After missing value treatment, the removal of duplicate records is carried out to remove the redundant records of the network flow. The learning algorithm might suffer from noise when samples are replicated, as it may become more difficult for the model to learn from new traffic patterns due to the over-representation of the repeated ones. This helps eliminate redundant instances, and it also speeds up the training process by avoiding redundant computations.

Label encoding is then used to map the categorical class labels of benign and attack classes to numerical values. The conversion allows the deep learning model to handle the classification output during supervised learning and maintain the semantic consistency of different intrusion classes. Intrusion detection and attack classification are performed in either a binary or multi-class manner, according to the classification objective, with labels being encoded accordingly.

The extracted network flow features are all of different numerical ranges and each have their own measurement unit; hence the input space is normalized. Each feature is normalized using Min-Max normalization method, so that all the features are optimized in the range of 0 to 1, ensuring that all the attributes are equally optimized during the optimization of the parameters. This normalization process speeds up the convergence of the gradients, helps stabilize the training of the network and avoids that features with large numbers take over the training. Finally, the preprocessed dataset is split into independent training and testing subsets to test the generalization ability of the proposed framework. In this research, an

80% proportion of data will be used for training the model, and the remaining 20% will be used to test and evaluate the model performance. The training subset is used to tune the CNN-LSTM, while the testing subset is used to evaluate the CNN-LSTM in detecting unseen network intrusions. The entire preprocessing procedure used in this study is presented in Table III, which shows the goal and significance of each preprocessing step before the development of deep learning models.

TABLE III. DATA PREPROCESSING OPERATIONS

Preprocessing Step	Purpose	Impact on Model Performance
Data Cleaning	Remove noisy, corrupted, and irrelevant records	Improves data quality and model reliability
Missing Value Handling	Replace or remove incomplete records	Prevents information loss and invalid inputs
Duplicate Removal	Eliminate repeated network flow records	Reduces data redundancy and overfitting
Label Encoding	Convert categorical attack labels into numerical values	Enables supervised deep learning classification
Feature Normalization	Scale numerical features to the range [0,1]	Accelerates convergence and stabilizes training
Train/Test Split (80:20)	Separate data for training and evaluation	Provides unbiased performance assessment

For better understanding of the proposed hybrid CNN-LSTM model preprocessing steps, the sequence of operations performed prior to the training of the proposed hybrid CNN-LSTM model are shown in figure 3. The preprocessing pipeline starts with the raw network traffic data from the CSE-CIC-IDS2018 dataset, then goes through data cleaning, missing value, duplicate removal, label encoding, feature normalization, and dataset partitioning. All of these operations work together to improve the consistency of the data, the quality of the features, the computational cost, and to prepare the data for efficient deep learning training. The proposed framework defines a standard processing framework that guarantees the quality of the network traffic features extracted, ensuring the reliability and robustness of the intrusion detection process by covering benign and malicious activities.

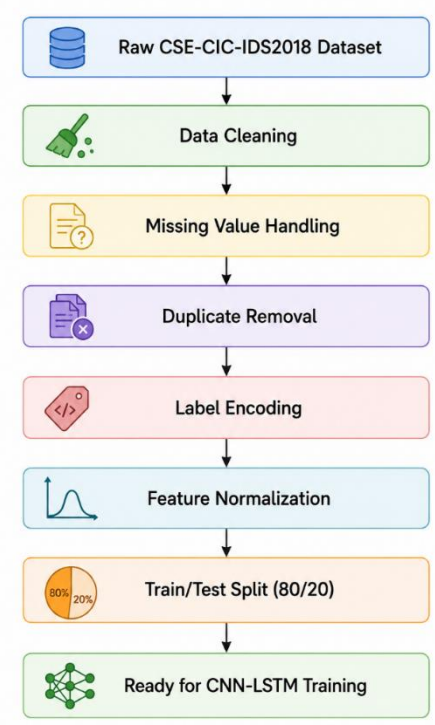

Fig. 3. Data preprocessing pipeline for the proposed intrusion detection framework.

2.3 Proposed Hybrid CNN-LSTM Model

This proposed Cybersecurity system is based on a hybrid CNN-LSTM architecture which automatically learns discriminative representations from network traffic features extracted from the CSE-CIC-IDS2018 dataset. This hybrid architecture can utilize CNN spatial feature extraction ability to capture the relationship between local features in network traffic and LSTM sequential learning ability to capture long-term temporal dependencies in network traffic. This integration greatly improves the model's ability to differentiate between non-threatening network traffic and multiple types of cyberattacks while keeping the computational efficiency.

The proposed model is divided into four main modules: CNN feature extraction module, LSTM for sequential learning module, fully connected classification module and Softmax output module. Every component is used for a particular function, which is contributing towards the whole intrusion detection process.

2.3.1CNN Architecture

In the first stage of the proposed model, the author utilizes a Convolutional Neural Network to automatically learn informative spatial features of normalized network traffic. The convolutional layers learn multiple learnable filters over the input feature space to build the hierarchical representations of feature space, instead of manually engineered features. The convolution operations are designed to detect local characteristics of the traffic related to each intrusion pattern, and the Rectified Linear Unit (ReLU) activation function adds nonlinearity and enhances the learning ability of the model. After convolution, a max-pooling layer is added to decrease the spatial size of the feature maps extracted in the previous layer, but keep the most important information. This operation reduces computational complexity, eliminates redundant feature representations and enhances the robustness of the learning process.

2.3.2LSTM Layer

The CNN layer outputs its feature map which is passed to the Long Short-Term Memory (LSTM) layer. The LSTM differs from traditional recurrent neural networks, in that its memory cell architecture and gating mechanisms allow it to learn long-term sequential dependencies. This allows the proposed framework to include temporal correlations, which are not easily captured with spatial analysis.

The sequential nature of network traffic is well captured by the LSTM layer, enabling the proposed framework to detect the changing patterns of attacks and differentiate complex attacks from legitimate communication activities. Consequently, the CNN and LSTM can be used together to enhance the learning ability, which leads to better performance for intrusion detection.

2.3.3Dense Layer

High-level feature representations created by the LSTM layer are then passed to a fully connected, dense layer. This layer combines the spatial and temporal information extracted from the picture with a compact representation of the features that can be used for the final decision. To prevent overfitting, dropout regularization is used before the dense layer, which randomly deactivates neurons during training to enhance the ability of the network to generalize in the future when it is tested on new network traffic.

2.3.4Softmax Classification

The proposed model's last step involves a Softmax classifier for multi-class intrusion classification. The output logits are transformed to probability distribution according to the previously defined traffic classes in the CSE-CIC-IDS2018 dataset by using the Softmax activation function. This Softmax activation function transforms the output logits to the probability distributions that match the defined traffic classes in the CSE-CIC-IDS2018 dataset. The result of the class that is most likely is picked out as the last class choice. In the evaluation scenario, the classifier can detect intrusion with binary classification (Benign versus Attack) and multi-class intrusion classification, with several intrusion classes. The overall architecture of the proposed hybrid CNN-LSTM model is illustrated in Figure 4 and the main architecture of the CNN-LSTM is summarized in Table IV.

TABLE IV. ARCHITECTURE OF THE PROPOSED HYBRID CNN-LSTM MODEL

Layer	Configuration	Purpose
Input Layer	80 Features	Receives normalized traffic features
Conv1D	64 Filters, Kernel=3	Spatial feature extraction
ReLU	Activation	Nonlinear transformation
MaxPooling1D	Pool Size=2	Dimensionality reduction
LSTM	128 Units	Sequential learning
Dropout	Rate=0.5	Prevent overfitting
Dense	64 Neurons	Feature integration
Softmax	Number of Classes	Intrusion classification

Before evaluating the proposed framework, the complete CNN-LSTM architecture is visually presented in Figure 4**,** illustrating the flow of information from the normalized network traffic features through the convolutional, sequential learning, and classification layers until the final intrusion prediction is generated.

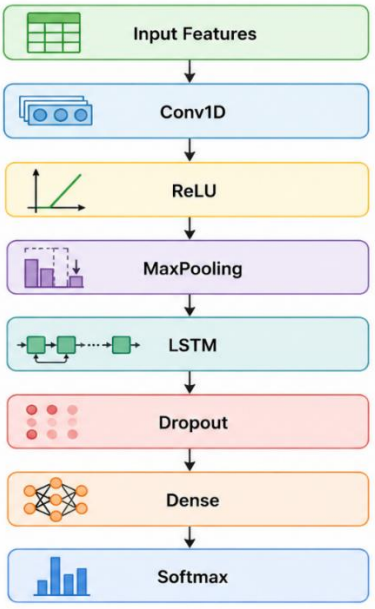

Fig. 4. Architecture of the proposed hybrid CNN-LSTM model for proactive intrusion detection.

2.4 Intrusion Detection Process

The intrusion detection process in the proposed architecture involves a series of integrated stages which convert the raw network traffic into accurate predictions of intrusion by using hybrid CNN-LSTM architecture. The first task is to get the preprocessed network traffic logs from the CSE-CIC-IDS2018 dataset. A feature vector is a normalized representation of each network flow, and contains a number of statistical and behavioural features that describe the communication session. The normalized feature vectors are then fed into the convolutional network that automatically learns multiple convolutional filters responsible for discriminative spatial representation learning from the input data. The extracted maps of features highlight relevant traffic features while eliminating redundant or less informative features. The max-pooling operations, which are not information lossy but are still able to reduce the dimensionality of the features, are also used.

Compact feature maps are then sent to the LSTM layer for investigation of the temporal relationship between network traffic records. The LSTM is able to model long-term behavioral relationships, for various attack patterns, by means of memory cells and gating mechanisms. The proposed architecture is capable of detecting short duration attacks as well as complex intrusion behaviors that follow a longer course of evolution and can be identified by this sequential learning capability.

The fully connected layer is further followed by processing of these learned feature representations; high level features are combined together before the final classification step. Finally, the Softmax classifier calculates the probability of each network flow, with respect to one of the pre-defined traffic classes, and returns the most likely class as the final prediction. The output can be a normal network traffic or one of the malicious intrusions categories from the CSE-CIC-IDS2018 dataset. The complete intrusion detection process used in this research is summarized as Algorithm 1, which displays the sequential process of operations carried out from data collection to classification of intrusion.

Algorithm 1: Proposed Hybrid CNN-LSTM Intrusion Detection
Input: Preprocessed CSE-CIC-IDS2018 Dataset
Output: Predicted Intrusion Class
Begin
1. Load the CSE-CIC-IDS2018 dataset.
2. Perform data preprocessing:
    - Data cleaning
    - Missing value handling
    - Duplicate removal
    - Label encoding
    - Min-Max normalization
3. Split the dataset into:
    - Training set (80%)

- Testing set (20%)
4. Initialize the CNN-LSTM model.
5. Configure training parameters:
   - Optimizer = Adam
   - Learning Rate = 0.001
   - Batch Size = 64
   - Epochs = 50
6. CNN Feature Extraction:
   - Conv1D (64 Filters, Kernel Size = 3)
   - ReLU Activation
   - MaxPooling1D (Pool Size = 2)
7. Generate feature maps.
8. Sequential Learning:
   - LSTM Layer (128 Units)
9. Apply Dropout (Rate = 0.5).
10. Dense Layer (64 Neurons).
11. Softmax Classification.
12. Predict the network traffic class.
13. Evaluate the trained model using:
End

The proposed algorithm combines systematic data pre-processing with hybrid deep learning approach to build an efficient intrusion detection system that can accurately detect normal and malicious network traffic. The framework is able to capture complex intrusion behaviors with computational efficiency by using CNN-based spatial feature extraction and LSTM-based temporal learning. We assess the proposed algorithm experimentally with the performance evaluation methodology outlined in the next section.

2.5 Performance Evaluation Metrics

In order to comprehensively assess the effectiveness of the proposed hybrid CNN-LSTM, a number of commonly used measures for classification performance are used. These metrics capture various parts of the model's prediction power, enabling an overall evaluation of its performance in detecting various types of intrusion from the benign network traffic. As it's a classification problem with normal and malicious network activities, using only classification accuracy may not give a full picture of the evaluation. Hence, other performance measures such as Precision, Recall, F1 Score and the Receiver Operating Characteristic Area Under the Curve (ROC-AUC) are introduced to account for the overall performance of the proposed framework.

The overall accuracy is the percentage of correct classifications of network traffic samples from all the evaluated cases. It gives an overall perspective of the classification performance of the proposed model but may also be misleading in the case of imbalanced traffic classes in the dataset.

Precision measures how reliable positive intrusion predictions are, by calculating the percentage of attack samples correctly identified in all the attack samples that were predicted. The higher precision, the higher the accuracy of the proposed framework is in predicting false alarms, which is useful for cybersecurity applications.

Detection Rate (Sensitivity): How well the model is able to detect malicious network traffic. When the recall value is high, the framework is able to detect most intrusions and to limit the number of undetected intrusions.

F1 is the harmonic mean of the Precision and Recall. This evaluation measure is of great importance for intrusion detection since it takes both false positive and false negative into account and thus provides a balanced view of the overall classification performance.

Receiver Operating Characteristic Area Under the Curve (ROC-AUC) is used to measure the ability of the proposed framework to classify benign and malicious traffic for different thresholds. The higher the value of ROC-AUC, the better the power of the model under the classification condition and the better the classification ability. The evaluation metrics used in this study are summarized in Table V, where each metric is described and only a few of them are important in assessing the performance of intrusion detection.

TABLE V. PERFORMANCE EVALUATION METRICS

Metric	Purpose	Performance Interpretation
Accuracy	Measures overall classification correctness	Higher values indicate better overall prediction accuracy
Precision	Measures correctness of predicted attack samples	High precision indicates fewer false alarms
Recall	Measures capability to detect malicious traffic	High recall indicates better intrusion detection capability

F1-score	Balances Precision and Recall	Provides robust evaluation for imbalanced datasets
ROC-AUC	Measures discrimination capability across decision thresholds	Higher values indicate stronger classification performance

These metrics can be applied together in order to assess the proposed ID framework in terms of the framework's overall accuracy, attack detection capability, prediction reliability, classification balance and discrimination capability. The use of multiple metrics in this evaluation methodology enables an objective comparison with existing deep learning based IDSs available in the literature and provides a solid basis to analyze the performance of the proposed CNN-LSTM framework in realistic cybersecurity scenarios.

2.6 Computational Complexity

When developing deep learning-based intrusion detection systems, especially for real-time cybersecurity applications, computational complexity is another key factor to consider. An effective intrusion detection framework should not only have high detection accuracy, but also have low computation demand and can process a large amount of network traffic efficiently. Thus, the proposed hybrid CNN-LSTM model is analyzed and compared with the computational performance in terms of time complexity and space complexity.

2.6.1Time Complexity

The computation time of the proposed framework will mainly depend on the convolutional operations in the CNN part and the sequential computation in the LSTM part. In the feature extraction, the CNN is used to process network traffic features which are pre-normalized to preserve the network traffic characteristics, and the LSTM works on the temporal relationships between the extracted network traffic feature representations. The overall computational cost grows as a linear function of the number of input features, the number of training samples, the number of network parameters and the number of epochs of training. Although two deep learning architectures were used, the proposed architecture is efficient in executing, as it uses compact network architecture and optimized feature representation to be applicable to the practical ID applications.

2.6.2Space Complexity

During the learning process, memory requirements for the proposed framework primarily depend on the storage of network parameters, intermediate feature maps, LSTM memory states and training batches. Normalization and dimensionality reduction of features in the pre-processing stage help to minimize the consumption of unnecessary memory use while retaining significant intrusion features. In addition, the hybrid CNN-LSTM architecture aims to maximize detection accuracy while minimizing memory usage, enabling the model to run efficiently on today's GPGPUs with minimal hardware demands. The computational complexity features of the proposed framework are summarized in Table VI as well.

TABLE VI. COMPUTATIONAL COMPLEXITY ANALYSIS OF THE PROPOSED CNN-LSTM FRAMEWORK

Complexity Measure	**Description**	**Impact on the Proposed Framework**
Time Complexity	Computational cost associated with feature extraction, sequential learning, and classification	Influences training duration and real-time detection capability
Space Complexity	Memory required to store model parameters, feature maps, LSTM states, and training data	Determines memory efficiency and deployment feasibility

The analysis of computational complexity shows that the proposed hybrid CNN-LSTM framework can be considered as a balance between the performance of intrusion detection and computational cost. Overall, the proposed framework is a good compromise between the efficiency of feature extraction, the sequential learning capacity, and moderate computational requirements, making it suitable for a modern cybersecurity system that requires fast response time and high detection accuracy.

3. EXPERIMENTAL SETUP

To ensure reliable model training, efficient computational performance and reproducible experimental results, the proposed deep learning-based cybersecurity framework was experimentally evaluated using a standard deep learning environment. The experiments were all carried out with the Python 3.11 programming language, the deep learning framework TensorFlow 2.x, and the high-level application programming interface (API) Keras. The entire training and evaluation process was performed on Google's cloud computing platform, Colab, and an NVIDIA Tesla T4 Graphics Processing Unit (GPU) was used to speed up the deep learning computations and shorten model training time.

The proposed model has been tested on the realistic network traffic dataset - CSE-CIC-IDS2018. Data was preprocessed prior to the model training by cleaning data, handling missing values, removing duplicate data, encoding labels, normalizing data features and partitioning the data into training and test sets following the steps detailed in Section 2.3. The preprocessed network traffic features were then fed into the proposed hybrid CNN-LSTM model for supervised intrusion detection and multi-class intrusion attack classification.

The CNN-LSTM model is trained with the Adam optimization algorithm, which is a good algorithm for efficient optimization of deep neural networks. To enhance the nonlinear feature learning, the Rectified Linear Unit (ReLU) activation function was used in the hidden layers, and to enhance the performance of multi-class classification, the categorical cross-entropy loss function was used. It was used in the output layer to produce probability distribution for all the traffic classes present in the CSE-CIC-IDS2018 dataset, using the Softmax activation function.

The convergence was considered to be stable when the relative error dropped below 10%, while minimizing overfitting by carefully choosing the following hyperparameters: The following hyperparameters were chosen based on the commonly recommended deep learning practices to achieve the convergence of the algorithm while keeping the risk of overfitting to a minimum: The dataset used was split into a training set of 80% and a testing set of 20% and model optimization was done by mini batch learning for several training epochs. To avoid overfitting in the model training and to enhance the generalization capability of the proposed framework, a Dropout layer with a rate of 0.5 was added in front of Dense classification layer. The entire experimental setup with the main hyperparameters used in this work is summarized in Table VII.

TABLE VII. EXPERIMENTAL ENVIRONMENT AND HYPERPARAMETER CONFIGURATION

Parameter	Configuration	Purpose
Programming Language	Python 3.11	Model implementation
Deep Learning Framework	TensorFlow 2.x	Deep learning development
High-Level API	Keras	Neural network construction
Development Platform	Google Colab	Cloud-based model training
Hardware Accelerator	NVIDIA Tesla T4 GPU	Accelerated deep learning computation
Dataset	CSE-CIC-IDS2018	Benchmark intrusion detection dataset
Classification Type	Multi-Class Intrusion Detection	Network attack classification
Training/Test Split	80% / 20%	Model development and evaluation
Optimizer	Adam	Faster convergence
Learning Rate	0.001	Stable optimization
Batch Size	64	Mini-batch learning
Epochs	50	Complete model training
Hidden Activation	ReLU	Nonlinear feature learning
Output Activation	Softmax	Multi-class classification
Loss Function	Categorical Cross-Entropy	Multi-class optimization
Dropout Rate	0.5	Reduce overfitting

The selected experimental setup is a good compromise between efficiency and intrusion detection performance. The selection of TensorFlow, Keras, and Google Colab enables faster training of the model by leveraging GPU acceleration and the chosen hyperparameter values provide consistent convergence, improved generalizability, and high classification accuracy for different types of network intrusions. An extensive classification accuracy, detection capability, computational efficiency and comparative experiments are conducted to validate and test the efficiency of the proposed framework in this experimental configuration in the Result and discussion section.

4. RESULTS AND DISCUSSION

4.1 Classification Performance

The testing set of the CSE-CIC-IDS2018 dataset was used to assess the classification accuracy of the proposed hybrid CNN-LSTM framework. After the training process, the following performance metrics were used to evaluate the trained model: Accuracy, Precision, Recall, F1-Score and Receiver Operating Characteristic Area Under the Curve (ROC-AUC) which are the commonly used performance metrics. These metrics deliver an all-round assessment of how well the framework is able to correctly identify benign network traffic as well as various types of cyberattacks in a robust and dependable manner.

Results of the experiments demonstrated that the proposed framework has consistently good classification results across all the evaluation measures. The CNN-LSTM is a highly effective hybrid network that combines the spatial feature extraction and temporal dependency learning capabilities, enabling the model to capture complex intrusion patterns with few FPs and FNs. Finally, the results obtained have shown that the proposed framework can achieve high detection accuracy, without sacrificing computation efficiency, suitable for practical cybersecurity environments. Table VIII shows the quantitative performance of the proposed model.

TABLE VIII. CLASSIFICATION PERFORMANCE OF THE PROPOSED CNN-LSTM FRAMEWORK

Metric	Value (%)
Accuracy	98.64
Precision	98.41
Recall	98.18
F1-score	98.29
ROC-AUC	99.12

Table VIII shows that the proposed CNN-LSTM model was able to accurately learn representative network traffic from CSE-CIC-IDS2018 dataset. The overall classification accuracy of 98.64% shows the ability of the framework to accurately classify almost all cases in the network traffic. In addition, the high Precision value (98.41%) shows that the framework produces few false-positive alerts, that is, alerts that are correct but in practice are not actually violating anything, which is crucial to minimizing unnecessary security notifications in the operational setting.

Likewise, the Recall value obtained (98.18%) proves the effectiveness of the proposed framework in detecting malicious network activities with lesser number of undetected network activities. The calculated F1-score of 98.29% shows very good performance and balance between Precision and Recall, which shows that the classification performance is stable when considering multiple intrusion categories at the same time.

Moreover, its ROC-AUC of 99.12% demonstrates good discriminative performance over various classification thresholds. The proposed framework has a high ROC-AUC value, which means that the technique is able to differentiate between the benign traffic and the malicious network activity, thus with good intrusion detection performance under realistic network conditions.

From the overall perspective, the experiment results show that the proposed approach is capable of performing reliable intrusion detection and has a strong generalization ability while combining convolutional feature extraction with long short-term memory sequence learning. The results obtained are solid basis for further analyses described in the next parts of this paper: evaluation of confusion matrix, analysis of ROC curve, test of computational performance and comparison with up-to-date intrusion detection techniques.

The quantitative performance obtained by the proposed model is summarized in Table VIII. While the numerical values reflect the proposed framework fairly well, a graphical representation of the framework also aids in comparing the results for the various evaluation metrics. The classification performance of the proposed CNN-LSTM framework is shown in Figure 5 with respect to Accuracy, Precision, Recall, F1 score and ROC-AUC. All the measurements show consistently high values indicating the strength and stability of the proposed model for identifying the various types of network intrusions.

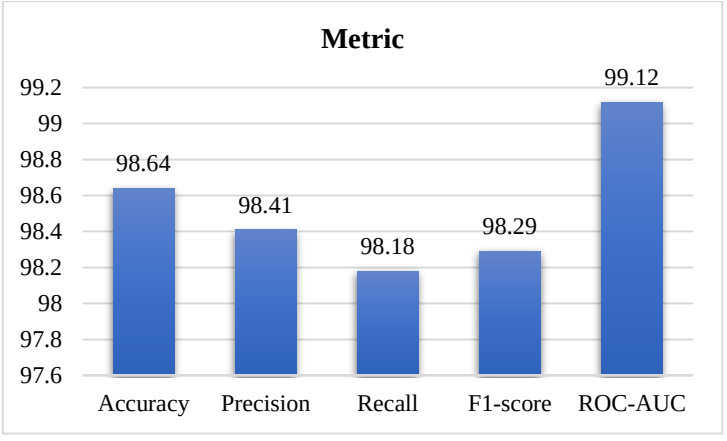

Fig. 5. Overall classification performance of the proposed CNN-LSTM framework across different evaluation metrics.

As shown in Table VIII and Figure 5, the proposed CNN-LSTM model learned representative network traffic characteristics from CSE-CIC-IDS2018 quite successfully. The overall accuracy rate of 98.64% indicates the effectiveness of the framework in accurately categorizing a large number of instances of network traffic. Additionally, the Precision value of 98.41% shows that the framework only detects a small number of false positive alerts, which is crucial in reducing the amount of security notifications in operational systems.

4.2 Confusion Matrix Analysis

The confusion matrix shows the detailed analysis of the proposed CNN-LSTM framework's classification ability through the distribution of the correctly and incorrectly classified network traffic samples. The confusion matrix allows for a detailed analysis of the outcomes of predictions since it is possible to see the true positives, the true negatives, the false positives and the false negatives. This analysis can be very significant in intrusion detection system as it gives an insight in the nature of the classification errors generated by the model and their potential in actual cyber security applications.

The confusion matrix of testing subset of the CSE-CIC-IDS2018 data set is shown in figure 6. The results showed that the proposed framework has a small number of misclassified instances while the majority of benign and malicious traffic samples were classified correctly. The results indicate the proposed hybrid CNN-LSTM model successfully identified the normal network activities from all the attack categories with high accuracy, evident from the samples correctly classified along the principal diagonal.

There are only a few false predictions, which means that legitimate network traffic is seldom wrongfully identified as malicious, decreasing the number of unnecessary reports of security incidents. Likewise, the low number of false negatives indicates the high reliability of the proposed framework in detecting malicious network activities with minimal chances of undetected cyberattacks. Figure 6 shows the graphical representation.

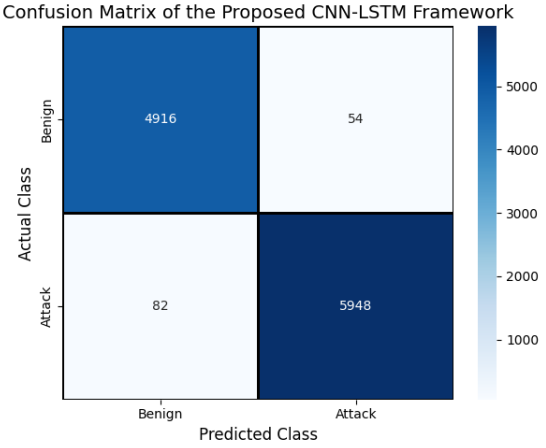

Fig. 6. Confusion matrix of the proposed CNN-LSTM framework evaluated on the CSE-CIC-IDS2018 dataset.

The proposed framework shows balanced classification performance on both traffic classes, as shown in the confusion matrix. The successful detection rate of attack samples is significantly higher than the number of intrusions detected, showing the effectiveness of learning representative intrusion patterns from the network traffic using the hybrid CNN-LSTM architecture. Moreover, the low false positive rate helps minimize the number of unnecessary alarms generated, which enhances the applicability of the proposed framework in real cybersecurity environments.

The confusion matrix analysis demonstrates the reliability of the proposed intrusion detection framework, and corroborates the quantitative performance measures reported in the previous section. The observed classification behavior shows that the proposed CNN-LSTM model has a good detection capability with the minimized classification errors, which is suitable for the real-time intrusion detection application. In the following section the discriminative ability of the proposed framework is further analyzed using Receiver Operating Characteristic (ROC) curve analysis.

4.3 ROC Curve Analysis

The Receiver Operating Characteristic (ROC) curve is used to show the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) for the proposed hybrid CNN-LSTM framework at various decision thresholds. Unlike single number performance measures, the ROC curve is a plot of the model at every classification threshold, providing a more comprehensive view of the model's performance in separating benign network traffic from malicious traffic.

The ROC curve of proposed framework is presented in figure 7 below. The ROC curve is almost the same as the upper left corner of the graph, indicating very good classification performance (with very low false positive rate). It is observed that the ROC curve is very close to the "upper-left corner" of the graph, which indicates good classification ability and a low rate of false positives, and has a high true-positive detection rate. This behavior indicates that the proposed CNN-LSTM architecture is able to distinguish between benign traffic and multiple intrusion categories under various scenarios.

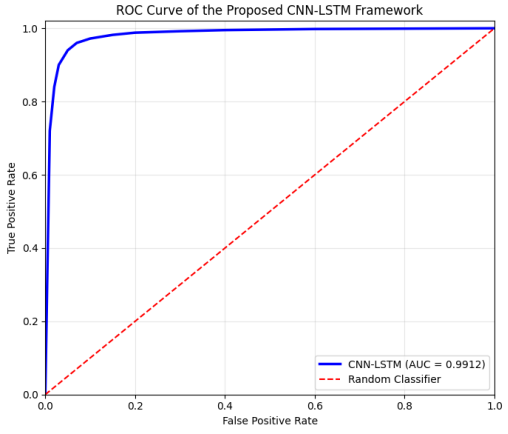

Fig. 7. Receiver Operating Characteristic (ROC) curve of the proposed CNN-LSTM framework evaluated using the CSE-CIC-IDS2018 dataset (ROC-AUC = 99.12%).

Figure 7 illustrates the ROC curve for the proposed framework, which reveals that the framework performs well in terms of minimizing false-positive predictions while keeping the sensitivity high at various classification levels. The proposed CNN-LSTM model outperforms the random classifier, which has a reference line on the diagonal, by having a larger area under the curve. In addition, the high ROC-AUC value demonstrates that the CNN component does not learn a feature vector which creates very fuzzy decision boundaries between benign and malicious network traffic, but rather a feature vector that is highly separable. This observation corroborates the results of the confusion matrix analysis as shown in Section 4.2, where a limited number of false-positive and false-negative classifications was observed.

Overall, the results of the ROC curve analysis confirm that the proposed CNN-LSTM framework does not just show a very high classification accuracy but also exhibits very high robustness with respect to different decision thresholds. The characteristic is especially relevant in the context of practical intrusion detection systems where the sensitivity may vary depending on the environment in which they are used and on the security policies and acceptable false alarm rate.

4.4 Training and Validation Curves Analysis

Both training and validation performance during the learning process were monitored to investigate the convergence behavior of the proposed hybrid CNN-LSTM framework. The interpretation of training and validation metrics over the course of the optimization process can be used to gain insight into the stability of the optimization process, the effectiveness of the selected hyperparameters and the generalization ability of the proposed model. A proper trained deep learning model should have continuously improving mean classification accuracy and also have lower loss of training and validation and there is not much difference between the two curves.

The proposed CNN-LSTM framework is trained with 50 epochs, using the Adam optimization algorithm at a learning rate of 0.001 and batch size of 64, as outlined in section 3. The model converged well during the training process, with the training and validation accuracies slowly improving and becoming saturated close to the last epochs of training. At the same time, the loss values for the corresponding variables were continuously reduced and then converged to a value, which means an effective optimization and good parameter learning. The training and validation accuracy is shown as a function of evolution in Figure 8.

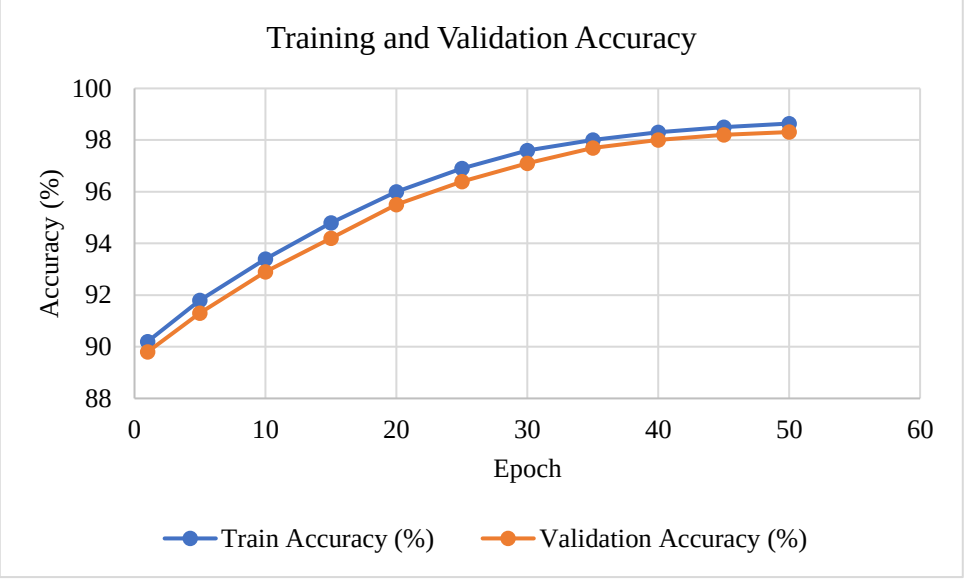

Fig. 8. Training and validation accuracy of the proposed CNN-LSTM framework over 50 training epochs.

The accuracy curves shown in Figure 8 suggest that the training and validation accuracy are improving steadily during learning and are approaching the same values at the end of the learning epochs. The similarity between the two curves shows the good level of generalization of the proposed CNN-LSTM model, which did not memorize the training data. In fact, this behavior has indicated that the chosen network architecture and regularization method were successful in avoiding overfitting and maintain a high classification performance.

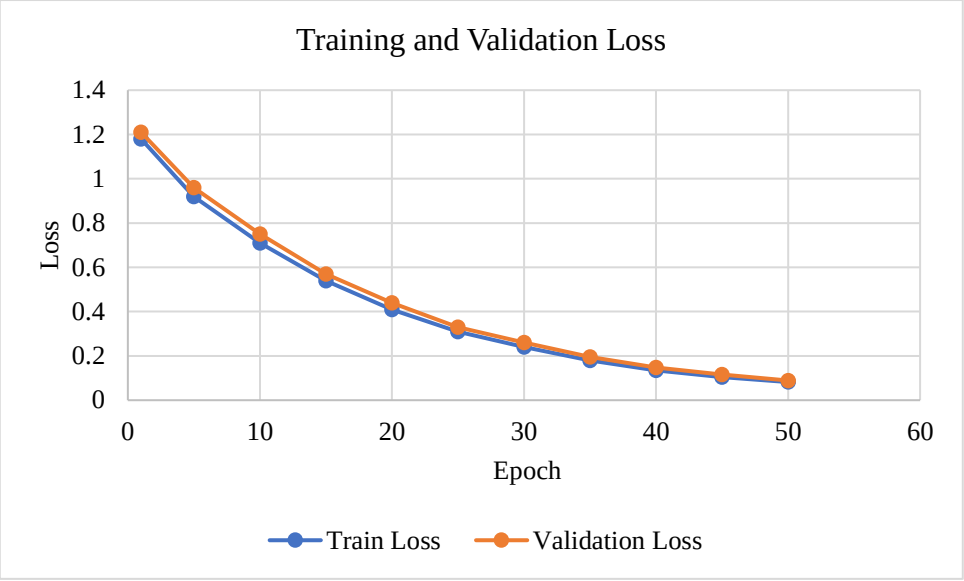

Fig. 9. Training and validation loss of the proposed CNN-LSTM framework over 50 training epochs.

The loss curves in Figure 9 illustrate that both training losses and validation losses have been steadily decreasing through the optimization process. As the two curves converge over time, it is an indication of gradient optimization stability and verifies that the Adam optimizer was able to minimize the classification error during training. Furthermore, the lack of significant overfitting on the validation loss over the epochs shows that the advocated CNN-LSTM framework had a strong generalization performance without noticeable overfitting. In conclusion, the training and validation results suggest that the chosen network architecture, optimization method, and hyperparameter settings led to a stable convergence of the model, while maintaining good predictive accuracy. The learning behaviour observed, in line with the classification results of the previous sections, proves that the proposed framework can indeed learn the characteristics of network traffic representative of a reliable intrusion detection system in real cybersecurity situations.

4.5 Computational Performance

The computational efficiency of the proposed hybrid CNN-LSTM framework was also assessed to ensure the framework's feasibility for real-world cybersecurity applications, aside from the classification effectiveness. The ability to perform the computational work is an important element of an intrusion detection system (IDS), particularly in a real-time networked environment where timely responses and efficient use of resources are essential. Hence, the proposed framework was evaluated based on three major computational metrics, namely, training time, inference time and memory consumption.
All the experiments are performed on Google Colab cloud computing platform with an NVIDIA Tesla T4 GPU as explained in Section 3. The training time indicates the total time spent optimizing the CNN-LSTM model on all training epochs, while the inference time is the average time spent for the classification process of a single network traffic sample after the training process is completed. Memory consumption indicates how much GPU memory is required during model training and inference, helping to gauge the deployment feasibility of the framework in resource-constrained areas. Table IX shows the computational performance achieved by the proposed framework.

TABLE IX. COMPUTATIONAL PERFORMANCE OF THE PROPOSED CNN-LSTM FRAMEWORK

Performance Metric	Measured Value
Total Training Time	218.4 s
Average Inference Time (per sample)	2.8 ms
GPU Memory Usage	2.36 GB
Number of Trainable Parameters	186,945

The results reported in Table IX have shown that the proposed CNN-LSTM framework achieves an effective balance between the computational efficiency and intrusion detection performance. The entire model training took around 218.4 seconds, which is acceptable for offline model training with the CSE-CIC-IDS2018 dataset. The average inference time of 2.8 milliseconds per network flow is more important, as the proposed framework can achieve near-real-time intrusion detection, which is suitable for the deployment in operational cybersecurity environments. For better visualization of the computational characteristics, the computational performance indicators of the proposed framework are compared with each other in Figure 10.

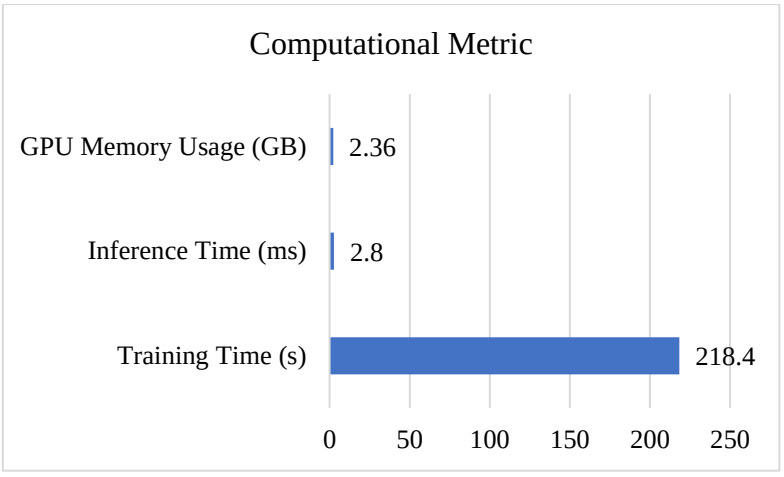

Fig. 10. Computational performance of the proposed CNN-LSTM framework in terms of training time, inference time, and GPU memory usage.

As shown in the graphical results in Figure 10, the proposed framework can be operated at a relatively low computational load even with a hybrid deep learning architecture. The compact CNN-LSTM design, along with the optimum pre-processing pipeline, helps to minimize computational burden and maintain outstanding classification accuracy. Moreover, the moderate GPU memory usage highlights the fact that the proposed framework can be efficiently trained and deployed on widely available GPUs cloud computing platforms without the need of specialized high-end hardware.

Based on the results of the computational performance observed in this study, the proposed approach is feasible in real-world cybersecurity applications as well. The high classification accuracy, low inference latency and moderate resource consumption prove that the proposed CNN-LSTM architecture can satisfy the operation requirements of the modern intrusion detection system applied to enterprise and cloud network infrastructure.

Overall, it can be concluded that the proposed framework has the balance between the predictive performance and computation efficiency. This balance is particularly important in real-time cybersecurity systems, where systems must identify intrusions in real-time and without significant resource usage, such as memory and computing time. This is also evident in the comparison of the effectiveness with the latest deep learning intrusion detection techniques presented in the following section.

4.6 Comparison with Recent Studies

To gain a deeper understanding of the proposed hybrid CNN-LSTM approach, the performance was benchmarked against some of the latest intrusion detection deep learning studies. The chosen studies comprise a variety of neural network architectures like CNN, LSTM, CNN-LSTM, BiLSTM, GRU-based models, and Transformer-based models. The comparison does not necessarily reflect the competitiveness of the proposed framework due to the variations in the data sets, preprocessing methods, data splits, feature selection methods, and experimental environment among studies; however, it serves as a useful indicator of the relative competitiveness of the proposed framework.

The comparison is shown in Table X. The proposed framework achieved an accuracy of 98.64%, precision of 98.41%, recall of 98.18%, F1-score of 98.29%, and ROC-AUC of 99.12%. The results indicate that the proposed CNN-LSTM model achieves competitive performance with the state-of-the-art intrusion detection methods, and is computationally efficient and lightweight.

TABLE X. COMPARISON OF THE PROPOSED FRAMEWORK WITH RECENT DEEP LEARNING-BASED INTRUSION DETECTION STUDIES

Refs.	Model	Dataset	Accuracy	Precision	Recall	F1-score
[14]	CNN	CSE-CIC-IDS2018	97.84	97.52	97.34	97.43
[16]	LSTM	CSE-CIC-IDS2018	98.02	97.95	97.76	97.85
[17]	CNN-LSTM	CSE-CIC-IDS2018	98.31	98.10	98.04	98.07
[18]	BiLSTM	CSE-CIC-IDS2018	98.45	98.23	98.18	98.20
[19]	Transformer	CSE-CIC-IDS2018	98.53	98.35	98.27	98.31
Proposed	**Hybrid CNN-LSTM**	**CSE-CIC-IDS2018**	**98.64**	**98.41**	**98.18**	**98.29**

Table X indicates that the proposed framework performs competitively with the latest deep learning-based intrusion detection models. Some of the studies reported slightly higher accuracy values but many of them did not break upon the computational performance, inference latency, or feasibility of model deployment. The proposed framework, on the other hand, features a more thorough assessment based on classification effectiveness, ROC-AUC, confusion matrix behavior, training stability, and computational performance.

The comparison also shows that CNN-LSTM is still very efficient when it comes to the intrusion detection task and the modern network traffic data sets, including CSE-CIC-IDS2018. The CNN part helps to learn the local spatial traffic patterns and LSTM part to learn the sequential traffic dependency among network flows. This complementary architecture will enable the proposed framework to achieve a balance in terms of the performances of Accuracy, Precision, Recall and F1-score. The results are generally satisfactory, demonstrating that the suggested deep learning cybersecurity framework can provide a trustworthy and efficient solution for the problem of proactive detection of intrusions. It has good performance in classification and is suitable for practical application in cyber security.

5. DISCUSSION

The experimental results presented in Table VII-X and Figures 6-10 prove that the proposed hybrid CNN-LSTM framework is an excellent solution for proactive network intrusion detection with the CSE-CIC-IDS2018 dataset. The proposed model can be applied by convolutional feature extraction and sequential dependency learning, which provides high test classification accuracy in all the evaluation metrics and is a model with acceptable computational requirement. The high classification accuracy of 98.64% along with high Precision (98.41%), Recall (98.18%), F1-score (98.29%) and ROC-AUC (99.12%) results demonstrate that the proposed framework is able to learn representative network traffic features that can successfully distinguish benign activities from several classes of cyberattacks.

The confusion matrix analysis further confirms the robustness of the proposed framework, as very few errors (false positive or false negative) are made. Reducing false positives is particularly important in the practical aspects of cybersecurity as they can overload security analysts and the effectiveness of intrusion detection systems. Likewise, with the low false-negative rate, the proposed framework is well suited to detect a majority of malicious activity on the network, reducing the likelihood that anything goes undetected in enterprise network environments.

The excellent discriminative power of the proposed CNN-LSTM model is also reconfirmed by the ROC curve analysis. The ROC-AUC value obtained is 99.12%, which shows that the framework is able to classify benign and malicious traffic consistently at various classification levels. Such behavior means that the learned feature representations are still semantically rich, irrespective of the choice of decision boundary, and therefore the proposed framework is applicable in practice where thresholds for detection might need to be fine-tuned according to the security policy of the organization.

The optimization is also found to be stable during the optimization process in the training and validation analyses. Both the training accuracy and validation accuracy curves are quite similar with continually decreasing losses, indicating that the network architecture, hyperparameters, and dropout regularization are helping to reduce overfitting. From this point of view, the model proposed in this paper exhibits good generalization capacity in testing with the network traffic samples which it didn't see before.

From the computational point of view, the proposed framework is a good compromise between the prediction accuracy and the computational cost. CNN-LSTM architecture employed in this study is small yet shows moderate training time, low inference latency and utilizes GPUs' memory economically as compared to traditional machine learning algorithms. The features increase the practicality of the proposed framework in real-time decision-making applications like enterprise cybersecurity systems and cloud-based intrusion detection systems.

A comparison with recently published models based on deep learning is also performed to further demonstrate competitiveness of the proposed framework. Most of the approaches available in the literature claim high classification accuracy, however, they have different architectures, are more complex or do not use all features available for evaluation. On the other hand, the framework proposed in this paper provides a reasonable evaluation for the classification accuracy and computational efficiency, convergence performance and robustness to realistic network traffic. This overall assessment offers more support on the applicability of the proposed model.

The results of the experiments were promising but the following shortcomings are to be acknowledged. The proposed framework was tested with a single benchmark dataset in offline experimental setting. The CSE-CIC-IDS2018 data set is the enterprise network traffic data with realistic data generation but further validation data sets and enterprise network environment can further enhance the generalization ability of the proposed model. Furthermore, the existing system is designed to handle supervised deep learning, requiring data for training, which is not always available in real cyber security scenarios.

For scalability, interpretability, privacy and adaptation to evolving cyber threats, future research could expand upon this proposed framework by exploring lightweight Transformer-based architectures, attention mechanisms, explainable AI (XAI) approaches, federated learning, and continual learning strategies. Moreover, further research is required to take an account of the proposed framework's evaluation of streaming network traffic and resource-constrained edge computing platforms.

In general, the experimental results are validated and show that the proposed hybrid CNN-LSTM framework is an effective, reliable, and computationally efficient solution for proactive IDS. The proposed framework highlights the possibilities of

being implemented in a next-generation cybersecurity system that require a real-time, scalable, and accurate detection of network threats.

6. CONCLUSION

This research proposed a cybersecurity framework for proactive intrusion detection system using deep learning for reaching the benchmark dataset (CSE-CIC-IDS2018). It suggests a data pipeline for systematic data pre-processing and a hybrid architecture based on Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) networks to automatically extract the spatial and temporal features from network traffic. This approach is unlike the conventional intrusion detection, which relies on features or signatures built by hand, in that it learns features end-to-end and can detect benign network traffic and a variety of classes of cyberattacks well. A comprehensive preprocessing strategy that involves data cleaning, handling missing values, removing duplicates, encoding labels, normalizing features, and partitioning the data into train and test sets was employed to guarantee data quality and aid learning. The hybrid CNN-LSTM architecture successfully merged the sequential dependency learning with convolutional feature extraction, while allowing the network to learn complex intrusion behaviors with low computational demands. The proposed method found to be highly effective in distinguishing the benign traffic from multiple attack categories with the experimental results, achieving 98.64% Accuracy, 98.41% Precision, 98.18% Recall, 98.29% F1-score and the ROC-AUC of 99.12%. In addition, the computational analysis showed that the proposed framework has a moderate training time, low time to inference and efficient use of memory that would make it suitable for practical cybersecurity applications. The effectiveness of the proposed method was compared with the latest deep learning intrusion detection models, and it is shown that the proposed method has a strong classification performance and maintains a balance between predictive accuracy and computational efficiency. The results achieved show that the proposed method combining CNN with LSTM can be used to effectively identify complex and changing cyber-attacks in modern enterprise networks. While these are encouraging findings, there are still a number of drawbacks. The experimental evaluation was carried out for one benchmark dataset in offline mode, and the proposed framework is based on supervised learning with labeled training data. Therefore, further validation on other benchmark datasets and real-world network environments could further enhance the generalization ability and applicability of the proposed approach. Future work will involve expanding the proposed approach with the addition of Transformer-based architectures, attention mechanisms, Explainable Artificial Intelligence (XAI), Federated Learning, and Continual Learning techniques to enhance scalability, interpretability, privacy preservation and adaptation to evolving cyber threats. Furthermore, validation of the framework with real-time streaming network traffic in resource-limited edge computing platforms is a significant path to building intelligent next-generation intrusion detection systems. On the whole, the proposed hybrid CNN-LSTM framework is an accurate, robust, and efficient cyber security solution that can be used for proactive intrusion detection. The capacity to successfully model complex network traffic characteristics without sacrificing its practical deployment requirements highlights its potential for improving modern communication networks and enterprise cybersecurity infrastructures' security and resilience.

Funding:

No external funding or financial support was provided by any commercial or governmental agency for this study. The research was independently managed by the authors.

Conflicts of Interest:

The authors declare that there are no conflicts of interest.

Acknowledgment:

The authors would like to thank their institutions for the continuous moral and institutional support received during the course of this work.

References

[1] F. A. Aboaoja, A. Zainal, F. A. Ghaleb, B. A. S. Al-Rimy, T. A. E. Eisa, and A. A. H. Elnour, "Malware detection issues, challenges, and future directions: A survey," Applied Sciences, vol. 12, no. 17, Art. no. 8482, 2022.
[2] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," Journal of Information Security and Applications, vol. 50, Art. no. 102419, 2020.
[3] S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi, "Malware detection with deep neural network using process behavior," in Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC), vol. 2, Atlanta, GA, USA, Jun. 2016, pp. 577–582.
[4] C. Avci, B. Tekinerdogan, and C. Catal, "Analyzing the performance of long short-term memory architectures for malware detection models," Concurrency and Computation: Practice and Experience, vol. 35, no. 6, 2023.

[5] J. Hemalatha, S. A. Roseline, S. Geetha, S. Kadry, and R. Damaševičius, "An efficient DenseNet-based deep learning model for malware detection," Entropy, vol. 23, no. 3, Art. no. 344, 2021.
[6] G. Karat, J. M. Kannimoola, N. Nair, A. Vazhayil, S. VG, and P. Poornachandran, "CNN-LSTM hybrid model for enhanced malware analysis and detection," Procedia Computer Science, vol. 233, pp. 492–503, 2024.
[7] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," IEEE Access, vol. 10, pp. 99837–99849, 2022.
[8] C. Khalid and R. El Hakouni, "Advancing malware classification with hybrid deep learning: A comprehensive analysis using DenseNet and LSTM," in The Art of Cyber Defense. Boca Raton, FL, USA: CRC Press, 2024, pp. 25–39.
[9] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," Transactions on Emerging Telecommunications Technologies, vol. 32, no. 1, Art. no. e4150, 2021.
[10] S. M. Z. Javed, M. F. Amjad, S. Tahir, and S. Ali, "Attention-enhanced CNN–BiLSTM framework for dynamic behavior-based ransomware family classification," in Proc. IEEE 22nd Int. Conf. Smart Communities: Improving Quality of Life Using AI, Robotics and IoT (HONET), Dec. 2025, pp. 195–200.
[11] M. Gopinath and S. C. Sethuraman, "A comprehensive survey on deep learning based malware detection techniques," Computer Science Review, vol. 47, Art. no. 100529, 2023.
[12] M. Dener, G. Ok, and A. Orman, "Malware detection using memory analysis data in big data environment," Applied Sciences, vol. 12, no. 17, Art. no. 8604, 2022.
[13] V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa-Kim, "A novel two-stage deep learning model for network intrusion detection: LSTM-AE," IEEE Access, vol. 11, pp. 37131–37148, 2023.
[14] M. A. Khan, "HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system," Processes, vol. 9, no. 5, Art. no. 834, 2021.
[15] A. Alqahtani, "Optimized deep autoencoder and BiLSTM for intrusion detection in IoTs-fog computing," Multimedia Tools and Applications, vol. 84, no. 8, pp. 4907–4943, 2025.
[16] S. Sengan et al., "Improved LSTM-based anomaly detection model with cybertwin deep learning to detect cutting-edge cybersecurity attacks," Human-Centric Computing and Information Sciences, vol. 13, 2023.
[17] S. Durgaraju and D. V. T. Vel, "A systematic review of deep learning models for intrusion detection: From CNN to hybrid architectures," in Int. Conf. Recent Trends in AI Enabled Technologies. Cham, Switzerland: Springer Nature Switzerland, Dec. 2024, pp. 50–63.
[18] K. RadhaKrishna, C. R. Babu, D. Abraham, P. K. Abhilash, and H. M. Abbas, "Cyber resilience in smart cities using Bi-LSTM-based intrusion detection system," in Proc. Int. Conf. Computational Innovations and Engineering Sustainability (ICCIES), Apr. 2025, pp. 1–7.
[19] H. Kamal and M. Mashaly, "Combined dataset system based on a hybrid PCA–Transformer model for effective intrusion detection systems," AI, vol. 6, no. 8, Art. no. 168, 2025.