Peer-to-Peer Networking and Applications (2025) 18:129
https://doi.org/10.1007/s12083-025-01944-7

Network security based combined CNN-RNN models for IoT intrusion detection system

Rahma Jablaoui[1] · **Noureddine Liouane**[1]

Received: 6 August 2024 / Accepted: 26 February 2025 / Published online: 26 March 2025

Abstract

Over the last few years, the rapidary growth of the Internet of Things (IoT), has significantly increases the intricacy of managing network security. As IoT networks broaden, they become progressively vulnerable to cyberattacks, requiring advanced threat detection and mitigation solutions. To address these challenges, Intrusion Detection System (IDS) is fundamental for identifying and alleviating potential menaces in IoT environments. This paper, investigates the use of Deep Learning (DL) techniques, which have proven to be h effective in detecting and classifying malicious network traffic. Hence, we propose an IDS architecture that combines Convolutional Neural Networks (CNN) used to extract spatial features with four variants of Recurrent Neural Networks (RNN) including Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), Gated Recurrent Unit (GRU), and Bidirectional GRU (BiGRU) utilized to capture temporal features. Together, these networks are able to anticipate and categorize malevolent cyberattacks in IoT network traffic more effectively than traditional methods. Experimental evaluation was conducted on the NF-UQ-NIDS, a comprehensive Netflow dataset that blends data from NF-BoT-IoT, NF-ToN-IoT, NF-UNSW-NB15, and NF-CSE-CIC-IDS2018 for binary classification. The efficacy of our proposed models was evaluated using various parameters such as accuracy, precision, recall, F1 score, False Alarm Rate (FAR) and Area Under Curve (AUC). A comprehensive comparative study was conducted that evaluated four proposed models in multiple datasets. The study underscores the potential of combining spatial and temporal DL models for advanced network security applications, demonstrating notable improvements in detection accuracy and reductions in false alarm rates. Consequently, the impressive results achieved by our models show the effectiveness of integrating CNN and different RNN variants for intrusion detection in IoT networks, offering a strong solution to protect IoT ecosystems against security threats.

Keywords DL · CNN · RNN · LSTM · BiLSTM · GRU · BiGRU · IDS · IoT · Netflow-based dataset

1 Introduction

The Internet of Things, or IoT, is a network of connected objects that interact with each other and exchange data on their own, without the need for human involvement, over the internet [1, 2]. With its ability to provide convenience, efficiency, and innovation in a variety of areas, including Smart Cities and Homes, Healthcare Monitoring, Industry, Business, Agriculture, Energy Management, and the Military, technology has become an indispensable part of our everyday life [3–11]. Forecasts suggest that by 2025, there will be more than 41 billion IoT devices, producing 79.4 zettabytes (ZB) of data according to the International Data Corporation (IDC) (https://infohub.delltechnologies.com/l/edge-to-core-and-the-internet-of-things-2/internet-of-things-and-dataplacement). Nonetheless, there are serious cybersecurity issues raised by this quick expansion. Hackers are constantly improving the methods they use to break into networks, and there are more opportunities for cyberattacks due to the sheer amount of data being transferred through various devices and protocols. Furthermore, malware attacks are becoming more complicated and large-scale, making identification and mitigation more difficult.

In order to address these problems, IDS are commonly recognized as crucial instruments for detecting and imped-

This article is part of Topical Collection: *4 - Track on IoT*
Guest Editor: Peter Langendoerfer

✉ Rahma Jablaoui
rahma.jablaoui@fsm.rnu.tn

Noureddine Liouane
noureddine.liouane@enim.rnu.tn

[1] Laboratory of Automatic, Signal, and Image Processing, University of Monastir, Monastir 5001, Tunisia

ing harmful actions in networks [12]. Traditional IDS approaches, while effective to some extent, often struggle to cope with the dynamic and complex nature of IoT environments [13]. DL approaches were recently included into IDS, which has greatly increased their intelligence and effectiveness in applications including image identification, computer vision, natural language processing, weather prediction, and behavior analysis. DL, an advanced subset of Machine Learning (ML) inspired by the neural networks found in the human brain, has demonstrated remarkable performance [14, 15]. Beyond the capabilities of conventional ML techniques, its deep computing powers offer a clear advantage in handling the complicated and diverse information typical of IoT scenarios. Convolutional Neural Networks (CNN), Deep Belief Networks (DBN), Deep Neural Networks (DNN), and Recurrent Neural Networks (RNN) are among the many varieties of deep neural networks that are frequently used in the field of intrusion detection [16]. These advanced networks present viable opportunities to enhance IDS capabilities in the ever-changing IoT security context.

This work introduces an enhanced IDS leveraging a DL model that merges CNN with RNN to effectively capture both spatial and temporal aspects in the data leading to a more robust classification of network traffic anomalies. In the realm of time series management, RNNs provide distinct advantages by effectively handling the retention of information from prior input within sequences. However, they face challenges such as the gradient vanishing issue. Researchers have devised several solutions to mitigate this, including well-known architectures like LSTM and its variant, BiLSTM, as well as GRU and its counterpart, BiGRU [17, 18]. To even further improve feature extraction, the suggested model includes a convolutional layer built with a 1D-CNN in addition to the RNN component. Therefore, by leveraging the strengths of the proposed architecture, our framework not only effectively captures the complex spatial and temporal dependencies within IoT network traffic and mitigates the gradient vanishing problem but also delivers superior detection performance, outperforming existing IDS solutions. The system is engineered to provide predictions for binary classification tasks, using contemporary, publicly accessible Netflow-based datasets like NF-BoT-IoT, NF-ToN-IoT, NF-UNSW-NB15, and NF-CSE-CIC-IDS2018, which together make up the NF-UQ-NIDS dataset. The system is designed to provide predictions for binary classification tasks. The reliability and efficacy of the established system are assessed through stringent training and testing processes. This article presents several key contributions.

- Development of a hybrid DL-based IDS that effectively identifies cyber threats in IoT networks.
- Enhancement of the feature extraction process by combining CNN and four RNN variants. The architecture includes a 1D-CNN-based convolutional layer followed by LSTM, BiLSTM, GRU, and BiGRU components, all designed with a consistent structure.
- Validation of the suggested models on the NF-UQ-NIDS dataset, a comprehensive collection that amalgamates four diverse databases: NF-BoT-IoT, NF-ToN-IoT, NF-UNSW-NB15, and NF-CSE-CIC-IDS2018. This dataset encompasses a wide array of network configurations and attack scenarios, providing a thorough testing ground.
- Execution of binary classification experiments to differentiate between attack samples and benign ones. The approach's efficacy is evaluated through a range of performance measures, and a comparative analysis with latest technologies is conducted to showcase the superiority of the suggested models.

The structure of the article is outlined in the subsequent parts: A brief summary of the literature is provided in Section 2, laying the groundwork for this research by summarizing current knowledge in the field. We present a description of DL models in Section 3, detailing the architectures of both the suggested CNN and RNN models. We also provide our proposed approach and designed architectures. Section 4 discusses the evaluation of experiments, including the Netflow-based datasets used, the metrics employed, and the model settings. Section 5 dives into the experimental setup and findings of our research. The study is concluded in Section 6, which provides a summary of the research's major findings and recommendations for possible future directions in the field.

2 Literature review

Given the increasing urgency to fortify cybersecurity defense s against intruders and cybercrimes, IDS has become paramount. A plethora of research endeavors have aimed to confront these challenges head-on by crafting resilient IDSs capable of vigilant monitoring of network traffic to detect and thwart malicious activities [19, 20]. DL has been thoroughly investigated in these works, with an emphasis on combining and integrating different DL algorithms to create high-performing IDS [21, 22]. This section summarizes a number of research that have used enhanced IDS based DL to investigate various facets of network traffic in IoT contexts.

For instance, by offering five NetFlow datasets derived from four well-known benchmark datasets: NF-BoT-IoT, NF-ToN-IoT, NF-UNSW-NB15, NF-CSE-CIC-IDS2018, and NF-UQ-NIDS. Sarhan et al. [23] have made a substantial contribution to the research community. The approach they use entails converting pcap files into NetFlow format, choosing 12 features, labeling the datasets, and assessing performance using an Extra Trees ensemble classifier in

both binary and multiclass classification experiments. In an additional work [24], Sarhan and associates expanded on this research by introducing a NetFlow-based feature set called UQ-NIDS-v2, which includes 43 features from the NIDS datasets UNSW-NB15, BoT-IoT, ToN-IoT, and CIC-CSE-IDS2018. The goal of this project was to assess and standardize traffic classifiers that use machine learning.

In [25], T. Altaf et al.'s investigation, authors introduce a new method for NIDS that makes use of graph neural networks (GNN). Their GNN-based NIDS, in contrast to conventional techniques, acts on multi-edges and multi-dimensional characteristics, allowing it to extract the complex structural elements of both legitimate and malignant behaviors. The model learns the graph topology and traffic patterns in complicated networks by concentrating on the full communication at the edge level among every pair of IoT nodes, using four well-known datasets including ToN IoT, BoT IoT, NF-ToN IoT, and NF-BoT IoT. The study carried out a thorough experimental analysis that was mainly focused on binary classification. The outstanding outcomes, which attain astonishingly excellent rates of accuracy that vary from 98.91% to 99.96% across all datasets, show the advantages of the suggested model.

Vishwakarma et al. [26] present a pioneering approach utilizing a 1D CNN model to detect and classify anomalies within IoT networks. To enhance the model's efficiency, transfer learning is employed, reducing both classification complexity and runtime requirements for multiclass and binary classification tasks. The experimental evaluation is comprehensive, covering a diverse range of datasets, including NF-ToN-IoT, NF-BoT-IoT, NF-CSE-CICIDS2018, NF-UNSW-NB15, NF-UQ-NIDS, CIC flowmeter-based IoT DS2, IoT Network Intrusion, MQTT-IoTIDS2020, and CIC-ToNIoT. Notably, the proposed model achieves exceptional accuracy and successfully identifies 20 different attack types. Additionally, verification is done on the model's real-time performance on a resource-constrained edge device, demonstrating its practical viability and outperforming existing methodologies.

In reference to [27], the study proposes two innovative Deep Convolutional Neural Network (DCNN) models, MyCNN and IoTCNN, specifically made to detect different types of intrusion threats, both harmless and malignant. Their method entailed using the NF-UNSW-NB15-v2 dataset, that includes nine different attack types, to train the models. They created Red, Green, and Blue (RGB) images from the network stream data in order to prepare it for training. These images were then fed into the neural network. The findings from experiments showed that the proposed DCNN models could detect a wide range of incursions, both malignant and benign, with high accuracy and with a significant reduction in computational burden in IoT networks. Interestingly, the accuracy of both models was higher, although the MyCNN model exhibited superior performance compared to the IoTCNN model.

According to [28], Santosh K. Smmarwar et al. presents an Explainable Artificial Intelligence (XAI)-based hybrid Android Malware Detection (AMD) system that integrates a Convolutional Neural Network (CNN) with a Bidirectional Gated Recurrent Unit (Bi-GRU) using deep learning models. Experiments were conducted on the CICAndMal2019 Android malware dataset, achieving an accuracy of 97.98%, along with precision, recall, and F1-score of 97.75%, 97.76%, and 97.75%, respectively, surpassing the performance of existing DL models.

The author of [29] describes an enhanced IDS based DL approach that combines CNN and BiGRU models. To address the challenge of imbalanced data, they use the ADASYN and RENN algorithms for data sampling and a hybrid technique that involves the RF algorithm with Pearson correlation analysis. CNN is used to extract spatial features. Whereas, BiGRU component specializes in capturing long-distance dependencies. Three datasets are used for the experimental evaluation: UNSW-NB15, NSL-KDD, and CIC-IDS2017. Notable accuracy levels of 85.55%, 99.81%, and 99.70%, respectively, are attained. These results reveal that the SRFCNN and BiGRU fusion perform better than the CNN-GRU model of the same data, demonstrating their potent feature extraction capabilities.

Likewise, Hnamte et al. [30] discuss the integration of CNN and BiLSTM networks into a unified model, named the DCNNBiLSTM model. This fusion is intended to capture both short and long-term dependencies in the data, greatly increasing detection capacities and classification accuracy. The two current traffic datasets used in the experimental evaluation are Edge-IIoT and CICIDS2018 on multiclass classification tasks. Using these datasets, the DCNNBiLSTM model achieves remarkably high accuracy rates reaching 100%.

Prabhat Kumar et al. [31] introduces an optimized and efficient ensemble learning-based Android malware detection framework. OEL-AMD addresses the limitations of existing malware detection methods by reducing false positives and improving detection rates. It employs statistical feature engineering and a Binary Grey Wolf Optimization-based selection method to refine feature sets (BGWO). By optimizing base learners, the framework achieves 96.95% accuracy for binary classification and 83.49% for category classification, using a publicly available android malware dataset (CICInvesAndMal2019). The results demonstrate the effectiveness of the proposed approach and compared with existing methods to evaluate their statistical significance.

Reference [32] explore the LS-DRNN method, which combines a number of methods to create a memory-efficient DL method for identifying botnet attacks in IoT networks. This technique combines SMOTE for class balance, DRNN

for efficient memory utilization, and Long Short-Term Memory Auto-encoder (LAE) for dimensionality reduction. The Bot-IoT dataset is used for the experimental evaluation, which highlights the method's superior performance over other techniques. The LS-DRNN model yields an amazing 86.49% reduction in memory space required for data storage, while still achieving strong classification performance in minority classes. Additionally, the LS-DRNN model shows notable improvements, in comparison to the DRNN model.

In their work, Yung-Chung Wang et al. [33] delve the use of DL approach in IDS. The study showcases remarkable accuracy exceeding 98% with individual algorithms including DNN, RNN, CNN, and LSTM. Unlike combined models CNN with RNN, these individual models are considered more suitable for implementation in IDS devices. The proposed approach not only enhances detection performance but also significantly reduces inference time, highlighting its potential for improving network security in a resource-efficient manner.

Paper [34] proposes an anomaly detection model called FlowGANAnomaly for detecting anomalous traffic in NIDS. FlowGANAnomaly streamlines anomaly detection with a generator-discriminator architecture, enhancing efficiency. It maps diverse traffic features into a unified space, improving anomaly identification. A novel anomaly scoring method integrates generator and discriminator outputs to boost recall rates. Experimental results on four public datasets (NSL-KDD, CIC-IDS2017, CIC-DDoS2019, and UNSW-NB15) show its superiority over existing ML and DL models for NIDS.

Study [35] leverages the DeepSecure framework for threat-hunting, which is a computational design science paradigm to develop an advanced IT solution. Authors employed a dynamic vector quantized variational autoencoder to uncover latent patterns within multivariate time series data. Building on this, a multiscale hierarchical attention-based bidirectional gated recurrent unit (Bi-GRU) mechanism is designed for threat detection. To enhance model interpretability, attention scores are visualized, providing deeper insights into decision-making processes. The proposed model is evaluated on the ToN-IoT and CSE-CIC-IDS2018 datasets, demonstrating its effectiveness in accurately identifying threats and improving the interpretability of complex models, helping organizations to respond rapidly to cyberattacks.

In summary, existing research has explored various DL approaches for intrusion detection in IoT networks. Building on these insights, our work proposes a CNN-RNN-based approach to enhance intrusion detection by leveraging the strengths of both architectures.

3 Proposed approach

3.1 Preliminary

3.1.1 Convolutional neural network (CNN)

The advent of CNN has sparked a revolution in DL, ushering in a new era of advancements in artificial intelligence [36]. CNNs have permeated diverse fields, such as computer vision, object detection, medical image analysis, security, surveillance, and video analysis, reshaping how these domains approach complex problems. Notably, CNNs have excelled in image classification tasks, achieving levels of accuracy that rival or even surpass human performance in certain contexts [37–40]. The architecture of a CNN is ingeniously crafted to autonomously extract and hierarchically learn spatial features from raw input data. This architecture is pivotal in addressing the challenge of parameter explosion by employing weight sharing, which significantly expedites the training process. As depicted in Fig. 1, CNNs comprise essential components, including convolutional layers

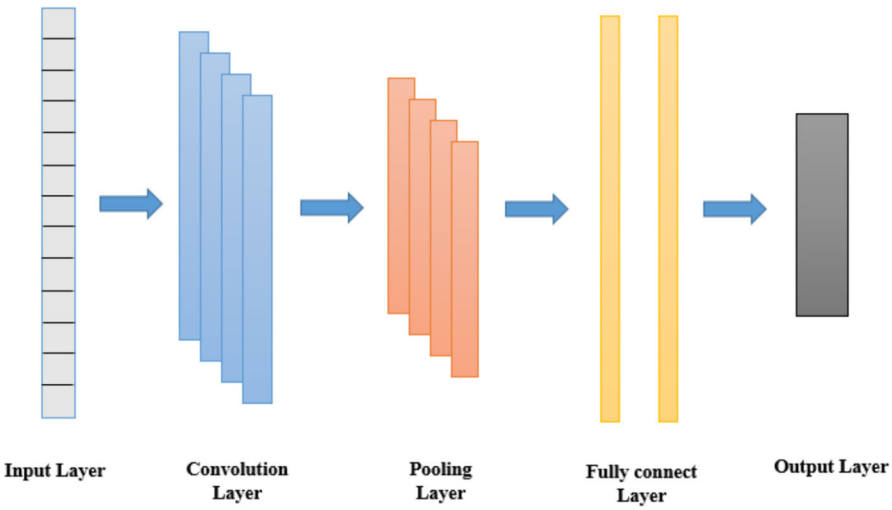


Fig. 1 CNN architecture

which extract features using various kernels, pooling layers that reduce dimensionality through downsampling, and fully connected layers that integrate these extracted features to produce final classification outcomes [41].

3.1.2 Recurrent neural network (RNN)

For processing sequential input, RNNs are a particular kind of artificial neural network that was ideal for applications like language translation, natural language processing (NLP), anomaly detection, and image captioning. RNN have been seamlessly integrated into popular applications like Siri, voice search, and Google Translate [42]. The architecture of an RNN consists of a chain of recurrent units or cells that process time series data. Every recurrent unit preserves a hidden state h_t that serves as a memory, enabling it to hold onto data from earlier time steps h_{t-1}. In Fig. 2, a basic RNN is shown. Nevertheless, It can be difficult for RNN to identify long-range relationships in sequences, though, because of the vanishing gradient problem. In order to overcome this constraint, more sophisticated RNN designs like LSTM network and its variant, BiLSTM, as well as GRU network and its variant, BiGRU, have been developed. These innovative frameworks incorporate gating mechanisms that greatly improve capture and control of information flow across longer sequences, enhancing their capacity to more effectively manage long-range dependencies [43].

3.1.3 Long short-term memory (LSTM)

LSTM is a sort of RNN that created to get around some of the drawbacks of conventional RNN, namely the vanishing gradient problem and trouble managing long-term dependencies. Proposed by Hochreiter et al. [44], LSTM is specifically

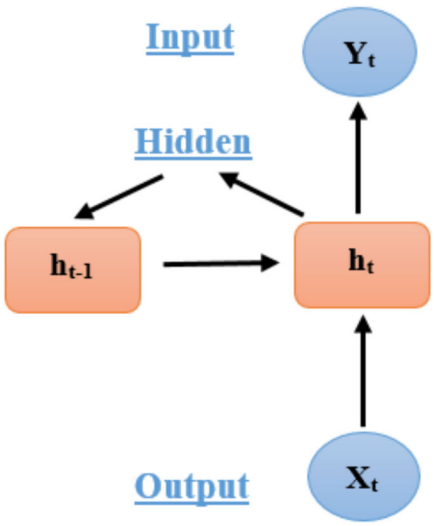


Fig. 2 Basic structure of RNN

engineered to retain information for extended periods, resulting in its suitability for numerous applications, involving image and video processing, machine translation, and language modeling. As seen in Fig. 3, the design of a typical LSTM network consists of a memory cell and three regulators that manage the data flow through the LSTM: an input gate, an output gate, and a forget gate. Together, these elements allow the LSTM to efficiently record and store significant data throughout lengthy sequences, making it a powerful tool for tasks requiring the analysis of sequential data. The forget gate, as defined by Eq. 1, determines whether to keep or discard information from the preceding hidden state C_{t-1} in an LSTM network. It utilizes the sigmoid function to evaluate the relevance of the current input x_t and the preceding hidden state h_{t-1} in this decision-making process. The output of the forget gate, denoted as f_t, ranges between 0 and 1. A value of 0 denotes complete forgetfulness of the information, whereas a value of 1 denotes complete content preservation.

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

Where, b_f : Bias on the hidden layer, f_t : forget gate vector, W_f : Weight on the hidden state, and x_t : input vector.

Concurrently, the input gate in an LSTM network regulates the incorporation of new data into the memory. Its computation involves a two-step process: First, it identifies crucial information for integration into the unit status using the sigmoid activation function. Then, it utilizes the tanh activation function to generate a fresh vector for updating the unit status. These steps are mathematically expressed in Eqs. 2 and 3. Following that, the current cell state C_t can be upgraded by multiplying the outputs of the forget gate and the input gate, as formulated in Eq. 4.

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{2}$$

$$\tilde{C}_t = \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

Where, i_t : input gate vector, W_i : Weight on the hidden layer, $bi\&bc$: bias vectors, $\tilde{C}_t$: Candidate vector, C_t : Cell state vector, f_t : Forget vector at time t, i_t : Input vector at time t, and C_{t-1} : Cell vector at time t − 1.

In parallel, the output gate in an LSTM network plays a crucial role in deciding whether the current cell value should contribute to the network's output. It begins by using a sigmoid layer to evaluate the cell state output, which is then executed by the tanh function to produce a value among -1 and 1. The output of the sigmoid gate is then multiplied by this value to guarantee that only the selected values are

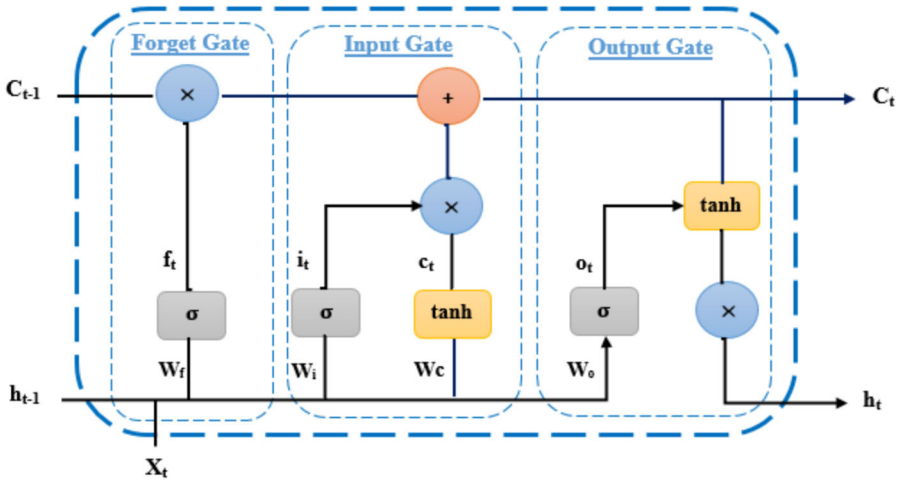


Fig. 3 Structure of LSTM cell

sent for further processing [45]. The detailed operations for generating the output are outlined in Eqs. 5 and 6.

$$O_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{5}$$

$$h_t = o_t * tanh\left(C_t\right) \tag{6}$$

Where, o_t : output vector, b_o : Constant value bias and W_o : Weight on hidden layer.

3.1.4 Bidirectional LSTM (BiLSTM)

The LSTM architecture is extended by BiLSTM, which learns from both directions at the same time to enhance model performance in sequence categorization tasks [46]. BiLSTM is an advantageous technique since it trains two LSTM networks instead of one when all timesteps in the input sequence are accessible. In determining the output, BiLSTM takes into account both forward and backward activations. The forward LSTM unit extracts the forward features of the input layer's data sequence, Whilst the backward LSTM unit extracts the backward features [47]. The outputs from both directions are then combined in the output layer. Figure 4 illustrates the BiLSTM's construction. The process is:

$$\overrightarrow{h_t} = LSTM\left(x_t, \overrightarrow{h_{t-1}}\right) \tag{7}$$

$$\overleftarrow{h_t} = LSTM\left(x_t, \overleftarrow{h_{t-1}}\right) \tag{8}$$

$$h_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] \tag{9}$$

3.1.5 Gated recurrent unit (GRU)

A common kind of RNN that has been applied to many different applications is GRU. It is renowned for being efficient in managing long-range dependencies over time in sequences and reducing the vanishing gradient issue that conventional RNNs frequently face. Compared to the more complex architecture of LSTM network, the GRU is a simplified and improved version that reduces the number of gates [48]. Unlike LSTM, GRU lacks a separate cell state and transfers information solely via the hidden state. Furthermore,

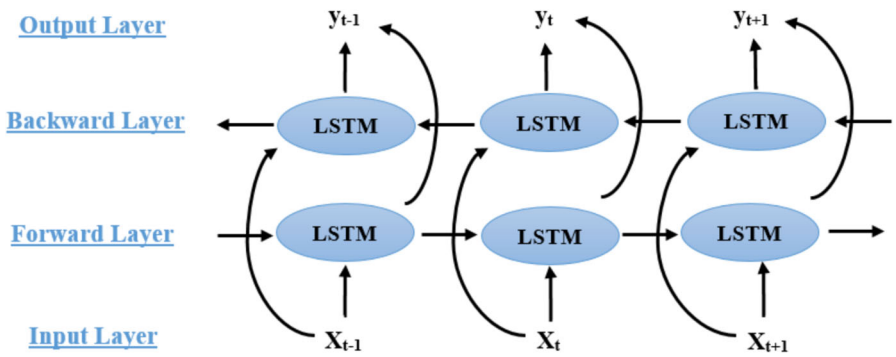


Fig. 4 BiLSTM structure

GRU requires only two gates to operate effectively, as shown in Fig. 5. This streamlined architecture results in lower tensor operation requirements and reduced training parameters, giving GRU a significant advantage in terms of performance and convergence. In contrast to LSTM, GRU architecture includes two gates: Update and Reset gate. The Update gate integrates the functionalities of the input and forget gate in LSTM, determining whether information should be retained or filtered. Its mathematical representation is captured in Eq. 10. On the other hand, the Reset gate dictates the amount of the memory's prior data needs to be discarded and is represented by Eq. 11. The interplay between these gates ultimately determines the final output. Equation 12 describes the GRU's present memory state, and Eq. 13 shows the GRU's last memory state [49].

$$r_t = \sigma\left(W_r\left[h_{t-1}, x_t\right]\right) \tag{10}$$

$$z_t = \sigma\left(W_z\left[h_{t-1}, x_t\right]\right) \tag{11}$$

$$\widetilde{h_t} = tanh\left(\left[r_t \times h_{t-1}, x_t\right]\right) \tag{12}$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \widetilde{h_t} \tag{13}$$

where the input, output, reset, and update gate vectors are denoted by the symbols, x_t, h_t, r_t, and z_t.

3.1.6 Bidirectional GRU (BiGRU)

The GRU design is expanded upon by BiGRU. It addresses the limitations of unidirectional GRU by incorporating information from both the forward and backward states simultaneously [50]. This simultaneous processing allows BiGRU to improve the accuracy of its output by leveraging information from both directions, thereby enhancing its capacity to identify intricate dependencies in sequential information. The architecture of the BiGRU cell, as depicted in Fig. 6, takes into account the current input x_t, the forward hidden state h_{t-1}, the backward hidden layer's output at time step t-1 to compute its current hidden layer state [51].

$$\overrightarrow{h_t} = GRU\left(x_t, \overrightarrow{h_{t-1}}\right) \tag{14}$$

$$\overleftarrow{h_t} = GRU\left(x_t, \overleftarrow{h_{t-1}}\right) \tag{15}$$

$$h_t = w_t\overrightarrow{h_t} + v_t\overleftarrow{h_t} + b_t \tag{16}$$

The GRU(.) function denotes the use of the GRU network for nonlinear transformation of the input data. Where, b_t is the bias of the state of the hidden layer at time t, w_t and v_t are the weights of the states h_{t-1} of the forward hidden layer and h_{t-1} of the backward hidden layer corresponding to BiGRU at time t, respectively.

3.2 Intrusion detection model based on CNN-RNN

The proposed approach introduces an advanced IDS by combining a 1-Dimensional CNN (1D-CNN) with four different RNN models, including LSTM, BiLSTM, GRU, and BiGRU. LSTM and GRU handle long-term dependencies effectively, while BiLSTM and BiGRU enhance this by processing information from both forward and backward sequences. This

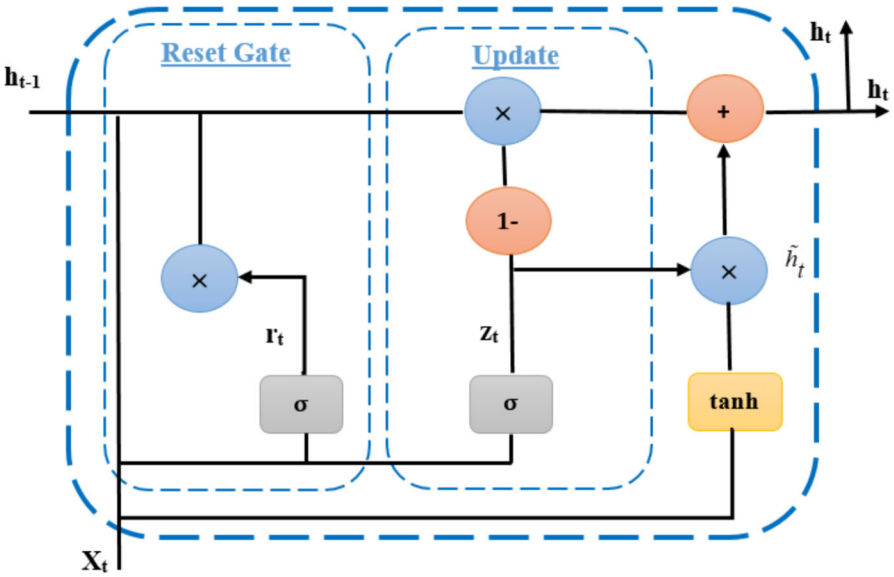


Fig. 5 Structure of GRU cell

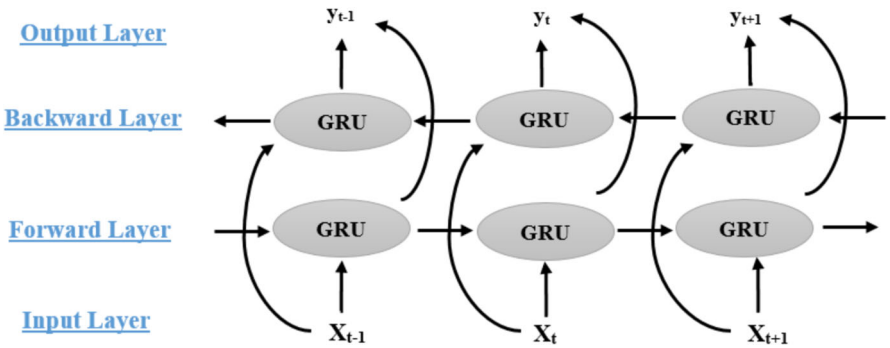


Fig. 6 BiGRU structure

integrated system aims to effectively detect intrusions in real-world data traffic within IoT networks. The rationale behind this fusion lies in the complementary nature of the two types of networks. The CNN acts as a spatial feature extractor [52], optimizing computational resources and being well-suited for IoT devices with limited capabilities. On the other hand, RNNs perform classification tasks by utilizing the sequential feature data provided by the CNN, effectively serving as a classifier. Additionally, integrating the RNN network into the CNN not only addresses the issue of parameter explosion but also enhances the IDS's capability to process characteristic data with long-distance attributes and extensive time series, thereby improving overall intrusion detection performance [53]. Our network architecture begins with an input layer followed by a 1D-CNN with 64 filters, which processes an input vector of size 8×1 to extract spatial features. A max-pooling layer follows in order to minimize the dimensionality of the data. We then introduce a Batch Normalization layer to adjust parameters between intermediate layers, aiding in faster training. The outputs are passed to an RNN layer consisting of LSTM, BiLSTM, GRU, and BiGRU models, each with 128 units, to capture temporal dependencies. To improve the stability of hidden states and mitigate overfitting risks, each RNN layer is accompanied by a dropout probability of 0.3. Lastly, a fully connected layer followed by an output layer is employed for categorization, with the Relu and Softmax activation functions utilized throughout the model. The proposed network architecture is visually summarized in Fig. 7, and additional details are provided in Table 1.

4 Experiments evaluation

4.1 Dataset

4.1.1 Dataset description

To evaluate the effectiveness of our methodology, a recently generated Netflow-based dataset called NF-UQ-NIDS was employed during experiments, which consolidates four widely recognized benchmarks: NF-BoT-IoT, NF-ToN-IoT, NF-UNSW-NB15, and NF-CSE-CIC-IDS2018 [23]. Table 2 offers a thorough summary of these datasets. Together, these datasets encompass 12 features and encompass a broad spectrum of attacks. Notably, they exhibit an imbalanced distribution of classes, with instances of normal and abnormal network traffic flow. The primary motivation behind the development of this dataset was to demonstrate the advantages of employing a standardized NetFlow feature set [54], effectively addressing the issue of disparate characteristics

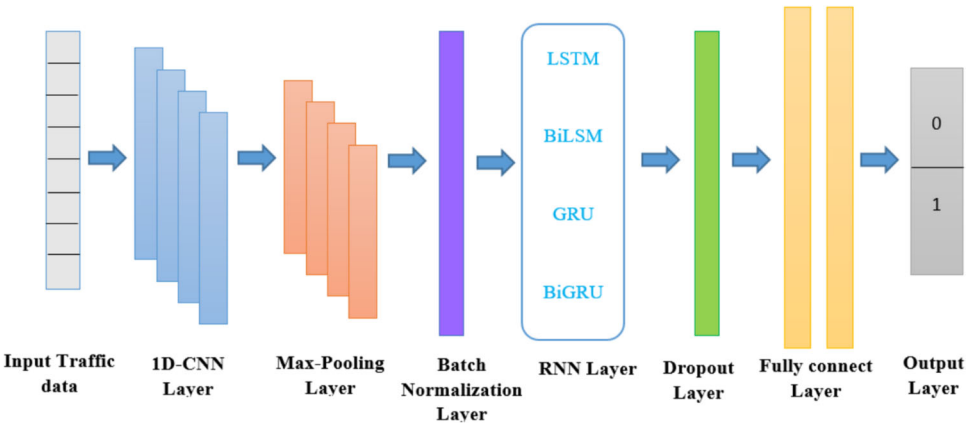


Fig. 7 CNN-RNN based intrusion detection architecture

Table 1 Configuration hyperparameters

Parameters	Values
1 D-CNN layer	Padding =same, Filter=64, Kernel size=3
RNN layer	128 memory units
Dropout	0.3
Activation function	Relu, Softmax
Iterations	50
Batch size	64
Optimizer	Adam
Learning rate	0.001
Cost function	Binary cross-entropy

across individual datasets and mirroring real-world network environments. Furthermore, this initiative should open up opportunities for the creation of larger, more comprehensive NIDS datasets, encompassing a diverse array of network configurations and attack scenarios.

4.1.2 Dataset preprocessing

As previously mentioned, four benchmark datasets are combined to create the NF-UQ-NIDS dataset, where the main categories have been combined into attack groups. Specifically, DoS assaults such as Hulk, SlowHTTPTest, Slowloris, and GoldenEye are consolidated under the DoS category. Similarly, DDoS assaults including HOIC, LOIC-HTTP, and LOIC-UDP are unified into the DDoS category, while various BruteForce attacks like XSS, SSH Web, and FTP are merged within a single BruteForce group. Additionally, SQL Injection attacks have been included in the injection attacks category, as outlined in Table 3. Furthermore, to prevent bias in the analysis and overfitting of models, four features:L4 DST PORT, L4 SRC PORT, IPV4 DST ADDR, and IPV4 SRC ADDR have been removed from the dataset. Subsequently, data cleaning procedures were applied to handle null and duplicate values, ensuring the effectiveness of the model evaluation. For the experiments, eight fundamental features that significantly impact the results were selected. These features were normalized to a range between 0 and 1. The labels were encoded using the one-hot encoding technique [55, 56]. Following this, 80% of the dataset was used for training, while 20% was used for testing. Given that this study involves binary classification, the Benign label was assigned the value 0, while the Attack labels were assigned the value 1 [57], as illustrated in Table 4.

4.2 Statical metrics

In the case of binary classification, Table 5 presents the confusion matrix with four possible outcomes: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). Attack samples are represented by TP as the number of correctly identified samples, and benign samples are represented by FP as the number of misclassified samples. The number of benign samples that were accurately identified is denoted by TN, while the number of misclassified attack samples is represented by FN [58].

The analysis of the overall results includes well-known evaluation Criteria like accuracy, recall, precision, F1 Score, and FAR [59]. The following formulas are used to compute these measures Eqs. 17-21.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{17}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{18}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{19}$$

$$F1score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{20}$$

Table 2 Dataset overview

Datasets	Total of samples	Features
NF-BoT-IoT	600,100	IPV4 SRC ADDR, IPV4 DST ADDR, L4 SRC PORT,
NF-ToN-IoT	1,379,274	L4 DST PORT, PROTOCOL, TCP FLAGS, L7 PROTO,
NF-UNSW-NB15	1,623,118	IN BYTES, OUT BYTES, IN PKTS, OUT PKTS,
NF-CSE-CIC-IDS2018	8,392,401	FLOW DURATION MILLISECONDS
NF-UQ-NIDS	11,994,893	

Table 3 Distribution of different attack classes

Datasets	Attack classes
NF-BoT-IoT	(Benign: 13859), (Reconnaissance: 470655), (DDoS: 56844), (DoS: 56833), (Theft: 1909)
NF-ToN-IoT	(Benign: 270279), (Backdoor: 17247), (DoS: 17717), (DDoS: 326345), (Injection: 468539), (MITM: 1295), (Password: 156299), (Ransomware: 142), (Scanning: 21467), (XSS: 99944)
NF-UNSW-NB15	(Benign: 1550712), (Fuzzers: 19463), (Analysis: 1995), (Backdoor: 1782), (DoS: 5051), (Exploits: 24736), (Generic: 5570), (Reconnaissance: 12291), (Shellcode: 1365), (Worms: 153)
NF-CSE-CIC-IDS2018	(Benign: 5534773), (DoS attacks-Hulk: 108129), (SSH-Bruteforce: 71148), (Infiltration: 59374), (DDoS attacks-LOIC-http: 49751), (DoS attacks-GoldenEye: 32582), (DoS attacks-Slowloris: 17109), (Bot: 15498), (DoS attacks-SlowHTTPTest: 14116), (FTP-BruteForce: 14116), (DDOS attack-LOIC-UDP: 1667), (DDOS attack-HOIC: 230), (Brute Force-Web: 173), (Brute Force-XSS: 101), (SQL Injection: 36)
NF-UQ-NIDS	(Benign: 7168002), (Reconnaissance: 478047), (Injection: 460848), (DDoS: 305586), (DoS: 248964), (Password: 144792), (XSS: 99913), (Brute Force: 85479), (Infiltration: 59374), (Exploits: 23418), (Scanning: 20618), (Fuzzers: 17994), (Backdoor: 17980), (Bot: 15498), (Generic: 4165), (Theft: 1849), (Shellcode: 1365), (MITM: 1288), (Analysis: 731), (Worms: 153), (Ransomware: 142)

$$FAR = \frac{FP}{(FP + TN)} \tag{21}$$

4.3 Model configuration and parameters

To execute and train the CNN-RNN architectures, we conducted experiments on a CPU equipped with an Intel(R) Core(TM) i7-8700U CPU running at 3.20 GHz, 16 GB of memory, and an Nvidia GeForce GT710 GPU. We utilized Python (version 3.7.13) with TensorFlow (version 2.9.1) and Keras (version 2.9.0) frameworks, along with supplementary library packages including pandas, numpy, matplotlib, and Sklearn. All trials were undertaken within a Windows 10 64-bit operating system environment.

In our experimental setup, each model underwent meticulous parameter tuning for optimal binary classification during training. We conducted training for 50 epochs, employing a batch size of 64 [60]. To enhance our models' performance, we adopted the Adaptive Moments (Adam) optimizer [61] setting the learning rate to 0.001 [62], a choice that consistently yielded high validation accuracy and maximum detection rates. Additionally, we applied an early stopping strategy with 10 iterations of patience to avoid overfitting and minimize the epochs count. This strategy halts the process of training when the validation loss fails to diminish after certain amount of iterations, effectively optimizing the model's performance. We employed the binary cross-entropy as loss function to adjust the optimizer weights. In convolution and dense layers, we applied the Relu activation function [63], whereas the output layer utilized Softmax [64]. Table 1 detailed the parameter configurations for our empirical findings.

Overall, the experimental settings were thoughtfully configured to enhance the effectiveness of the CNN-RNN

Table 4 Details of class labels

Datasets	Category	Label	Number of samples
NF-BoT-IoT	Benign	0	13,859
	Attack	1	586,241
NF-ToN-IoT	Benign	0	270,279
	Attack	1	1,108,995
NF-UNSW-NB15	Benign	0	1,550,712
	Attack	1	72,406
NF-CSE-CIC-IDS2018	Benign	0	7,373,198
	Attack	1	1,019,203
NF-UQ-NIDS	Benign	0	9,208,048
	Attack	1	2,786,845

Table 5 Confusion Matrix

Actual	Predicted	
	Attack	Benign
Attack	TP	FN
Benign	FP	TN

architectures. These carefully selected components established a rigorous framework, ensuring the reliability and validity of our results.

5 Results evaluation

This section provides a comprehensive summary of our performance analysis, highlighting key findings and discussing the strengths and limitations of our models. We conducted a meticulous comparison between our models and existing methodologies in the literature to draw meaningful insights and conclusions.

5.1 Performance analysis

5.1.1 Training and validation accuracy and loss changes

The training and validation accuracy and loss curves provide important information about the model's functionality and generalization capability across the training epochs. Figures 8, 9, 10, 11 and 12 depict the changes in training and validation accuracy and loss across 50 epochs for our proposed models, providing a visual representation of the model's learning phase and convergence.

As depicted in Fig. 8, the training accuracy of the NF-BoT-IoT dataset shows a consistent increase across all models with each epoch, indicating effective learning from the training data. However, the validation accuracy of the CNN-LSTM and CNN-BiLSTM models exhibits a slight decline after epoch 45, suggesting a potential issue with overfitting. Regarding the loss metrics, both training and validation losses demonstrate a steady decrease throughout the epochs for all models, except for the validation loss of the CNN-BiLSTM model, which initially decreases, peaks around epoch 7, and then stabilizes.

Figure 9 makes it clear that the training accuracy of the NF-ToN-IoT dataset steadily increases with each epoch for all models. However, the validation accuracy exhibits an unstable pattern of increase, fluctuating notably, especially from the beginning to around epoch 20. Specifically, the CNN-LSTM, CNN-BiLSTM, CNN-GRU, and CNN-BiGRU models initially experience a decrease in validation accuracy, reaching a peak around epochs 4, 5, 8, and 19, respectively, before stabilizing. As for the loss metrics, both training and validation losses consistently decrease over the epochs. The training loss shows a slow decrease at the beginning, gradually converging, while the validation loss decreases more slowly but steadily throughout the training process, albeit with some fluctuations. These trends suggest that the model is improving its performance and effectively generalizing to unseen inputs, despite challenges observed in validation accuracy.

Figure 10 illustrates the training accuracy for the NF-UNSW-NB15 dataset, which shows a consistent increase across all models with each epoch, indicating effective learning from the training data. In contrast, the validation accuracy exhibits some fluctuation before stabilizing. Both training and validation losses steadily decrease throughout the epochs for all models, suggesting that the models are learning effectively and generalizing well to the validation data.

The training and validation accuracy, along with loss curves for the NF-CSE-CIC-IDS2018 dataset, are presented in Fig. 11. Initially, both training and validation accuracies

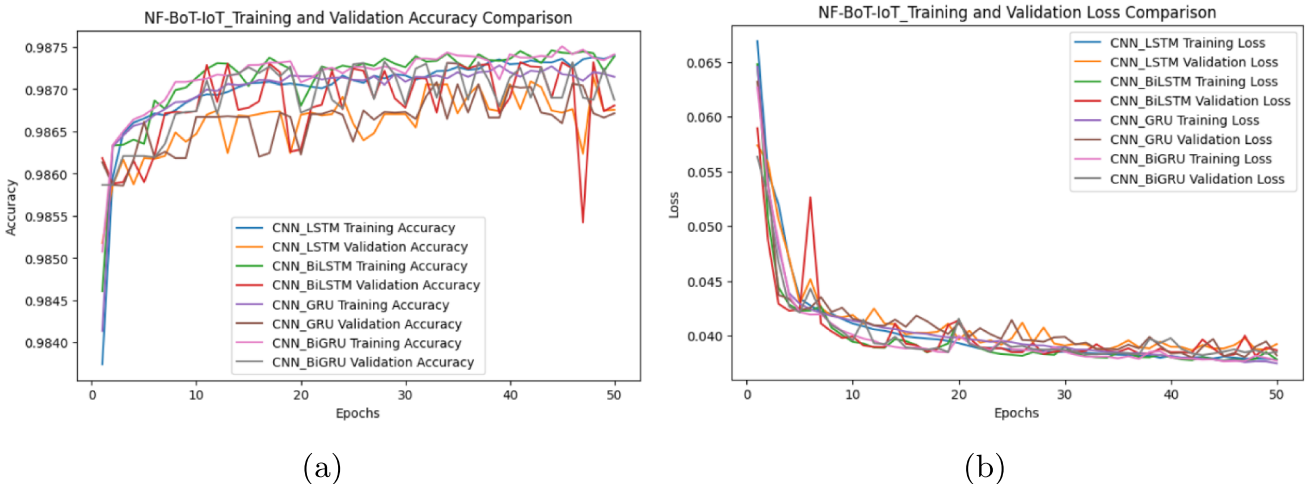


Fig. 8 NF-BoT-IoT-Training and Validation (a) Accuracy. (b) Loss comparison

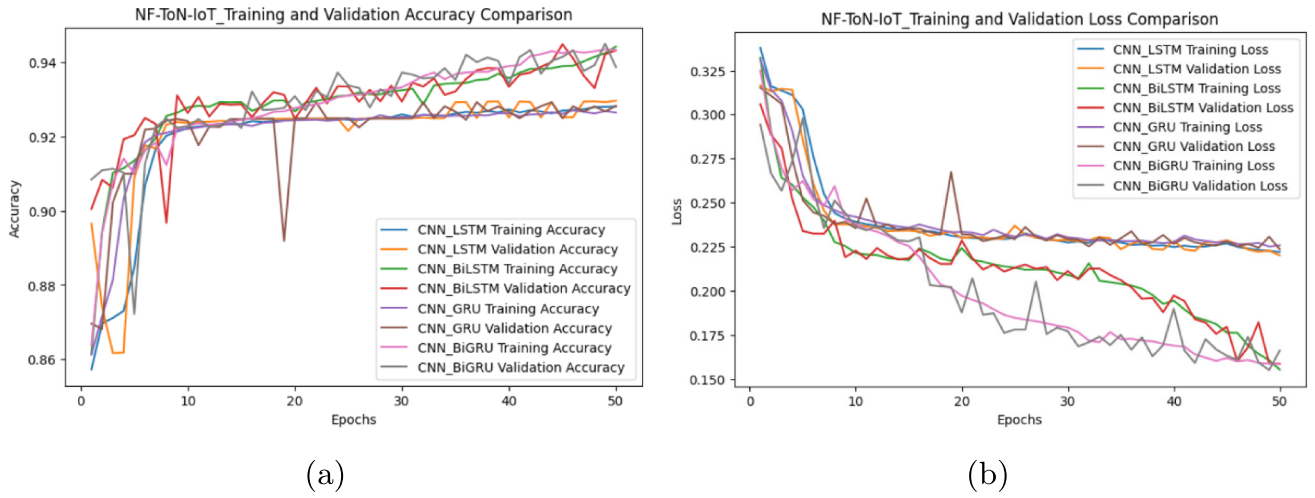


Fig. 9 NF-ToN-IoT-Training and Validation (a) Accuracy. (b) Loss comparison

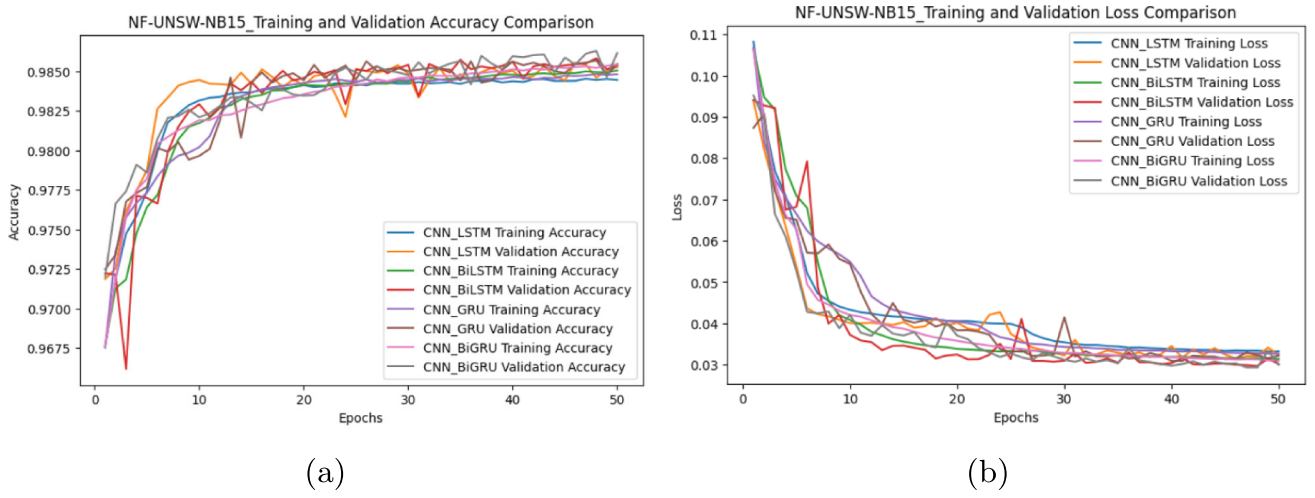


Fig. 10 NF-UNSW-NB15-Training and Validation (a) Accuracy. (b) Loss comparison

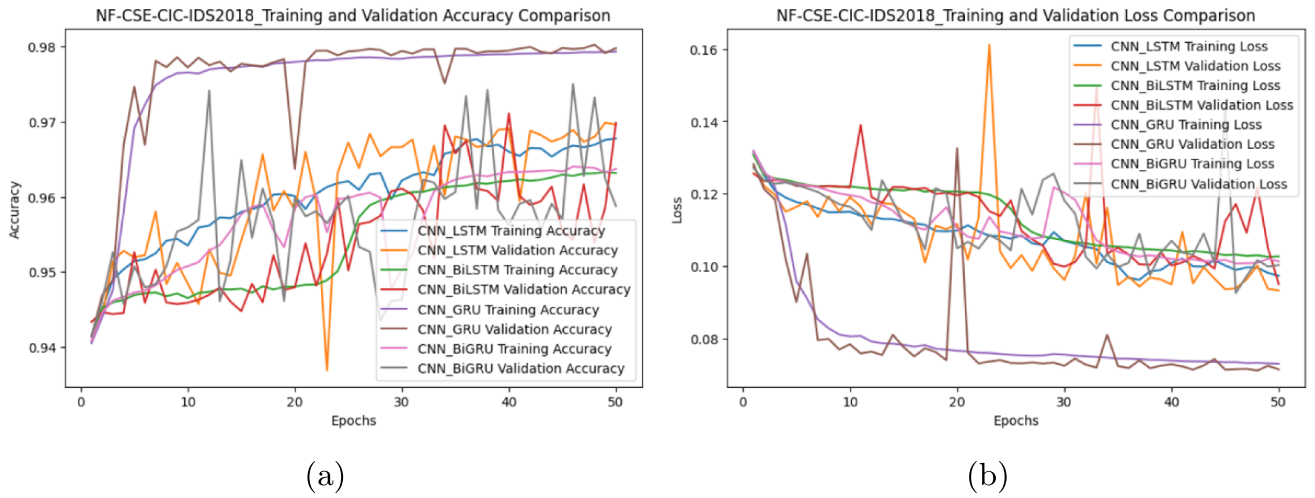


Fig. 11 NF-CSE-CIC-IDS2018-Training and Validation (a) Accuracy. (b) Loss comparison

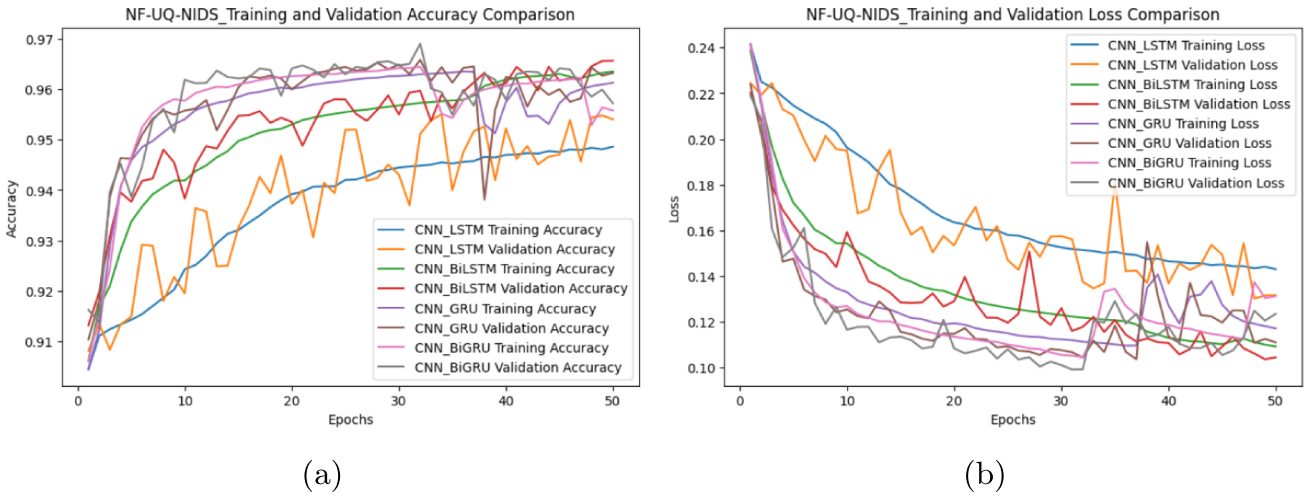


Fig. 12 NF-UQ-NIDS-Training and Validation (a) Accuracy. (b) Loss comparison

exhibit erratic behavior, with frequent fluctuations. While the training accuracy of the CNN-LSTM, CNN-BiLSTM, and CNN-GRU models steadily increases over epochs, indicating learning, the CNN-BiGRU model's accuracy remains less stable. Conversely, the validation accuracy fluctuates widely, suggesting challenges in generalization. In contrast, the training loss consistently decreases, indicating improved fit to the training data. However, the validation loss fluctuates significantly, showing peaks and valleys. This behavior implies that while the models are fitting well to the training data, they may struggle to extrapolate to unknown data, possibly indicating overfitting.

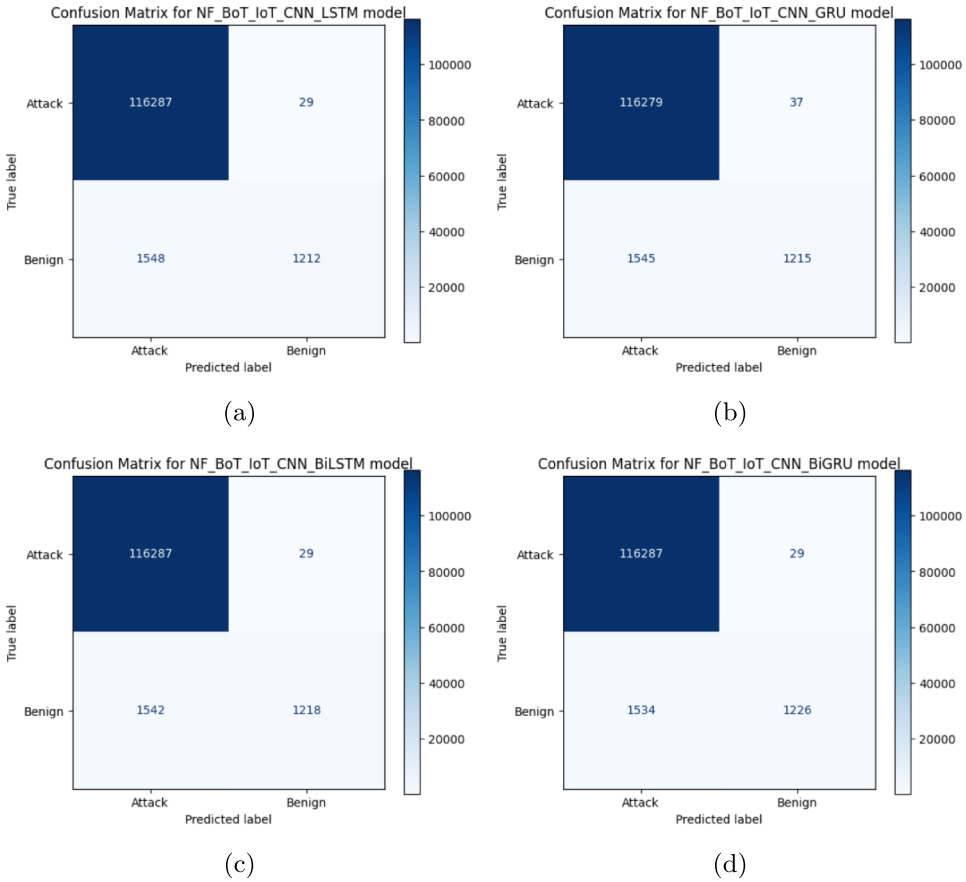


Fig. 13 NF-BoT-IoT-Confusion Matrix comparison

In Fig. 12, the training accuracy of the CNN-LSTM and CNN-BiLSTM models for the NF-UQ-NIDS dataset consistently rises with each epoch, attaining nearly 96% by the end of training. However, the training accuracy of the CNN-GRU and CNN-BiGRU models decreases after epoch 30, showing some fluctuations but ultimately improving over time. Similarly, the validation accuracy for the CNN-LSTM and CNN-BiLSTM models exhibits fluctuations but improves overall, while for the CNN-GRU and CNN-BiGRU models, it decreases after epoch 30, yet shows some fluctuations before improving again.

5.1.2 Confusion matrix analysis

Confusion matrix analysis plays a pivotal role in assessing our models' performance in binary classification [65]. It offers valuable insights into class confusion, model strengths, and areas needing improvement. A high TP count and low FN count signify the model's adeptness in recognizing attack instances, showcasing high recall. Likewise, a high TN count and low FP count indicate proficient benign class recognition. These metrics, combined with overall accuracy, gauge how well the model can distinguish between the two classes. The analysis of confusion matrices, as depicted in Figs. 13, 14, 15, 16 and 17, indicates that all models effectively distinguished classes across various datasets, underscoring their efficacy

5.1.3 ROC Curve analysis

To gauge our models' effectiveness, we employ Receiver Operating Characteristic (ROC) curves [66], pivotal for appraising DL models, especially in binary classification tasks. The ROC curve visually depicts the model's True Positive Rate (TPR) against the False Positive Rate (FPR), where greater TPR and lesser FPR indicating superior performance. Further, the Area Under Curve (AUC) metric quantitatively assesses the model's utility, with a smooth curve and a large AUC signifying robustness and generalizability. Figure 18 showcases ROC curves for the four models across the five datasets. Notably, the NF-UQ-NIDS dataset presents the most promising outcomes. Among the models, the CNN-

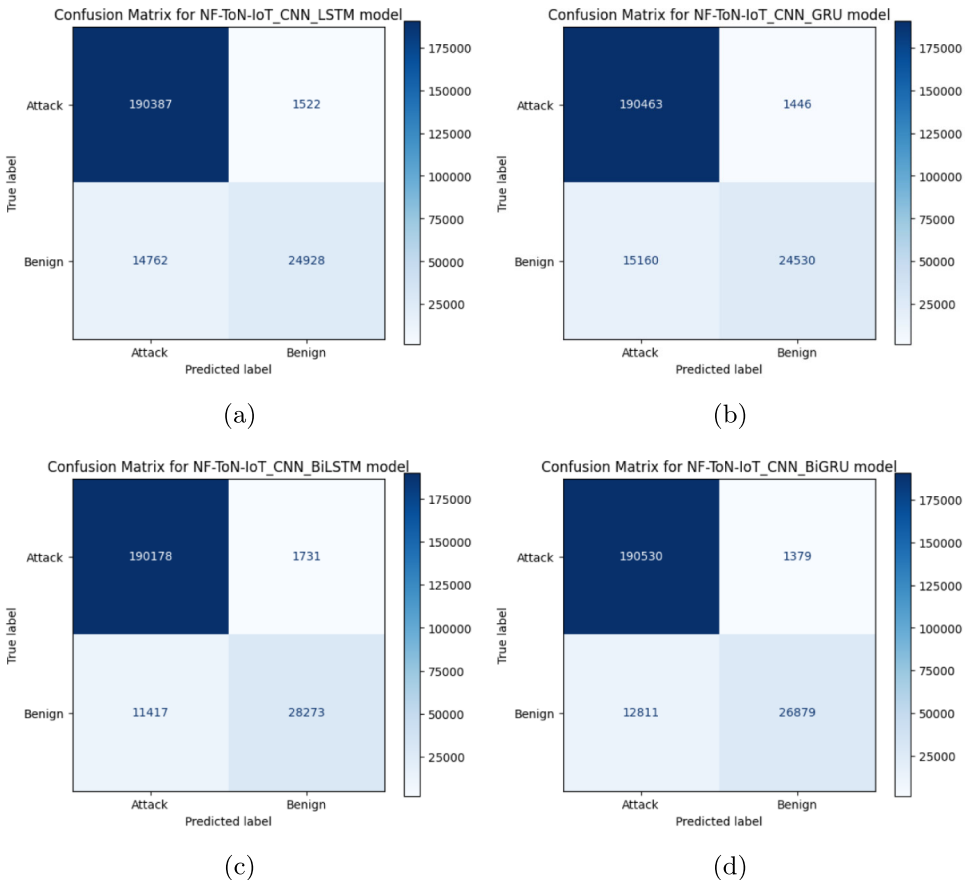


Fig. 14 NF-ToN-IoT-Confusion Matrix comparison

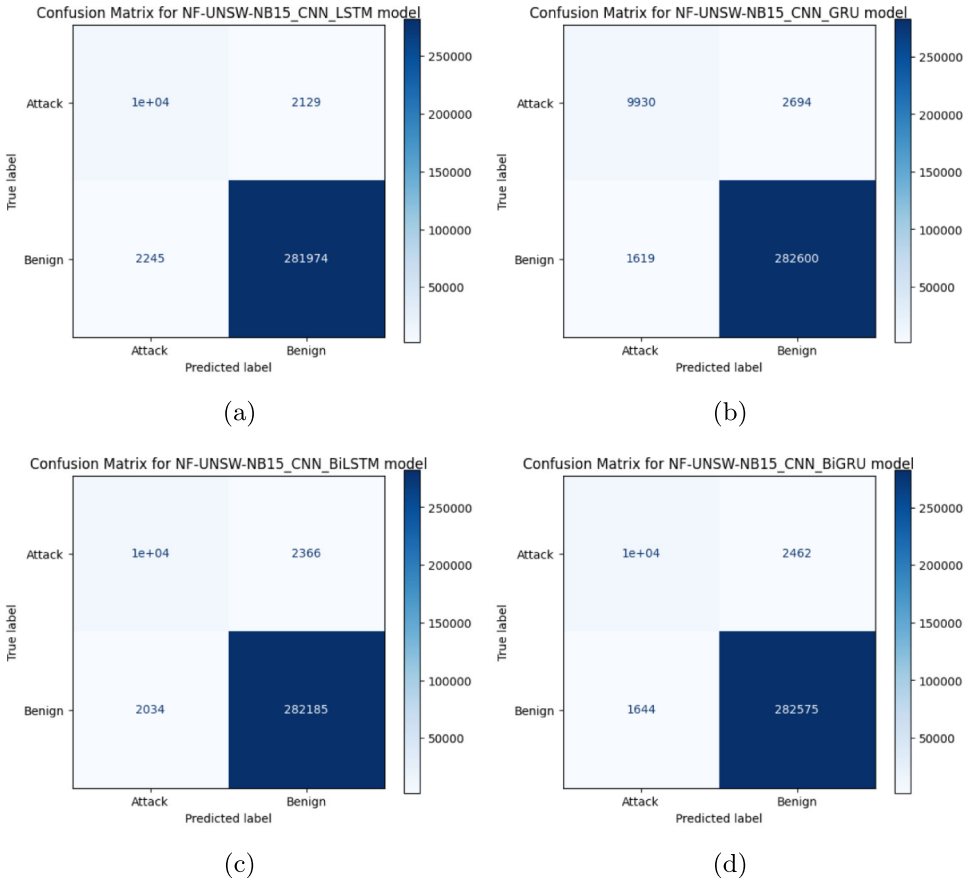


Fig. 15 NF-UNSW-NB15-Confusion Matrix comparison

GRU approach emerges as the most effective classifier to distinguish between attack and benign classes, achieving an impressive AUC score of 0.936.

5.1.4 Accuracy, precision, recall, F1-score and FAR

This research employed four models including CNN-LSTM, CNN-BiLSTM, CNN-GRU, and CNN-BiGRU for binary classification across various datasets: NF-BoT-IoT, NF-ToN-IoT, NF-UNSW-NB15, NF-CSE-CIC-IDS2018, and NF-UQ-NIDS. Each dataset underwent conversion to a binary label classification, distinguishing between benign and attack behaviors. The outcomes of these trials are detailed in Tables 6, 7, 8, and 9. These tables provide a comprehensive overview of the models' performance, highlighting their ability to differentiate between different types of network traffic effectively.

Figure 19 illustrates the comparative accuracy of our four models across the five distinct datasets used. The NF-BoT-IoT and NF-UNSW-NB15 datasets show the most promising results, with accuracies above 98.50% for all models. However, the NF-ToN-IoT dataset generally lags behind, presenting the worst results, with an accuracy around 94.32% for the CNN-BiLSTM model. Notably, the CNN-CRU and CNN-BiLSTM models reveal the highest accuracy, assessed at 97.97% and 96.56%, respectively, for the NF-CSE-CIC-IDS2018 and NF-UQ-NIDS datasets. These yields highlight the influence of dataset characteristics on model performance, underscoring the essential role of dataset selection in model evaluation and deployment for intrusion detection tasks. These subtle observations are vital for identifying the most valuable model for particular application scenarios, eventually improving the effectiveness of the IDS.

The evaluation results for the CNN-LSTM model's binary classification are provided in Table 6. Interestingly, the NF-BoT-IoT dataset accomplished the greatest accuracy at 98.68%. It also demonstrated commendable precision, recall, and F1 Score in favor of the attack class, with values of 98.69%, 99.98%, and 99.33%, respectively. However, the CNN-LSTM model display the poorest recall and F1 score for the benign class, likely due to the benign categories being less representative in the dataset, as indicated in Table 4. Additionally, as illustrated in Fig. 20, the NF-BoT-IoT-CNN-LSTM model showcased the lowest FAR among all models for the benign categories, indicating its capability to minimize false positive detections, measured at 0.0002.

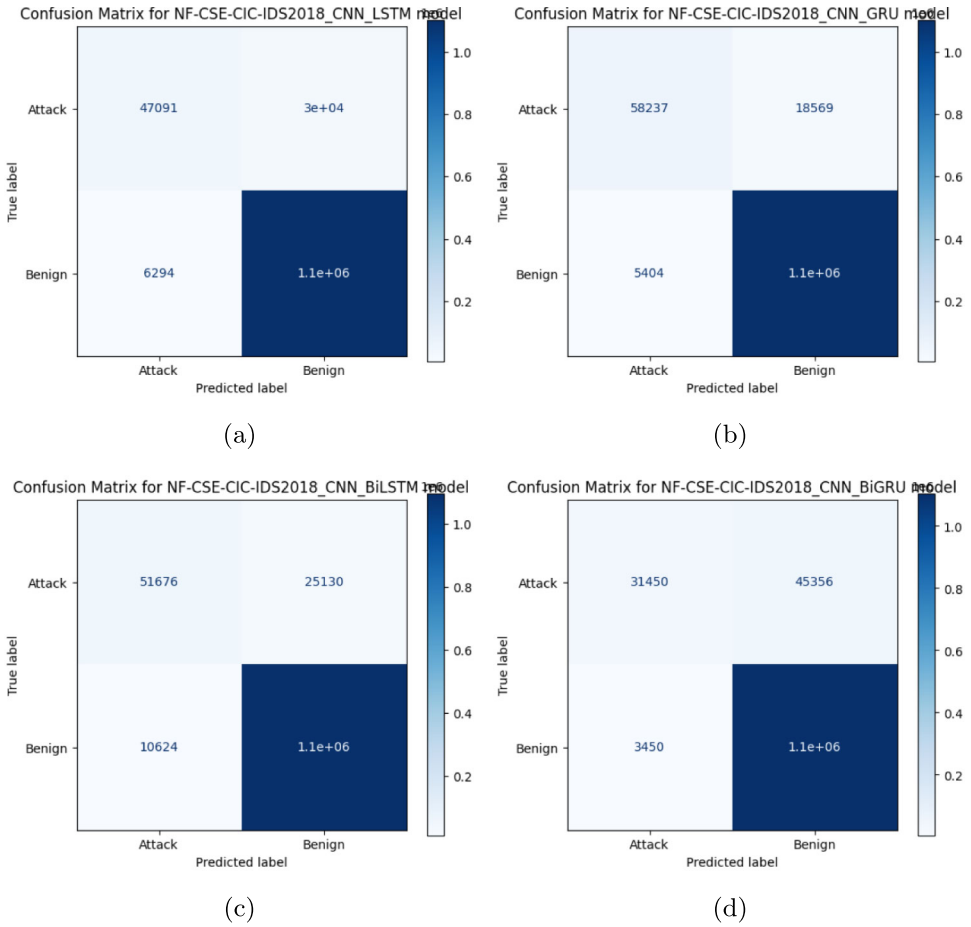


Fig. 16 NF-CSE-CIC-IDS2018-Confusion Matrix comparison

In contrast, for the NF-ToN-IoT dataset, the CNN-LSTM model achieved the lowest accuracy at 92.97%. While the detection rates (recall) for the attack classes were high, the detection rates for the benign class were notably lower. Furthermore, the CNN-LSTM model's FAR was 0.3719 for the attack class and 0.0079 for the benign class, a consequence of the benign categories being a minority in the dataset. Conversely, the CNN-LSTM model demonstrated its effectiveness on the NF-UNSW-NB15, NF-CSE-CIC-IDS2018, and NF-UQ-NIDS datasets, achieving accuracies of 98.53%, 96.96%, and 95.40%, respectively. Given that the benign categories were the most representative in these datasets (see Table 4), the CNN-LSTM model exhibited good detection rates for the benign class and lower FAR for the attack class, measured at 0.0079, 0.0057, and 0.0186, respectively.

Table 7 presents the outcomes of the CNN-BiLSTM approach. In comparison to the CNN-LSTM model, the CNN-BiLSTM performs admirably, showcasing competitive accuracy of 98.68% for NF-BoT-IoT, 94.32% for NF-ToN-IoT, 98.54% for NF-UNSW-NB15, 96.98% for NF-CSE-CIC-IDS2018, and 96.56% for NF-UQ-NIDS. For the NF-BoT-IoT and NF-ToN-IoT datasets, the CNN-BiLSTM approach correctly identified the attack class but misclassified the benign class. It also exhibited a relatively low FAR, measured at 0.0002 and 0.009 for the benign categories, compared to 0.5587 and 0.2877 for the attack class. The CNN-BiLSTM model show excellent efficiency with regard to precision, recall, and F1 Score for the benign category in the NF-UNSW-NB15, NF-CSE-CIC-IDS2018, and NF-UQ-NIDS datasets. However, it showed limitations to identify the attack group, achieving rates of approximately 82.54%, 67.28%, and 87.44%, respectively. This limitation is attributed to the issue of an unbalanced dataset. Like CNN-LSTM model, the FAR value for the attack class was comparatively low relative to the benign group when employing the CNN-BiLSTM model for the NF-UNSW-NB15, NF-CSE-CIC-IDS2018, and NF-UQ-NIDS datasets, as illustrated in Fig. 20.

Results of the CNN-GRU method are detailed in Table 8. The CNN-GRU consistently demonstrates strong performance across all datasets, with the NF-CSE-CIC-IDS2018 dataset achieving the highest accuracy compared to CNN-LSTM, CNN-BiLSTM and CNN-BiGRU in the same dataset at approximately 97.97%. This suggests that this dataset may

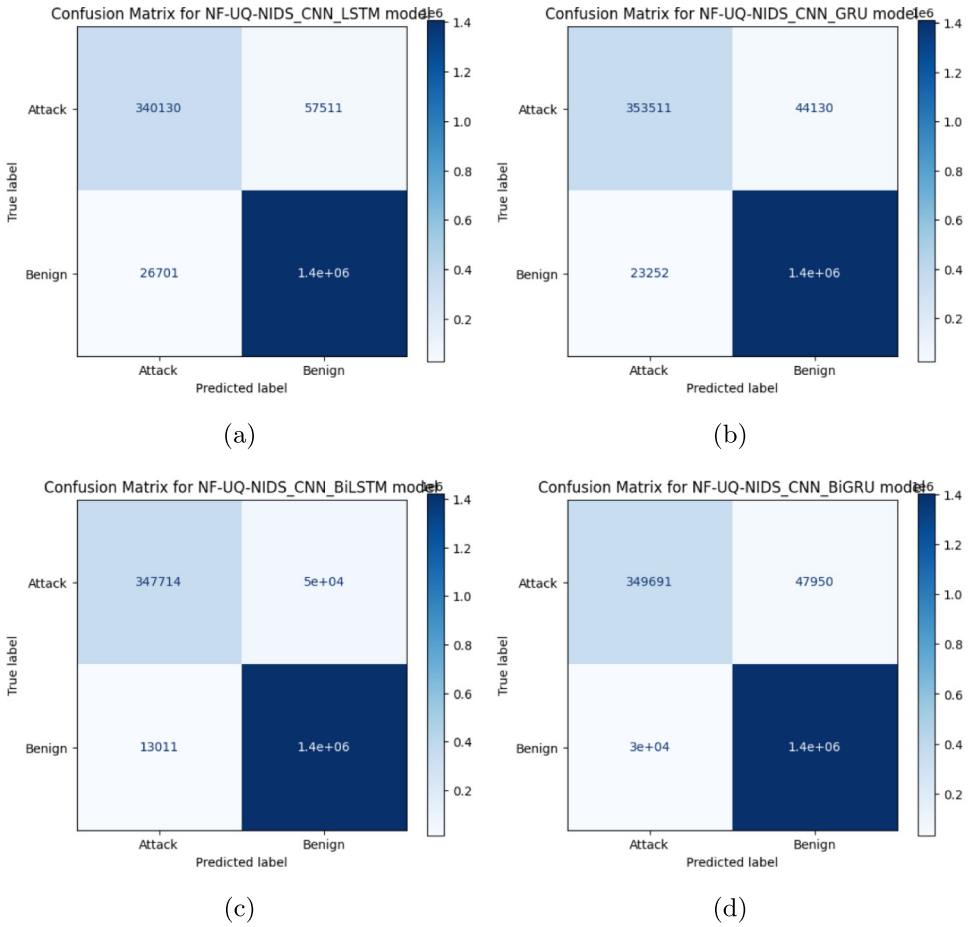


Fig. 17 NF-UQ-NIDS-Confusion Matrix comparison

be more appropriate for such applications as opposed to other methods. Additionally, it achieved its highest accuracy for the NF-BoT-IoT dataset, measured at 98.67%. Similar to the CNN-LSTM and CNN-BiLSTM models, the CNN-GRU accomplished excellent precision, recall, and F1 score for the attack class opposed to benign group for the NF-BoT-IoT and NF-ToN-IoT datasets. Besides, the FAR across the NF-BoT-IoT and NF-ToN-IoT datasets was 0.5598 and 0.382 for the attack class, and 0.0003 and 0.0075 for the benign class, respectively. Conversely, for the NF-UNSW-NB15, NF-CSE-CIC-IDS2018, and NF-UQ-NIDS datasets, the CNN-GRU model correctly identified the benign class against the attack group. The FAR was 0.0057, 0.0049, and 0.0162 for the attack category, and 0.2134, 0.2418, and 0.111 for the benign class once the CNN-GRU model was employed with these datasets.

Table 9 illustrates the achievements of the CNN-BiGRU approach. Remarkably, this method outperformed both the CNN-LSTM and CNN-GRU models, with the exception of the NF-CSE-CIC-IDS2018 dataset. Its standout performance was on the NF-BoT-IoT and NF-UNSW-NB15 dataset, reaching the highest accuracy of approximately 98.69% and 98.62%, respectively, underscoring its capability in this dataset. Like the CNN-LSTM, CNN-BiLSTM, and CNN-GRU models, the CNN-BiGRU model showed strong precision, recall, and F1 scores when distinguishing attack instances from benign categories in the NF-BoT-IoT and NF-ToN-IoT datasets. Moreover, it maintained a low FAR in these datasets, with values of 0.5558 and 0.3228 for the attack class, and 0.0003 and 0.0072 for the benign class, respectively. Conversely, for the NF-UNSW-NB15, NF-CSE-CIC-IDS2018, and NF-UQ-NIDS datasets, the CNN-BiGRU model effectively identified benign instances compared to attack categories. The FAR values were 0.0058, 0.0031, and 0.0213 for the attack class, and 0.195, 0.5905, and 0.1206 for the benign class when using the CNN-BiGRU model with these datasets.

5.2 Comparison with the existing literature

In our research, we executed a series of experiments focusing on binary classification. We utilized four distinct models including CNN-LSTM, CNN-BiLSTM, CNN-GRU, and CNN-BiGRU, across five Netflow datasets: NF-

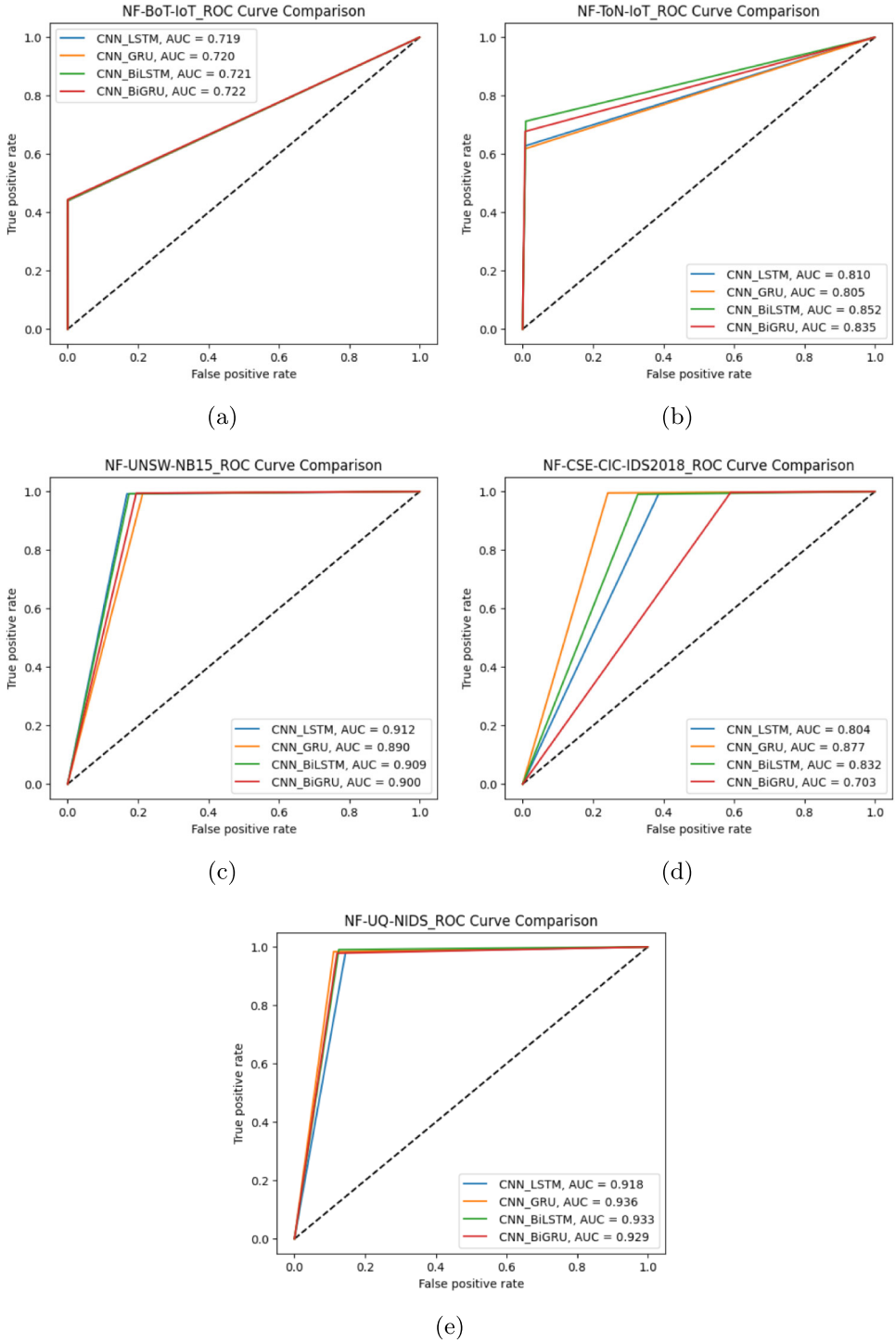


Fig. 18 ROC Curve comparison

BoT-IoT, NF-ToN-IoT, NF-UNSW-NB15, NF-CSE-CIC-IDS2018, and NF-UQ-NIDS. To gauge the efficacy of these models, we compared their performance against the most advanced DL techniques available today for network security. Our findings, detailed in Table 10, showcase the performance of our models in relation to the best results reported in recent literature. Notably, our CNN-BiGRU model stood out with an accuracy of 98.69%, surpassing the performance of Extra Trees [23] and E-GraphSAGE [67] on the NF-BoT-IoT dataset, where they achieved accuracies of 93.82% each. Moreover, our CNN-BiLSTM model outperformed stacked MLP [68] and 2D-ACNN [69] on the NF-ToN-IoT

Table 6 CNN-LSTM model performance for binary classification

Datasets	Class	Accuracy	Precision	Recall	F1 Score	FAR
NF-BoT-IoT	Attack	98.68%	98.69%	99.98%	99.33%	0.5609
	Benign		97.66%	43.91%	60.58%	0.0002
NF-ToN-IoT	Attack	92.97%	92.28%	99.21%	95.9%	0.3719
	Benign		94.25%	62.81%	75.38%	0.0079
NF-UNSW-NB15	Attack	98.53%	82.38%	83.14%	82.76%	0.0079
	Benign		99.25%	99.21%	99.23%	0.1686
NF-CSE-CIC-IDS2018	Attack	96.96%	88.21%	61.31%	72.34%	0.0057
	Benign		97.37%	99.43%	98.39%	0.3869
NF-UQ-NIDS	Attack	95.4%	92.72%	85.54%	88.98%	0.0186
	Benign		96.07%	98.14%	97.09%	0.1446

Table 7 CNN-BiLSTM model performance for binary classification

Datasets	Class	Accuracy	Precision	Recall	F1 Score	FAR
NF-BoT-IoT	Attack	98.68%	98.69%	99.98%	99.32%	0.5587
	Benign		97.67%	44.13%	60.79%	0.0002
NF-ToN-IoT	Attack	94.32%	94.34%	99.1%	96.66%	0.2877
	Benign		94.23%	71.23%	81.13%	0.009
NF-UNSW-NB15	Attack	98.54%	82.97%	82.54%	82.76%	0.0075
	Benign		99.22%	99.25%	99.24%	0.1746
NF-CSE-CIC-IDS2018	Attack	96.98%	82.95%	82.86%	67.28%	0.0096
	Benign		97.76%	99.50%	99.04%	0.3272
NF-UQ-NIDS	Attack	96.56%	96.39%	87.44%	91.7%	0.0091
	Benign		96.6%	99.09%	97.83%	0.1256

Table 8 CNN-GRU model performance for binary classification

Datasets	Class	Accuracy	Precision	Recall	F1 Score	FAR
NF-BoT-IoT	Attack	98.67%	98.69%	99.97%	99.32%	0.5598
	Benign		97.04%	44.02%	60.57%	0.0003
NF-ToN-IoT	Attack	92.83%	92.63%	99.25%	95.82%	0.382
	Benign		94.43%	61.8%	74.71%	0.0075
NF-UNSW-NB15	Attack	98.55%	85.98%	78.66%	82.16%	0.0057
	Benign		99.06%	99.43%	99.24%	0.2134
NF-CSE-CIC-IDS2018	Attack	97.97%	91.51%	75.82%	82.93%	0.0049
	Benign		98.34%	99.51%	98.92%	0.2418
NF-UQ-NIDS	Attack	96.32%	93.83%	88.9%	91.3%	0.0162
	Benign		96.97%	98.38%	97.67%	0.111

Table 9 CNN-BiGRU model performance for binary classification

Datasets	Class	Accuracy	Precision	Recall	F1 Score	FAR
NF-BoT-IoT	Attack	98.69%	98.7%	99.98%	99.33%	0.5558
	Benign		97.69%	44.42%	61.07%	0.0002
NF-ToN-IoT	Attack	93.87%	93.7%	99.28%	96.41%	0.3228
	Benign		95.12%	67.72%	79.12%	0.0072
NF-UNSW-NB15	Attack	98.62%	86.07%	80.5%	83.19%	0.0058
	Benign		99.14%	99.42%	99.28%	0.195
NF-CSE-CIC-IDS2018	Attack	95.88%	90.11%	40.95%	50.31%	0.0031
	Benign		96.05%	99.69%	97.84%	0.5905
NF-UQ-NIDS	Attack	95.72%	91.99%	87.94%	89.92%	0.0213
	Benign		96.7%	97.87%	97.28%	0.1206

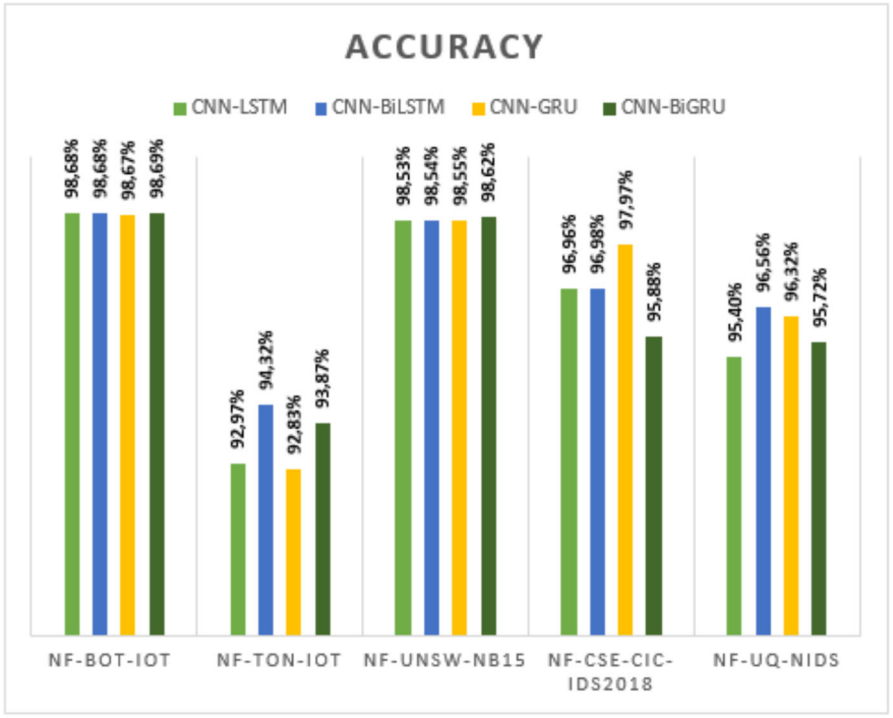


Fig. 19 Accuracy comparison

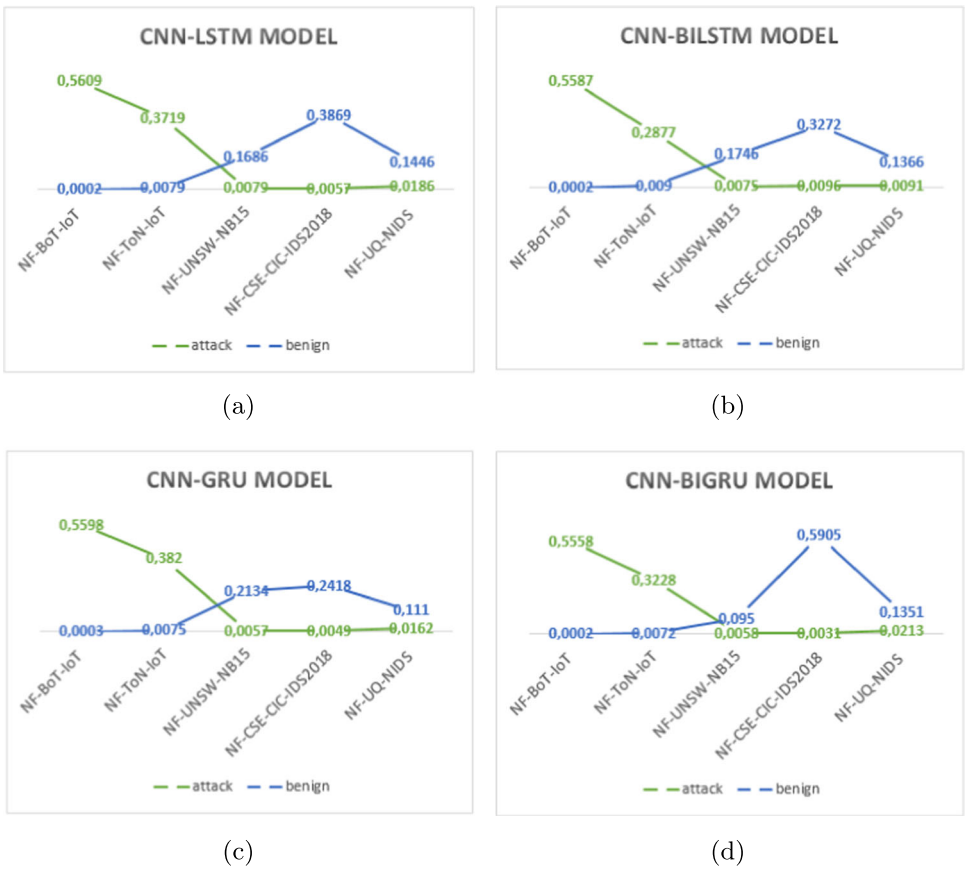


Fig. 20 FAR comparison

Table 10 Comparison with the previous existing methods in the same datasets

Method	Dataset	Accuracy
Extra Trees [23]	NF-BoT-IoT	93.82%
E-GraphSAGE [67]		93.57%
Our CNN-BiGRU		**98.69%**
Stacked MLP [69]	NF-ToN-IoT	90.17%
2D-ACNN [70]		90.10%
Our CNN-BiLSTM		**94.32%**
1D CNN [26]	NF-UNSW-NB15	98.32%
Our CNN-BiGRU		**98.62%**
Extra Trees [23]	NF-CSE-CIC-IDS2018	95.33%
GNN [68]		95.79%
Stacked MLP [69]		97.26%
Our CNN-GRU		**97.97%**
2D-ACNN [70]	NF-UQ-NIDS	95.20%
Our CNN-BiLSTM		**96.56%**

dataset, achieving an accuracy of approximately 94.32% compared to their accuracies of 90.17% each. On the NF-CSE-CIC-IDS2018 and NF-UQ-NIDS datasets, our models demonstrated competitive performance with stacked MLP [68], 2D-ACNN [69], and GNN [70]. These comparative experiments underscore the superior performance of our models, highlighting their potential to significantly enhance network IDS.

6 Conclusion and future work

Research on intrusion detection technology in the Internet environment is crucial for enhancing cybersecurity defenses and mitigating the increasing threats in cyberspace. Integrating DL techniques into IDS holds significant promise for improving detection accuracy and reducing false alarm rates caused by high-dimensional data. Addressing this challenge, this paper proposes an IDS model that combines CNN with four RNN variants to overcome issues such as difficult feature extraction, low detection rates, and poor generalization abilities of traditional IDS. To achieve this, we developed four models: CNN-LSTM, CNN-BiLSTM, CNN-GRU, and CNN-BiGRU. The CNN is utilized for extracting spatial traffic features, while the RNN addresses the time-series features of the datasets. LSTM, BiLSTM, GRU, and BiGRU are chosen for their ability to mitigate the problem of vanishing gradient and long-term dependencies compared to traditional RNNs. Experiments were conducted on Netflow-based datasets, specifically NF-UQ-NIDS, which combines four well-known datasets: NF-BoT-IoT, NF-ToN-IoT, NF-UNSW-NB15, and NF-CSE-CIC-IDS2018, for binary classification. Our evaluations involved a comparative study of four DL models on each dataset using various performance measures such as accuracy, recall, precision, F1 score, FAR and AUC. Results demonstrate the effectiveness of our approaches compared to existing DL methods in recent years, with our models achieving high accuracy of up to 98.69% with low FAR. Additionally, all models successfully detected both Benign and Attack classes.The methodology can also be extended to other cybersecurity applications, such as anomaly detection in cloud computing and software-defined networks (SDN). Future work will focus on reducing training time by exploring optimization techniques tailored to each dataset, thereby enhancing the overall performance of these models in real-world applications. Moreover, the study plans to investigate other DL methods using different real-world datasets for IDS in IoT networks. In addition, exploring advanced architectures, such as transformer-based models and attention mechanisms, could further enhance the detection capability of IDS in complex network.

Author Contributions R.J. developed the method's concept and implementation, performed the experiments and analysis, and wrote the manuscript. N.L. provided supervision and reviewed the final manuscript.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Funding Not applicable.

Conflict of Interest The authors declare no conflict of interest.

Consent to Participate Not applicable.

Consent for Publication Not applicable.

Competing interests The authors declare no competing interests.

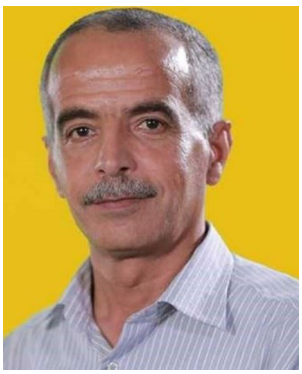
References

1. Babun L, Denney K, Celik ZB, McDaniel P, Uluagac AS (2021) A survey on iot platforms: communication, security, and privacy perspectives. Comput Netw 192:108040
2. Wu F, Lyu F, Ren J, Yang P, Qian K, Gao S, Zhang Y (2023) Characterizing internet card user portraits for efficient churn prediction model design. IEEE Trans Mob Comput 23(2):1735–1752
3. Sobin C (2020) A survey on architecture, protocols and challenges in iot. Wirel Pers Commun 112(3):1383–1429
4. Bhuiyan MN, Rahman MM, Billah MM, Saha D (2021) Internet of things (iot): A review of its enabling technologies in healthcare applications, standards protocols, security, and market opportunities. IEEE Internet Things J 8(13):10474–10498
5. Tong Z, Ye F, Yan M, Liu H, Basodi S (2021) A survey on algorithms for intelligent computing and smart city applications. Big Data Min Anal 4(3):155–172
6. Qays MO, Ahmad I, Abu-Siada A, Hossain ML, Yasmin F (2023) Key communication technologies, applications, protocols and future guides for iot-assisted smart grid systems: a review. Energy Rep 9:2440–2452
7. Babangida L, Perumal T, Mustapha N, Yaakob R (2022) Internet of things (iot) based activity recognition strategies in smart homes: a review. IEEE Sens J 22(9):8327–8336

8. Madhiarasan M (2021) Design and development of iot based solar powered versatile moving robot for military application. Int J Syst Assur Eng Manag 12(3):437–450
9. Mohy-Eddine M, Guezzaz A, Benkirane S, Azrour M, Farhaoui Y (2023) An ensemble learning based intrusion detection model for industrial iot security. Big Data Min Anal 6(3):273–287
10. Smmarwar SK, Gupta GP, Kumar S (2024) Android malware detection and identification frameworks by leveraging the machine and deep learning techniques: a comprehensive review. Telemat Inform Rep 100130
11. Smmarwar SK, Gupta GP, Kumar S (2022) Deep malware detection framework for iot-based smart agriculture. Comput Electr Eng 104:108410
12. Anderson JP (1980) Computer security threat monitoring and surveillance. James P. Anderson Company, Technical Report
13. Thakkar A, Lohiya R (2022) A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. Artif Intell Rev 55(1):453–563
14. Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Hasan M, Van Essen BC, Awwal AA, Asari VK (2019) A state-of-the-art survey on deep learning theory and architectures. Electronics 8(3):292
15. Kumar P, Jolfaei A, Islam AN (2025) An enhanced deep-learning empowered threat-hunting framework for software-defined internet of things. Comput Secur 148:104109
16. Al-Garadi MA, Mohamed A, Al-Ali AK, Du X, Ali I, Guizani M (2020) A survey of machine and deep learning methods for internet of things (iot) security. IEEE Commun Surv Tutor 22(3):1646–1685
17. Do NQ, Selamat A, Krejcar O, Herrera-Viedma E, Fujita H (2022) Deep learning for phishing detection: Taxonomy, current challenges and future directions. IEEE Access 10:36429–36463
18. Jablaoui R, Liouane N (2024) Efficient rnn models for iot intrusion detection system. In: 2024 International conference on control, automation and diagnosis (ICCAD). IEEE, pp 1–6
19. Yang Z, Liu X, Li T, Wu D, Wang J, Zhao Y, Han H (2022) A systematic literature review of methods and datasets for anomaly-based network intrusion detection. Comput Secur 116:102675
20. Gümüşbaş D, Yıldırım T, Genovese A, Scotti F (2020) A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. IEEE Syst J 15(2):1717–1731
21. Jablaoui R, Liouane N (2024) An effective deep cnn-lstm based intrusion detection system for network security. In: 2024 International conference on control, automation and diagnosis (ICCAD). IEEE, pp 1–6
22. Zhong W, Yu N, Ai C (2020) Applying big data based deep learning system to intrusion detection. Big Data Min Anal 3(3):181–195
23. Sarhan M, Layeghy S, Moustafa N, Portmann M (2021) Netflow datasets for machine learning-based network intrusion detection systems. In: Big data technologies and applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings 10. Springer, pp 117–135
24. Sarhan M, Layeghy S, Portmann M (2022) Towards a standard feature set for network intrusion detection system datasets. Mobile networks and applications, 1–14
25. Altaf T, Wang X, Ni W, Yu G, Liu RP, Braun R (2023) A new concatenated multigraph neural network for iot intrusion detection. Internet of Things. 22:100818
26. Vishwakarma M, Kesswani N (2023) A transfer learning based intrusion detection system for internet of things
27. Sayed N, Shoaib M, Ahmed W, Qasem S, Albarrak A, Saeed F (2022) Augmenting iot intrusion detection system performance using deep neural network. Comput Mater Contin 74(1):1351–1374
28. Smmarwar SK, Gupta GP, Kumar S (2023) Xai-amd-dl: An explainable ai approach for android malware detection system using deep learning. In: 2023 IEEE World conference on applied intelligence and computing (AIC). IEEE, pp 423–428
29. Cao B, Li C, Song Y, Fan X et al (2022) Network intrusion detection technology based on convolutional neural network and bigru. Comput Intell Neurosci 2022
30. Hnamte V, Hussain J (2023) Dcnnbilstm: An efficient hybrid deep learning-based intrusion detection system. Telemat Inform Rep 10:100053
31. Smmarwar SK, Gupta GP, Kumar S, Kumar P (2022) An optimized and efficient android malware detection framework for future sustainable computing. Sustain Energy Technol Assess 54:102852
32. Popoola SI, Adebisi B, Ande R, Hammoudeh M, Atayero AA (2021) Memory-efficient deep learning for botnet attack detection in iot networks. Electronics 10(9):1104
33. Wang Y-C, Houng Y-C, Chen H-X, Tseng S-M (2023) Network anomaly intrusion detection based on deep learning approach. Sensors 23(4):2171
34. Li Z, Wang P, Wang Z (2024) Flowgananomaly: Flow-based anomaly network intrusion detection with adversarial learning. Chin J Electron 33(1):58–71
35. Kumar P, Javeed D, Islam AN, Luo XR (2025) Deepsecure: A computational design science approach for interpretable threat hunting in cybersecurity decision making. Decis Support Syst 188:114351
36. Li Z, Liu F, Yang W, Peng S, Zhou J (2021) A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Trans Neural Netw Learn Syst 33(12):6999–7019
37. Reddy SR, Varma GS, Davuluri RL (2023) Optimized convolutional neural network model for plant species identification from leaf images using computer vision. Int J Speech Technol 26(1):23–50
38. Thanki R (2023) A deep neural network and machine learning approach for retinal fundus image classification. Healthc Anal 3:100140
39. Amjoud AB, Amrouch M (2023) Object detection using deep learning, cnns and vision transformers: a review. IEEE Access
40. Al-Turaiki I, Altwaijry N (2021) A convolutional neural network for improved anomaly-based network intrusion detection. Big Data 9(3):233–252
41. Khan A, Sohail A, Zahoora U, Qureshi AS (2020) A survey of the recent architectures of deep convolutional neural networks. Artif Intell Rev 53:5455–5516
42. Aleesa A, Zaidan B, Zaidan A, Sahar NM (2020) Review of intrusion detection systems based on deep learning techniques: coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions. Neural Comput Appl 32:9827–9858
43. Udas PB, Karim ME, Roy KS (2022) Spider: A shallow pca based network intrusion detection system with enhanced recurrent neural networks. J King Saud Univ - Comput Inf Sci 34(10):10246–10272
44. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
45. Smagulova K, James AP (2019) A survey on lstm memristive neural network architectures and applications. Eur Phys J Spec Top 228(10):2313–2324
46. Ma Z, Li J, Song Y, Wu X, Chen C et al (2022) Network intrusion detection method based on fcwgan and bilstm. Comput Intell Neurosci 2022
47. Pooja T, Shrinivasacharya P (2021) Evaluating neural networks using bi-directional lstm for network ids (intrusion detection systems) in cyber security. Glob Transit Proc 2(2):448–454
48. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555

49. Singh NB, Singh MM, Sarkar A, Mandal JK (2021) A novel wide & deep transfer learning stacked gru framework for network intrusion detection. J Inf Secur Appl 61:102899
50. Wang S, Shao C, Zhang J, Zheng Y, Meng M (2022) Traffic flow prediction using bi-directional gated recurrent unit method. Urban Inform 1(1):16
51. Yu H, Kang C, Xiao Y, Ting Y (2023) Network intrusion detection method based on hybrid improved residual network blocks and bidirectional gated recurrent units. IEEE Access
52. Liu Y, Pu H, Sun D-W (2021) Efficient extraction of deep image features using convolutional neural network (cnn) for applications in detecting and analysing complex food matrices. Trends Food Sci Technol 113:193–204
53. Xiao Z, Xu X, Xing H, Luo S, Dai P, Zhan D (2021) Rtfn: A robust temporal feature network for time series classification. Inf Sci 571:65–86
54. Sarhan M, Layeghy S, Portmann M (2022) Towards a standard feature set for network intrusion detection system datasets. Mob Netw Appl, 1–14
55. Çetin V, Yildiz O (2022) A comprehensive review on data preprocessing techniques in data analysis. Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi 28(2):299–312
56. Fan C, Chen M, Wang X, Wang J, Huang B (2021) A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data. Front Energy Res 9:652801
57. Hudnurkar S, Rayavarapu N (2022) Binary classification of rainfall time-series using machine learning algorithms. Int J Electr Comput Eng 12(2):1945–1954
58. Hasnain M, Pasha MF, Ghani I, Imran M, Alzahrani MY, Budiarto R (2020) Evaluating trust prediction and confusion matrix measures for web services ranking. IEEE Access 8:90847–90861
59. Canbek G, Taskaya Temizel T, Sagiroglu S (2021) Benchmetrics: A systematic benchmarking method for binary classification performance metrics. Neural Comput & Applic 33(21):14623–14650
60. Devarakonda A, Naumov M, Garland M (2017) Adabatch: Adaptive batch sizes for training deep neural networks. arXiv:1712.02029
61. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv:1412.6980
62. Iiduka H (2021) Appropriate learning rates of adaptive learning rate optimization algorithms for training deep neural networks. IEEE Trans Cybern 52(12):13250–13261
63. Banerjee C, Mukherjee T, Pasiliao E Jr (2019) An empirical study on generalizations of the relu activation function. In: Proceedings of the 2019 ACM southeast conference, pp 164–167
64. Wang M, Lu S, Zhu D, Lin J, Wang Z (2018) A high-speed and low-complexity architecture for softmax function in deep learning. In: 2018 IEEE Asia pacific conference on circuits and systems (APCCAS). IEEE, pp 223–226
65. Riehl K, Neunteufel M, Hemberg M (2023) Hierarchical confusion matrix for classification performance evaluation. J R Stat Soc Ser C Appl Stat 72(5):1394–1412
66. Søreide K (2009) Receiver-operating characteristic curve analysis in diagnostic, prognostic and predictive biomarker research. J Clin Pathol 62(1):1–5
67. Lo WW, Layeghy S, Sarhan M, Gallagher M, Portmann M (2022) E-graphsage: A graph neural network based intrusion detection system for iot. In: NOMS 2022-2022 IEEE/IFIP Network operations and management symposium. IEEE, pp 1–9
68. Xu R, Wu G, Wang W, Gao X, He A, Zhang Z (2024) Applying self-supervised learning to network intrusion detection for network flows with graph neural network. arXiv:2403.01501
69. Krishnan D, Shrinath P (2024) Robust botnet detection approach for known and unknown attacks in iot networks using stacked multi-classifier and adaptive thresholding. Arab J Sci Eng, 1–17
70. Nizamudeen SMT (2023) Intelligent intrusion detection framework for multi-clouds-iot environment using swarm-based deep learning classifier. J Cloud Comput 12(1):134

Rahma Jablaoui Received the Master degree in Electronics and Microelectronics from the Faculty of sciences of Monastir (FSM), University of Monastir, Tunisia. Currently Ph.D. student at the Faculty of sciences of Monastir (FSM) and the National Engineering School of Monastir Tunisia (ENIM), University of Monastir, Tunisia. His current research interests include Artificial Intelligence, deep learning and anomaly detection models for the Internet of Things and embedded system.



Noureddine Liouane is professor in the Engineering Department at the ENIM-Monastir. He received a MSEE degree in 1988 from the ENSET at Tunis University and the Ph.D. degree from Ecole Centrale de Lille, France in 1998. His research interests include evolutionary optimization methods for events systems, computer science and operational research.