Enhancing IoT Security: A Comparative Study of CNN and RNN-Based Anomaly Detection Using the CICIoT2023 Dataset

Amal M. Al-Ghamdi, and Marwah M. Alansari

***Abstract*— The rapid growth of Internet of Things (IoT) devices has resulted in numerous security challenges. Anomaly detection has become a crucial aspect of IoT security, as it helps identify unusual behaviors and potential threats within IoT networks. Deep learning techniques have shown considerable potential in recognizing anomalies across various domains. However, the dynamic and continuously developing nature of cyberattacks complicates the process of monitoring and identifying among different types of attacks effectively. This study aims to enhance IoT security by conducting a comparative analysis of two deep learning algorithms: convolutional neural networks (CNNs) and recurrent neural networks (RNNs). We deploy these models on the new CICIoT2023 dataset to detect large-scale attacks targeting IoT devices. Our findings reveal that the CNN model outperforms others with an accuracy of 98%, while the BiLSTM model achieves 85%, and the hybrid CNN-BiLSTM model reaches 94%. These results show that CNNs are better at extracting spatial features and that hybrid models might be able to use both spatial and temporal dependencies. By providing insights into model selection and optimization, this research contributes to the development of robust intrusion detection systems for IoT security.**

***Index Terms*—convolutional neural network, recurrent neural network, IoT security, comparative analysis, deep learning**

I. INTRODUCTION

The rapid development and widespread adoption of the Internet of Things (IoT) have made it one of the most prominent and rapidly growing fields in recent years. Consequently, IoT has emerged as a significant area of study within academic and research communities. IoT refers to a pervasive network where diverse objects and entities in our environment are interconnected through the Internet, enabling seamless communication and interaction without human intervention. This connectivity includes a diverse range of objects linked through wireless and wired networks, each equipped with unique addressing mechanisms. Such interconnectedness facilitates collaborative efforts and information exchange, fostering the development of innovative applications and services. Remarkable examples include smart homes, intelligent transportation systems, connected vehicles, advanced power grids, smart cities, and efficient traffic management systems [1]. As a result, IoT has gained considerable traction and is already transforming various industries.

The IoT foresees a transformation of daily life through the proliferation of billions of smart devices, estimated to reach approximately 75 billion by 2025, aimed at automating routine tasks [2]. This transformative technology holds vast potential to enhance human life across diverse sectors, including healthcare, transportation, agriculture, education, and numerous business applications [3]. By 2020, projections indicate that the global IoT market will reach $457.29 billion [4], with an estimated 50 billion interconnected IoT devices [5], highlighting the anticipated expansion and the critical need to secure these systems.

As with other emerging technologies, security challenges present significant obstacles to the effective deployment of IoT systems. Addressing these challenges is essential for instilling public confidence in IoT devices and encouraging broader adoption. As these devices increasingly integrate into daily life, it is crucial to ensure robust protection against various known threats [1]. The IoT ecosystem benefits from diverse connectivity options, such as Wi-Fi, 4G/5G, Bluetooth, cloud services, and advanced analytics. However, these advancements also introduce new challenges that require careful consideration [4].

This study addresses the growing number of cyberattacks targeting IoT devices, driven by their rapid expansion. There is an increasing demand for robust detection methods utilizing machine learning and deep learning approaches to identify malicious and anomalous activities in the IoT domain. Effective techniques are crucial for minimizing security vulnerabilities and safeguarding IoT devices from potential threats. Moreover, there is a need to compare and evaluate different techniques to enhance IoT security. This research aims to assess the effectiveness of deep learning techniques for anomaly cybersecurity datasets available for developing machine learning classifiers for IoT anomaly detection has increased, with noteworthy examples including BoT-IoT, CIC2019DDoS, and BoT-IoT 2020. CICIoT2023, a newly released dataset, offers extensive records across 33 categories of attacks targeting IoT environments, offering greater accuracy compared to earlier datasets. The primary objective of this study is to compare two specific deep

Manuscript received September 2, 2024; revised February 22, 2025.

Amal M. Al-Ghamdi is a postgraduate student in Cyber Security, at College of Computing and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia (e-mail: g210003945@seu.edu.sa).

Marwah M. Alansari is an assistant professor in Information Technology Department, College of Computing and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia (corresponding author to provide phone: 00966115157000; e-mail: m.alansari@seu.edu.sa).

learning models for anomaly detection in IoT networks using the CICIoT2023 dataset. Through comprehensive comparative analysis, this study seeks to identify the strengths, weaknesses, and performance characteristics of these models. The contributions of this research include:

1) Designing an anomaly detection model for IoT networks using convolutional neural networks (CNNs) and recurrent neural networks (RNNs) based on CICIoT2023.
2) Building models from scratch for designing multiclass classifiers and evaluating their performance.
3) Investigating the effectiveness and reliability of the new real-time dataset, CICIoT2023.

The findings of our research can provide valuable insights to researchers and practitioners, guiding them in the selection and implementation of effective anomaly detection methods for IoT networks.

The remainder of this paper is organized as follows: Section II provides a brief overview of the two deep learning techniques used in our research: CNNs and RNNs. Section III outlines the research methodology employed for the comparative analysis. In Section IV, we discuss related work. Section V summarizes machine learning and deep learning methods for classifying attacks on IoT platforms. Section VI presents an overview of two common IoT attack datasets and compares them. Section VII details the implementation of a multiclass CNN and a type of RNN using the CICIoT2023 dataset. In Section VIII, we present and discuss our detailed results. Finally, Section IX outlines our conclusions and highlights future directions for research.

II. DEEP LEARNING BACKGROUND

Deep learning, a subset of machine learning, facilitates the acquisition of knowledge from prior experience [6]. It has become one of the most prominent topics in computer science [7], leveraging artificial neural networks and other machine learning algorithms that mimic the structure of biological neural networks. These algorithms consist of multiple layers that enable feature extraction, transformation, and pattern analysis using supervised or unsupervised learning methods. Inspired by the human brain, deep learning models are capable of processing, labeling, and categorizing input data to derive meaningful insights. This technique excels in handling large and complex datasets, surpassing the capabilities of traditional machine learning algorithms. Moreover, deep learning can operate without supervision and effectively manage unstructured and unlabeled data.

The term ‘deep’ refers to the large number of layers in these models. Typically, deep learning models comprise three types of layers: an input layer that receives data, hidden layers responsible for extracting patterns, and an output layer that generates results. The output of one layer becomes the input for the subsequent layer, establishing a hierarchical learning structure [6]. Deep learning has led to substantial advancements across various fields, resulting in transformative impacts on industries and businesses. Its success has sparked a global race among leading economies and technology companies to explore and push the boundaries of deep learning further. In certain areas, deep learning has even demonstrated performance that surpasses human capabilities [8]. Furthermore, deep learning has enhanced industrial applications by automating tasks that were previously labor-intensive, contributing to its rising popularity in the digital age.

Deep learning models have been particularly effective in understanding complex structures across diverse domains, including natural language processing, speech recognition, image recognition, and video analysis [6]. These models offer a range of architectures, each with specific applications and compatibility. In the context of cybersecurity, deep learning provides powerful tools for designing effective detection systems. This study discusses two typical deep learning models:

1) Convolutional Neural Networks (CNNs): CNNs have emerged as a widely adopted deep learning technique across various domains, particularly in computer vision, where they excel in simulating the human visual system [6]. CNNs consist of convolutional and pooling layers. The convolutional layers extract essential features from the input data, while the pooling layers enhance the generalization of these features. CNNs operate on two-dimensional (2D) data, which involves converting input data into matrix form for efficient anomaly detection [9]. This capability makes CNNs well-suited for applications involving image processing and visual pattern recognition.

2) Recurrent Neural Networks (RNNs): RNNs are designed to analyze time-series and sequential data by incorporating previous outputs as inputs for current processing. This feedback loop makes RNNs particularly effective for tasks involving temporal sequences, such as object and human tracking in video footage. A specific type of RNN, known as a Long Short-Term Memory (LSTM) network, includes a memory cell that helps retain information over time, allowing it to capture long-term dependencies in sequential data. LSTM networks are capable of learning temporal and sequential patterns from complex data sequences, making them ideal for applications in speech recognition and anomaly detection [6].

III. RESEARCH METHODOLOGY

In this study, we use Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) because they work well and have been used successfully in many areas, such as finding problems in IoT networks. Specifically, we aim to construct these models and evaluate their performance using established evaluation metrics. The research methodology consists of the following stages:

1) *Comparative Analysis of Deep Learning Models:* Our research focuses on conducting a comparative analysis of two distinct deep learning models for classifying various types of attacks on IoT platforms. We aim to build robust deep learning models using the recently published IoT attack dataset, CICIoT2023. The selection of CNN and RNN models was based on their demonstrated capability to accurately identify and classify anomalies within the

network, as well as insights gathered from the literature review. Fig. 1 depicts the research flowchart and outlines the methodology stages.

2) *Review of Previous Studies:* The objective of this stage was to identify commonly used machine learning models for detecting attacks on IoT platforms. By reviewing existing research, we gathered valuable insights into the models that have shown promising results in IoT anomaly detection.

3) *Analysis of Existing IoT Attack Datasets:* In this stage, we focused on examining frequently used datasets in prior studies to ensure the effectiveness of the proposed models. Our goal was to select a realistic and recently published dataset that would provide representative results. After careful consideration, we chose the CICIoT2023 dataset for our study due to its comprehensive coverage of 33 categories of attacks. Section VI provides detailed information on IoT attack datasets.

4) *Data Preprocessing:* The quality of training data significantly influences the effectiveness of deep learning models. IoT datasets often include data from diverse sensors across multiple domains, leading to challenges such as inconsistent data distribution. To address these issues, we performed extensive data preprocessing, including error correction, normalization, and feature extraction. These steps were essential to ensure the integrity of the dataset and establish a solid foundation for model training and analysis.

5) *Model Development:* We developed three deep learning models: a CNN built from scratch, an RNN model utilizing a Bidirectional Long Short-Term Memory (BiLSTM) architecture, and a hybrid model combining CNN and BiLSTM techniques. We trained and evaluated all models using the CICIoT2023 dataset. We chose BiLSTM for its proven ability to capture long-term dependencies in sequential data, which makes it well-suited for time-series anomaly detection.

6) *Comparative Performance Evaluation:* In this stage, we assessed the models using key performance metrics, including accuracy, precision, recall, and F1-score. We selected these metrics due to their capacity to thoroughly assess classification performance. We conducted a detailed comparison of the CNN, RNN (BiLSTM), and hybrid models based on these metrics to identify their relative strengths and weaknesses.

IV. RELATED WORK

Extensive research literature has utilized machine learning and deep learning algorithms for detecting attacks on IoT platforms, owing to their capability to analyze complex, large-scale datasets effectively. These methods typically fall into two categories: binary classifiers, which differentiate between attacks and normal traffic, and multiclass classifiers, which aim to distinguish among various types of attacks. In the context of IoT security, we now provide a brief overview of several commonly used machine learning and deep learning models. We also discuss the strengths and limitations of these approaches to highlight the factors that affect their performance in IoT environments.

A. Machine learning approaches for IoT attack detection

Previous studies have extensively investigated the application of machine learning methods to detect distributed denial-of-service (DDoS) attacks on IoT devices. One study explored the use of three machine learning algorithms: decision tree, k-nearest neighbors (KNN), and naïve Bayes. These algorithms were employed to classify network traffic as either benign or indicative of a DDoS attack. The experiments utilized the CIC2019DDoS dataset, which included various DDoS attack types, such as UDP, DNS, SYN, and NetBIOS attacks. The results showed that the decision tree algorithm achieved an impressive accuracy rate of 100%, KNN reached an accuracy of 98%, while Naïve Bayes achieved a relatively lower accuracy of 29% [11]. Similarly, another study proposed a network intrusion detection approach using decision trees, a widely adopted machine learning algorithm for classification tasks. Decision trees recursively split features based on conditions that help determine the class labels of instances. The authors emphasized the importance of data quality enhancement through preprocessing and feature selection techniques to improve the performance of the decision tree classifier. They underscored the potential for integrating machine learning techniques into Intrusion Detection Systems (IDS) to enhance their capabilities [12].

Furthermore, [13] introduced a novel Intrusion Detection System (IDS) that was specifically designed for securing IoT networks. The proposed system utilized supervised machine learning algorithms, including KNN, logistic regression, multilayer perceptron, decision trees, and random forests. This study employed a unique feature set specifically designed for IoT environments, facilitating the accurate and automatic detection of various attacks, such as DDoS, DoS, reconnaissance, and information theft [13]. However, the study noted that machine learning algorithms often encounter challenges when handling large volumes of training data, particularly in maintaining high accuracy. Consequently, these models may struggle with scalability and performance when applied to large-scale IoT datasets.

In [35], they introduce an edge-enabled virtual honeypot-based intrusion detection system (EVHIDS) to secure Vehicle-to-Everything (V2X) networks. The system gets 99.01% accuracy and better detection rates for attacks like botnets (99.02%) and DoS (99.22%) by using a virtual honeypot (PotRSU) to attract and study threats and machine learning for real-time intrusion detection. It incorporates data preprocessing for optimized ML performance and enables collaboration between PotRSU, roadside units, and cloud servers for continuous threat updates. The EVHIDS

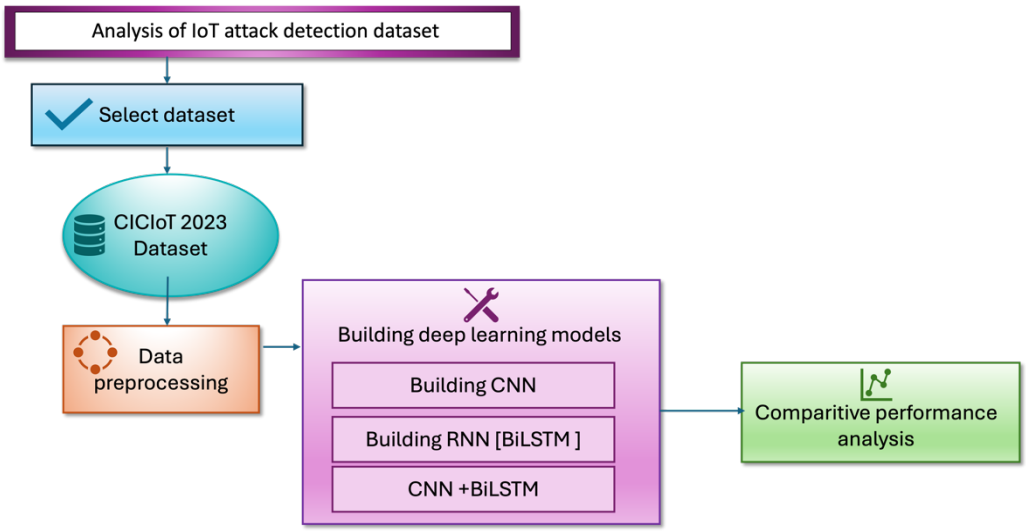


Fig. 1: Conceptual view of our research methodology

enhances V2X security, offering a scalable, adaptive solution for intelligent transportation systems.

B. Deep learning approaches for IoT attack detection

Existing Intrusion Detection Systems (IDSs) based on machine learning have shown promise but still face limitations, particularly in handling the complexities of IoT networks. Advanced deep learning techniques have been explored to address these limitations, enhancing anomaly detection and network security. Among these, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have demonstrated significant efficacy in network traffic classification tasks [14]. The proposal of a Pearson Correlation Coefficient-CNN (PCC-CNN) model addressed security concerns related to IoT communication protocols, achieving an impressive accuracy of 99.89% and a low misclassification rate of 0.001 [15]. Similarly, Mahmoud et al. [16] introduced a CNN-based anomaly detection model, delivering high precision, recall, and F1 scores. In addition to these efforts, Qian et al. [17] created an edge-cloud-based IDS that combines RNNs and BiLSTMs. This IDS performed better than models trained on the full set of attributes, showing how important it is to choose the right features to improve performance.

In the context of IoT-based electric vehicle charging stations, Kilichev et al. [18] presented an ensemble architecture combining CNNs, LSTMs, and GRUs. Their results showed exceptional accuracy in both binary and multiclass classification tasks. Furthermore, Meliboev et al. [19] showed that combining CNNs with LSTMs is better for finding network intrusions, which shows that hybrid models work effectively. Vinayakumar et al. [20] compared deep neural networks (DNNs) to classical machine learning classifiers and confirmed the advantage of DNNs in detecting advanced cyberattacks. Addressing class imbalance issues in datasets, Sharma et al. [21] introduced a filter-based DNN approach that leveraged Generative Adversarial Networks (GANs) to synthesize data for minority attack categories, improving multiclass classification accuracy from 85.0% to 91.0% using the UNSW-NB15 dataset. Other researchers have also explored dataset-specific IDS solutions; for instance, Vishwakarma et al. [23] created the NF-UQ-NIDS dataset to enhance real-time intrusion detection in IoT networks, effectively capturing a diverse range of attack behaviors.

Several studies have investigated novel approaches for addressing Distributed Denial of Service (DDoS) and Denial of Service (DoS) attacks in IoT systems. Ma et al. [26] suggested a CNN model that uses a multilayer convolution feature fusion mechanism and a categorical cross-entropy loss function. On the NSL-KDD dataset, this model was very accurate and had a low rate of false alarms. Similarly, Adefemi et al. [27] used a refined LSTM (RSTM) model to find DoS attacks. They improved performance by using preprocessing methods such as encoding, dimensionality reduction, and normalization. Ahmad et al. [28] further explored deep learning techniques, utilizing a DNN with mutual information to identify anomalies in the IoT-Botnet 2020 dataset, thereby outperforming traditional methods. Ullah et al. [29] built on their research from 2021 and suggested a more advanced framework for finding anomalies that combines LSTMs, BiLSTMs, GRUs, and CNNs to make feature learning better and consistency across scenarios. Their hybrid deep learning model demonstrated robust performance on seven diverse datasets. Saba et al. [30] developed a CNN-based IDS to analyze IoT network traffic, effectively identifying anomalies using the NID and BoT-IoT datasets. Lastly, Ajiboye et al. [36] examine the use of Autoencoder Neural Networks (AeNN) for dimensionality reduction to enhance Deep Neural Networks (DNN) in Intrusion Detection Systems (IDS) for IoT. Using the BoT-IoT dataset, the study develops a binary classification model to detect attacks like DoS, DDoS, and reconnaissance. Results show AeNN improves accuracy, with increases of up to 13.1% in some cases by enabling better feature selection. Despite minor declines in certain scenarios, the study concludes that combining AeNN and DNN offers an efficient, accurate solution for IoT network security [36].

Ultimately, these studies underscore the effectiveness of deep learning models in enhancing IoT network security. Techniques such as CNNs and RNNs have consistently delivered superior results, as evidenced by various research efforts. However, challenges like computational costs, data dependency, and class imbalance persist. Future research should focus on developing scalable, lightweight IDS solutions while addressing these limitations.

TABLE I SUMMARY FOR TYPES OF MACHINE LEARNING MODELS USED FOR DETECTING ATTACKS IN IOT SYSTEMS

Study	Year	Machine Learning							Classification type		Dataset
		DT	SVM	NB	KNN	RF	MLP	LR	Binary	Multi	
[11]	2021	x		x	x				x		CIC2019DDoS
[12]	2021	x							x		NSLKDD+ CICIDS2017
[13]	2021	x	x		x	x	x	x	x	x	BoT-IoT
[35]	2024	x	x	x	x	x		x	x		Simulated

V. ANALYSIS OF PREVIOUS DETECTION SYSTEMS

We summarize the machine learning and deep learning approaches presented in Section IV. Table I highlights the machine learning algorithms employed in earlier studies [11], [12], and [13]. Studies [11] and [12] proposed supervised machine learning algorithms to develop binary classifiers for detecting Distributed Denial of Service (DDoS) attacks. Among these, decision trees outperformed KNN and Naïve Bayes in [11], demonstrating superior accuracy and reliability. In [12], decision trees were the only algorithm explored, albeit using different datasets, as detailed in Table I. In [13], the BoT-IoT dataset was utilized as it provides a realistic benchmark for forensic analytics, capturing various IoT-specific attack scenarios. The machine learning algorithms in [13] were applied to build both binary and multiclass classifiers, offering a performance comparison of the proposed models. This research underscores the critical need for robust machine learning methodologies tailored to IoT environments to effectively detect and mitigate cyber threats. As IoT technologies continue to progress, it is crucial to conduct ongoing research to improve the accuracy, reliability, and scalability of IDSs.

Table II summarizes the deep learning approaches discussed in Subsection IV-B. CNN-based methods are the most commonly used models, as demonstrated in studies [14], [15], [16], [19], [24], [26], [28], [29], and [30]. These models have been mostly applied to binary classification tasks, though some have also been adapted for multiclass scenarios. The BoT-IoT dataset, a comprehensive IoT benchmark, is the most frequently utilized dataset for training and evaluating these classifiers, given its realistic attack simulations and relevance to IoT environments. Besides CNNs, hybrid deep learning models have been proposed, combining CNNs with RNNs, LSTMs, and GRUs to improve feature extraction and classification accuracy. These hybrid models demonstrate promising results in detecting a wide range of IoT attacks.

Deep learning models have shown significant advantages over traditional machine learning approaches in handling complex and high-dimensional IoT data. For instance, CNN architectures excel at automatic feature extraction and classification, enabling accurate detection of anomalies and cyber threats. However, these models are computationally intensive and require large amounts of labeled data, which can limit their practical application in resource-constrained IoT environments. Research efforts such as those in [24], [26], and [29] address these challenges by proposing optimized models and leveraging transfer learning to reduce dependency on large datasets. While machine learning methods like decision trees and KNN offer simplicity and computational efficiency, they struggle to match the accuracy and adaptability of deep learning approaches in dynamic IoT environments. The increasing complexity of IoT networks and the diversity of cyber threats necessitate further exploration of hybrid and ensemble learning models. Combining the strengths of both machine learning and deep learning approaches can offer a more balanced trade-off between performance and resource requirements.

Overall, these findings highlight the ongoing evolution of IDS methodologies for IoT networks. Future research should prioritize the development of lightweight, scalable models capable of real-time intrusion detection without compromising accuracy. Additionally, ulitlizing synthetic data generation techniques, such as Generative Adversarial Networks (GANs), can address class imbalance issues and enhance the robustness of IDS models. By addressing these challenges, the next generation of IDS solutions can effectively safeguard IoT ecosystems against emerging cyber threats.

VI. OVERVIEW OF IOT ATTACK DATASETS

Various IoT attack datasets have been utilized to develop machine learning and deep learning models for detecting attacks in IoT environments. In our study, we focused on two key datasets: CICIoT2023 and BoT-IoT. The CICIoT2023 dataset was employed to train and evaluate a range of deep learning models, while the BoT-IoT dataset was used during the comparative analysis stage to benchmark model performance. These datasets were selected due to their relevance and comprehensiveness in representing real-world IoT attack scenarios. CICIoT2023 offers updated attack patterns and traffic data, making it ideal for training robust models capable of addressing emerging threats. On the other hand, the BoT-IoT dataset, widely recognized for its diverse attack scenarios and realistic IoT traffic, serves as a reliable benchmark for comparative studies.

In the following section, we provide a detailed explanation of the CICIoT2023 and BoT-IoT datasets, followed by a comprehensive comparison to highlight their respective features, strengths, and limitations.

TABLE II SUMMARY OF COMMONLY USED DEEP LEARNING APPROACHES FOR ANOMALY DETECTION IN IOT SYSTEMS

Studies	Year	Deep learning approaches									Classification type		Dataset
		Supervised approaches				Unsupervised approaches				Hybrid approaches	Binary	Multi	
		ANN	CNN	RNN	DNN	DBM	RBM	DBN	DA				
[14]	2020		x	x	x	x	x	x	x		x	x	CSE-CIC-IDS2018 BoT-IoT
[15]	2023		x								x	x	CSE-CIC-IDS2018 BoT-IoT
[16]	2021		x								x	x	CSE-CIC-IDS2018 BoT-IoT
[28]	2021		x	x	x						x		IoT-Botnet 2020
[29]	2022		x	x						CN N+ RNN	x	x	NSL-KDD BoT-IoT IoT- NI IoT-23 MQTT MQTTSet IoT-DS2
[32]	2023			x						RNN+BiLSTM	x		BoT-IoT NSL-KDD
[18]	2024									CNN+LSTM+ GRU	x	x	Edge-IIoTset
[19]	2022		x	x						CNN+ LSTM	x		UNSW NB15 NSL-KDD KDDCup 99
[20]	2019				x						x	x	KDDCup 99 NSL-KDD UNSW-NB15 Kyoto WSN-DS CICIDS 2017
[22]	2023				x						x		Owned Dataset
[27]	2022			x							x		CICIDS-2017 NSL-KDS
[26]	2020		x								x		NSL-KDD
[25]	2021			x							x	x	NSL-KDD
[24]	2023	x	x	x						ANNs+CNNs +RNN	x	x	IoT-23
[30]	2022		x								x	x	NID BoT-IoT
[23]	2022				x						x	x	NF-BoT IoT NF-ToN-IoT NF-CSE CIC IDS2018 NF-UNSW NB15 NF-UQ-NIDS
[21]	2023				x					GANs+DNN		x	UNSW-NB15

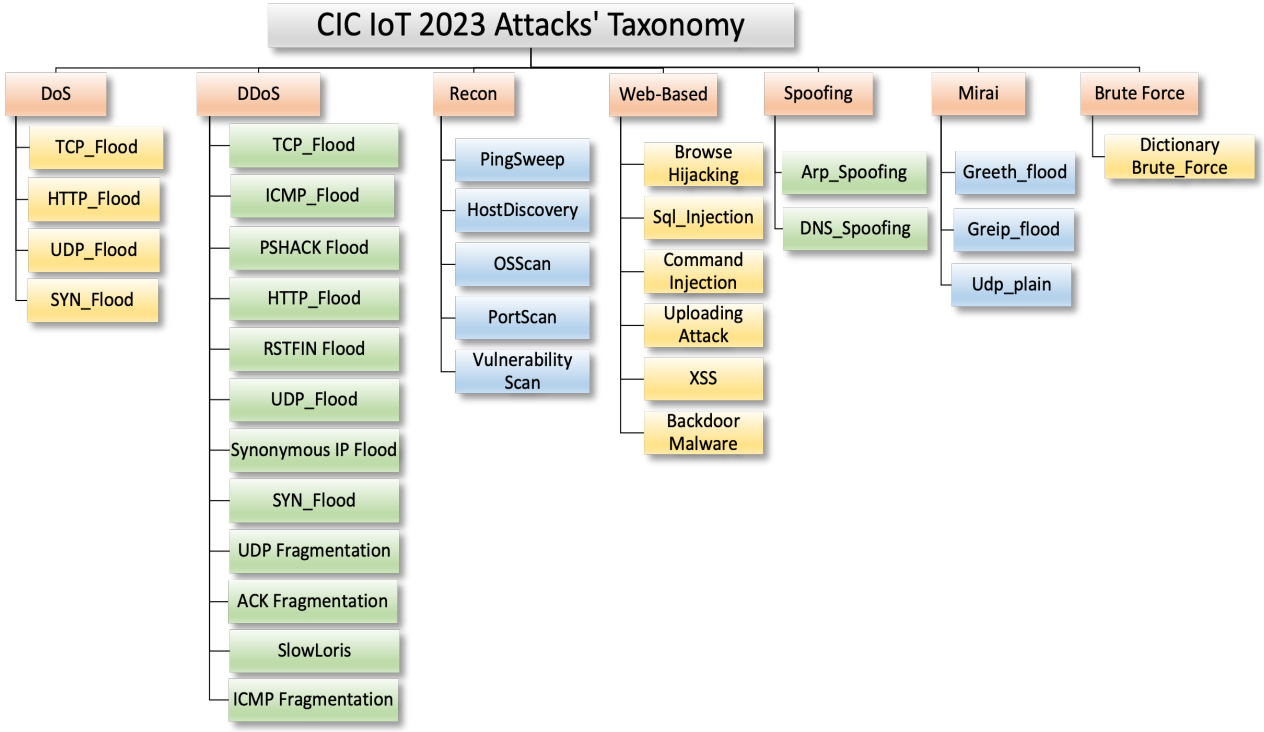


Fig. 2: Categories and types of attacks in the CICIoT2023 dataset

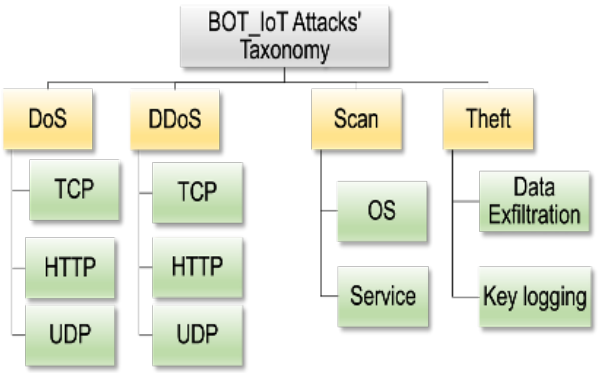


Fig. 3: Categories and types of attacks in the BoT-IoT dataset

A. *Overview of the CICIoT2023 Dataset*

The CICIoT2023 dataset [10] is a significant resource for IoT security research due to its unique and comprehensive features. As one of the largest and most recent datasets available in 2023, it provides researchers with a robust foundation for developing and evaluating intrusion detection systems. The dataset encompasses 33 distinct types of attacks, classified into seven categories, making it both extensive and diverse. A key feature of CICIoT2023 is its realistic nature, as it represents real-time IoT scenarios, simulating practical attack patterns and network behaviors. The dataset was generated using a meticulously designed testbed that included 105 IoT devices, alongside Zigbee and Z-Wave devices, to ensure diversity in network topology and attack simulations. This comprehensive setup allowed for the execution of 33 unique attacks on the network, leveraging other IoT devices to replicate realistic threat scenarios. These characteristics make the CICIoT2023 dataset particularly well-suited for analyzing and mitigating security vulnerabilities in modern IoT environments. Additionally, CICIoT2023 offers detailed documentation and taxonomy of the various attacks and their categories, as illustrated in Fig. 2. This taxonomy provides researchers and practitioners with a structured understanding of the attack landscape, further enhancing the dataset's utility in IoT security applications. By expanding a wide range of attack types and device interactions, CICIoT2023 sets a benchmark for comprehensiveness and realism in IoT dataset development.

B. *BoT-IoT Dataset*

The BoT-IoT dataset [31] was developed using a comprehensive testbed designed to simulate a realistic smart home IoT network environment. The testbed included a set of interconnected virtual machines (VMs) connected via a local area network (LAN) and the Internet, with the PFSense system serving as the gateway to provide Internet connectivity. To replicate the infrastructure of a real-world IoT network, an Ubuntu server was used to emulate IoT resources, while a Kali Linux VM was employed as the attack system to generate malicious traffic. The Ostinato tool simulated normal network activity, generating realistic traffic patterns to ensure a balanced dataset. To further enhance the dataset's realism, a smart home framework was developed, incorporating five IoT devices operating locally and connected to cloud services through a Node-RED system. This setup utilized the MQTT protocol for transmitting IoT messages to the cloud, closely mimicking modern IoT communication architectures. The result is a dataset that accurately reflects the network activity of a typical smart home environment, encompassing both benign and malicious traffic.

TABLE III COMPARISON OF CICIOT2023 DATASET WITH BOT-IOT DATASET [10]

Attack Type		CICIoT23 Dataset	BoT-IoT Dataset
DoS	TCP Flood	✓	✓
	HTTP Flood	✓	✓
	UDP Flood	✓	✓
	SYN Flood	✓	
DDoS	TCP Flood	✓	
	ICMP Flood	✓	
	PSHACK Flood	✓	
	HTTP Flood	✓	✓
	RSTFIN Flood	✓	
	UDP Flood	✓	✓
	Synonymous IP Flood	✓	
	SYN Flood	✓	
	UDP Fragmentation	✓	
	ACK Fragmentation	✓	
	Slow Loris	✓	
	ICMP Fragmentation	✓	
Recon	Ping Sweep	✓	
	Host Discovery	✓	
	OS Scan	✓	✓
	Port Scan	✓	✓
	Vulnerability Scan	✓	
Web-Based	Browse Hijacking	✓	
	SQL Injection	✓	
	Command Injection	✓	
	Uploading Attack	✓	
	XSS	✓	
	Backdoor Malware	✓	
Spoofing	Arp Spoofing	✓	✓
	DNS Spoofing	✓	
Mirai	Greeth flood	✓	
	Greip flood	✓	
	UDP plain	✓	
Brute Force	Dictionary Brute Force	✓	✓

TABLE IV DATASET PORTIONS FOR TRAINING, VALIDATION, AND TESTING

Category	Rows	Training (80%)	validation (10%)	Testing (10%)
DDos	33,984,560	27,187,648	3,398,456	3,398,456
Dos	8,090,738	6,472,590.4	809,073.8	809,073.8
Mirai	2,634,124	2,107,299.2	263,412.4	263,412.4
Benign	1,098,195	878,556	109,819.5	109,819.5
Spoofing	486,504	389,203.2	28,650.4	28,650.4
Recon	354,565	283,652	35,456.5	35,456.5
Web	24,829	19,863.2	2,482.9	2,482.9
Brute Force	13,064	10,451.2	1306.4	1306.4

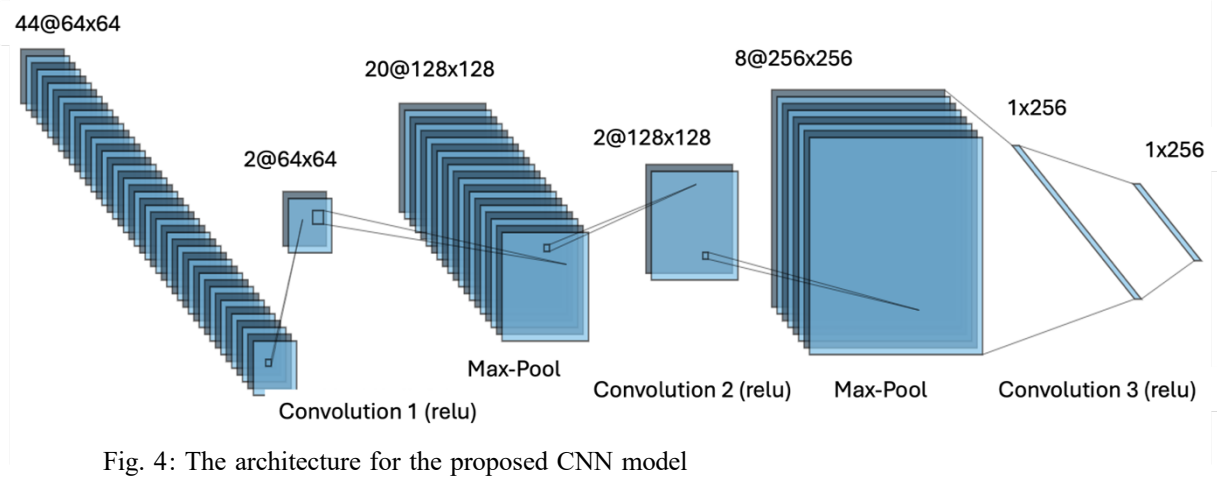


Fig. 4: The architecture for the proposed CNN model

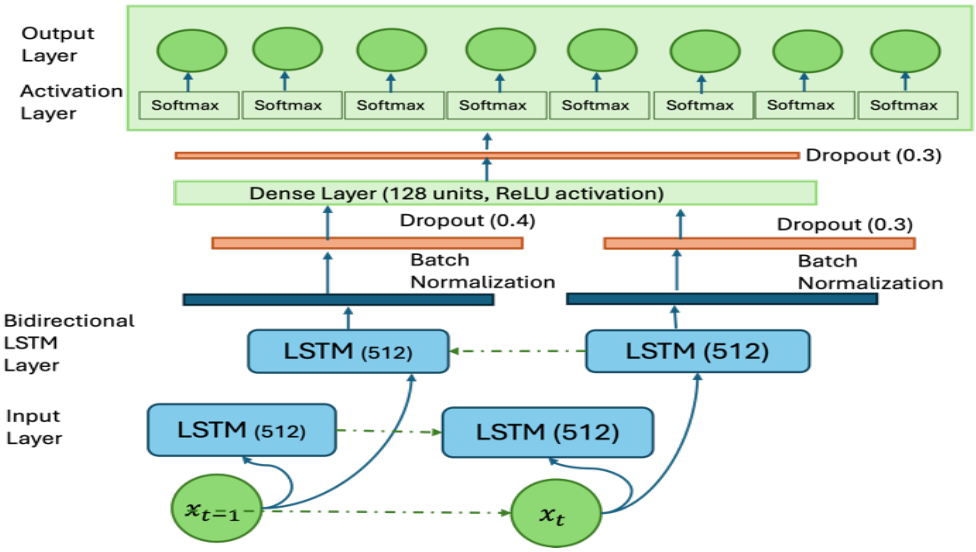


Fig. 5: The architecture for the proposed BiLSTM model

As illustrated in Fig. 3, the BoT-IoT dataset comprises four main attack categories, further subdivided into ten subcategories. This taxonomy provides a structured framework for understanding and analyzing different types of attacks, making the dataset a valuable resource for researchers and practitioners. Because it shows a wide range of real-life IoT network activity, the BoT-IoT dataset makes it possible to test and develop advanced intrusion detection and mitigation methods that are specifically designed to work in IoT settings.

C. CICIoT2023 vs. BoT-IoT

While both the CICIoT2023 and BoT-IoT datasets are valuable resources for IoT security research, the CICIoT2023 dataset [10] offers several remarkable advancements and improvements over the BoT-IoT dataset [31]. First, the CICIoT2023 dataset is based on an extensive topology comprising 105 real IoT devices, whereas the BoT-IoT dataset utilizes a smaller set of virtual machines (VMs) to emulate the IoT environment. By using real IoT devices, CICIoT2023 captures the complexities and refinements inherent in IoT networks, including interactions between various device types and protocols, such as ZigBee and Z-Wave. This realism ensures that the dataset more accurately reflects real-world IoT scenarios, enhancing its applicability for intrusion detection research. Second, the CICIoT2023 dataset encompasses a broader range of attack scenarios, with 33 distinct attack types classified into seven categories. In comparison, the BoT-IoT dataset focuses on four main attack categories with 10 subcategories, offering a less comprehensive representation of the threat landscape. Table III highlights the differences in attack coverage between the two datasets, illustrating how CICIoT2023 provides a more detailed and structured taxonomy of attacks. This taxonomy facilitates the development and evaluation of advanced intrusion detection and mitigation techniques by providing a clear categorization of attack types.

Moreover, the CICIoT2023 dataset includes attacks specifically designed to exploit the capabilities of other IoT devices, simulating realistic attack vectors observed in real-world scenarios. This approach enhances the dataset's practical relevance, as it reflects the evolving tactics and techniques employed by malicious actors targeting IoT systems. By providing a detailed and realistic representation of IoT threats, CICIoT2023 enables researchers to better understand the underlying patterns and characteristics of attacks, leading to the development of more targeted and

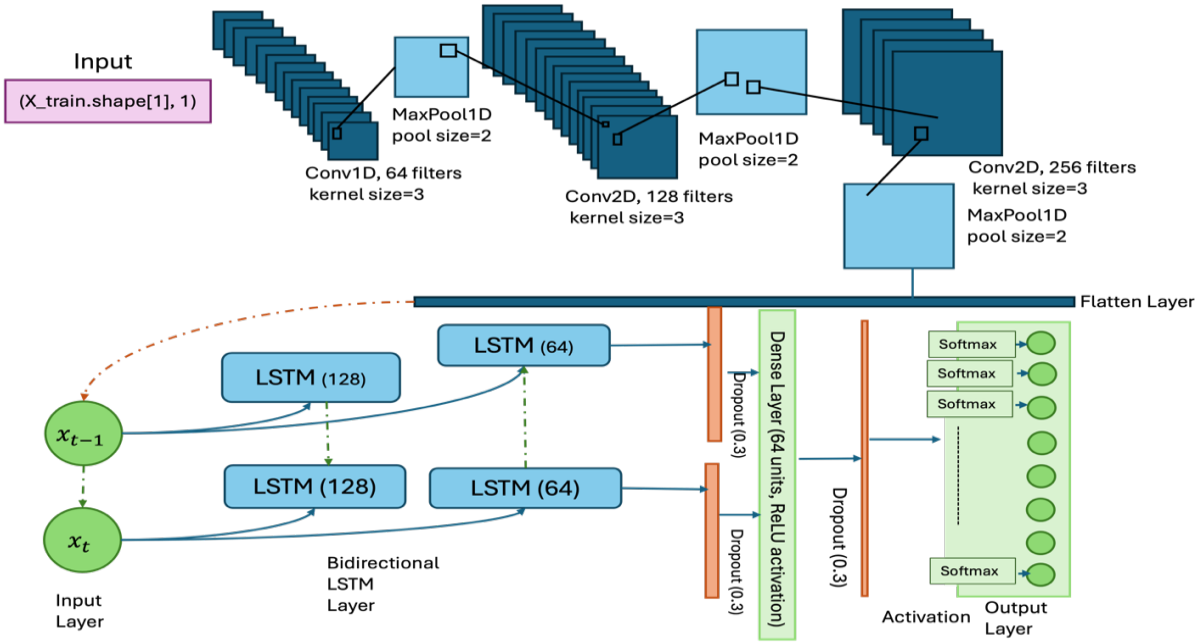


Fig. 6: The architecture for the proposed hybrid CNN-BiLSTM model

effective security solutions. In summary, the CICIoT2023 dataset stands out as a more comprehensive and realistic resource compared to the BoT-IoT dataset. Its extensive topology, diverse attack scenarios, and structured taxonomy make it an invaluable tool for addressing the complex and evolving challenges of IoT security.

VI. IMPLEMENTATION

For this experiment, we used Google Colab Pro with 334.6 GB of RAM and a 225.3 GB disk using TensorFlow and TPU v2 as hardware accelerators. Four phases were used to implement the proposed framework, as described in the following.

A. *Pre-processing of datasets*

The preprocessing of the CIC IoT 2023 dataset included consolidating attack labels, removing duplicates, and normalizing the numeric features, which are crucial for preparing the data in a manner that can effectively support the subsequent training and evaluation of deep learning models.

B. *The proposed CNN model architecture*

The proposed Convolutional Neural Network (CNN) model is designed with a typical layered architecture, consisting of convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for the final classification task. This architecture leverages hierarchical feature learning, allowing the network to capture both simple and complex patterns present in the input data, making it well-suited for the multiclass classification problem addressed in this study (details in Section II).

Fig. 4 illustrates the architecture of our proposed CNN model. Looking deeply in Fig. 4, *the input layer* of the model applies an initial convolution operation with 64 filters, each of size 3×1, to the input data. The ReLU activation function is employed to introduce non-linearity, allowing the network to learn complex patterns. The input shape is determined by the number of features in the training dataset, defined as *X_train.shape[1]*, with a single channel dimension, as the data is one-dimensional. The model architecture consists of three *convolutional blocks*. Each block is composed of a convolutional layer followed by a max-pooling layer:

- The number of filters in each convolutional layer increases progressively across the blocks: 64, 128, and 256, respectively. This incremental increase enables the model to capture a diverse set of features, ranging from simple edges to more complex patterns.
- The convolutional layers use the ReLU activation function, which introduces non-linearity and prevents issues like the vanishing gradient problem.

Max-pooling layers, specified as MaxPool1D (pool_size=2) as shown in Fig. 4, follow each convolutional layer. These layers down sample the feature maps by reducing the dimensionality along the time axis by a factor of 2. Max-pooling retains the most prominent features while discarding less significant information, reducing the computational load and minimizing the risk of overfitting. The output of the last convolutional block is passed through a *Flatten layer*, which reshapes the multi-dimensional feature maps into a one-dimensional vector. This transformation prepares the data for the fully connected layers, enabling the model to interpret the extracted features for classification purposes. The network includes four *densely connected layers*, which progressively reduce the number of neurons:

- The fully connected layers have *256, 128, and 64 neurons*, respectively, using the ReLU activation function for non-linearity.

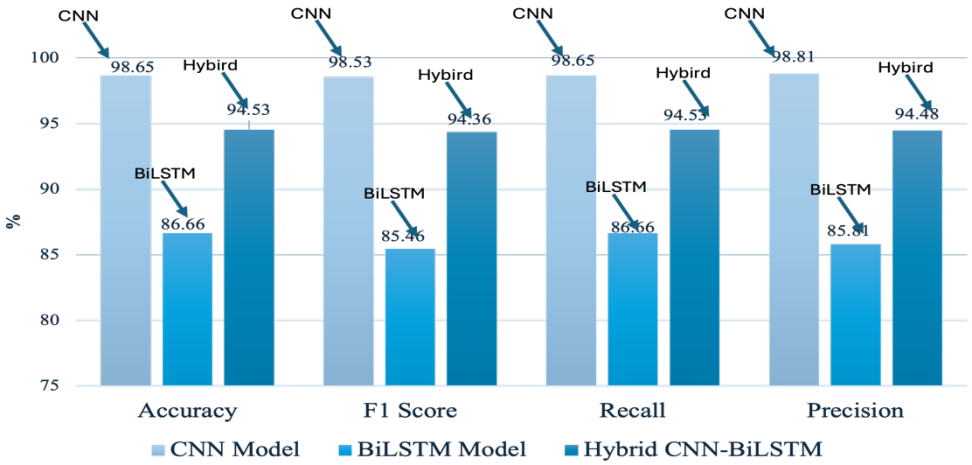


Fig. 7: A comparison among accuracy, F1Score, recall, and precision across all proposed models.

- To prevent overfitting, *L2 regularization* is applied to each dense layer. Additionally, *dropout layers* with a dropout rate of 20% are added after each dense layer, randomly dropping a fraction of the neurons during training to enhance generalization.

- The final layer of the CNN model *is a dense layer with 8 neurons*, corresponding to the number of classes in the CICIoT2023 dataset. A *softmax activation function* is used in this layer, providing a probability distribution over the 8 classes, allowing the model to perform multi-class classification.

C. The proposed BiLSTM model architecture

The proposed Recurrent Neural Network (RNN) model utilizes *Bidirectional Long Short-Term Memory (BiLSTM)* layers to effectively capture temporal dependencies in the sequence data. This design choice aims to leverage the bidirectional nature of the BiLSTM, enabling the model to learn from both past and future contexts simultaneously, which is particularly beneficial for tasks that involve sequence analysis, such as network traffic anomaly detection. Fig. 5 depicts our proposed BiLSTM model architecture. To begin with, the BiLSTM model is defined as a *Sequential model*, which organizes layers in a linear stack. The architecture begins with a *Bidirectional LSTM layer comprising 512 units,* with the *return_sequences=True* parameter enabled. This configuration ensures that the LSTM layer outputs a sequence of vectors, rather than a single hidden state, allowing the subsequent layer to process temporal dependencies more effectively. *The input shape* is specified as *(X_train.shape[1], 1)*, where *X_train.shape[1]* corresponds to the number of features in the dataset, and the channel dimension is set to 1 for one-dimensional data.

To mitigate the risk of overfitting, a *Dropout layer* is incorporated immediately after the first BiLSTM layer with a dropout rate of *50%*, randomly dropping half of the neurons during training. This helps in regularizing the model by preventing it from relying too heavily on specific neurons. The second layer is another *Bidirectional LSTM*, also configured with *512 units* but without threturn_sequences parameter. As a result, this layer outputs a single vector, summarizing the learned temporal features across the entire input sequence. This compressed representation is then passed to the final dense layer for classification. The final layer of the model is a *dense layer* with *8 neurons*, corresponding to the number of output classes in the dataset. A *softmax activation function* is applied in this layer to produce a probability distribution across the 8 classes, making it suitable for multi-class classification tasks.

D. The proposed hybrid CNN-BiLSTM model architecture

This proposed model architecture integrates Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) layers to effectively capture both spatial and temporal features in the dataset. This hybrid model leverages the strengths of CNN for feature extraction and BiLSTM for temporal dependency learning, making it well-suited for complex sequence classification tasks, such as network attack detection. Fig. 6 illustrates a detailed architecture for the proposed CNN-BiLSTM model. As shown in Fig. 6, the architecture starts with CNN layers for features extraction. The initial layers of the model consist *of three consecutive 1D convolutional layers*, each followed by *a max-pooling operation*.

The *CNN layers* are designed to automatically learn and extract hierarchical spatial features from the input data:

- *The first convolutional layer* uses *64 filters* with a kernel size of 3 and a *ReLU activation function*, which helps capture local patterns in the data.
- A *MaxPooling layer* with a pool size of 2 reduces the dimensionality, preserving the most important features while reducing computational complexity.
- This is followed by a *Conv1D layer with 128 filters* and another max-pooling layer, further refining the learned spatial features.
- The *final convolutional block* uses *256 filters*, enhancing the model's ability to detect complex patterns and anomalies.

By stacking multiple *convolutional layers*, the model extracts a rich set of spatial features, which are crucial for distinguishing different types of network attacks. The max-pooling layers help to downsample the data, mitigating the risk of overfitting and speeding up training. After extracting spatial features, the model passes the output through a series of *Bidirectional LSTM (BiLSTM) layers*. As shown in Fig. 6,

the *BiLSTM layers* are designed to capture temporal dependencies by processing the sequence data in both forward and backward directions:

- The first *BiLSTM layer* consists of *128 units* and utilizes *return_sequences=True*, enabling the layer to output a sequence rather than a single hidden state. This helps in learning long-range dependencies within the data.
- A *Dropout layer* with a rate *of 0.3 follows*, serving as a regularization technique to prevent overfitting by randomly dropping a fraction of the neurons during training.
- The second *BiLSTM layer* has *64 units*, further refining the learned temporal patterns, followed by another dropout layer with the same rate.

The inclusion of *BiLSTM layers* allows the model to effectively capture bidirectional temporal patterns, which are essential for understanding the sequential nature of network traffic and identifying potential attack sequences. The *output* from *the BiLSTM layers* is fed into a *dense layer* with *64 units* and a *ReLU activation function*. This layer acts as a bridge, transforming the learned features into a more compact representation suitable for classification. A dropout layer (rate of 0.3) is added to further mitigate overfitting risks. Finally, the model ends with *a dense layer* using a *softmax activation function*, which outputs probabilities for each of the attack classes.

E. Training and testing of DL models stage

To facilitate the training and evaluation of deep learning models, we implemented a systematic approach to partition the CICIoT2023 dataset. Following best practices in machine learning and deep learning, the dataset was divided into three subsets: training (80%), validation (10%), and testing (10%). The training subset, comprising the majority of the dataset, was used to enable the deep learning algorithms to learn robust feature representations by capturing underlying patterns and relationships within the data (see Table IV). The validation subset was employed during the training phase to monitor performance, assess generalization capabilities, and guide decisions on hyperparameter tuning and early stopping. Finally, the testing subset, an independent portion of the data, provided an unbiased evaluation of the model's performance on unseen data, ensuring its reliability for practical deployment. This partitioning approach established a robust framework for evaluating model performance and developing accurate, generalizable predictive systems. To train the models, we used the Adam optimizer with a learning rate of 0.0001, chosen for its adaptability and robustness in optimizing deep neural networks. The loss function, sparse categorical crossentropy, was selected as it is well-suited for multi-class classification tasks. Additionally, several regularization techniques were implemented to enhance model performance and prevent overfitting:

- **EarlyStopping**: This technique monitored validation loss and halted training if no improvement was observed for five consecutive epochs, reducing training time and mitigating overfitting.
- **ReduceLROnPlateau**: This method dynamically adjusted the learning rate, reducing it by a factor of 0.5 when the validation loss plateaued, enabling finer optimization during later epochs.

By combining a well-structured dataset partitioning strategy with effective optimization and regularization techniques, we ensured a rigorous and reliable evaluation of our deep learning models' capabilities.

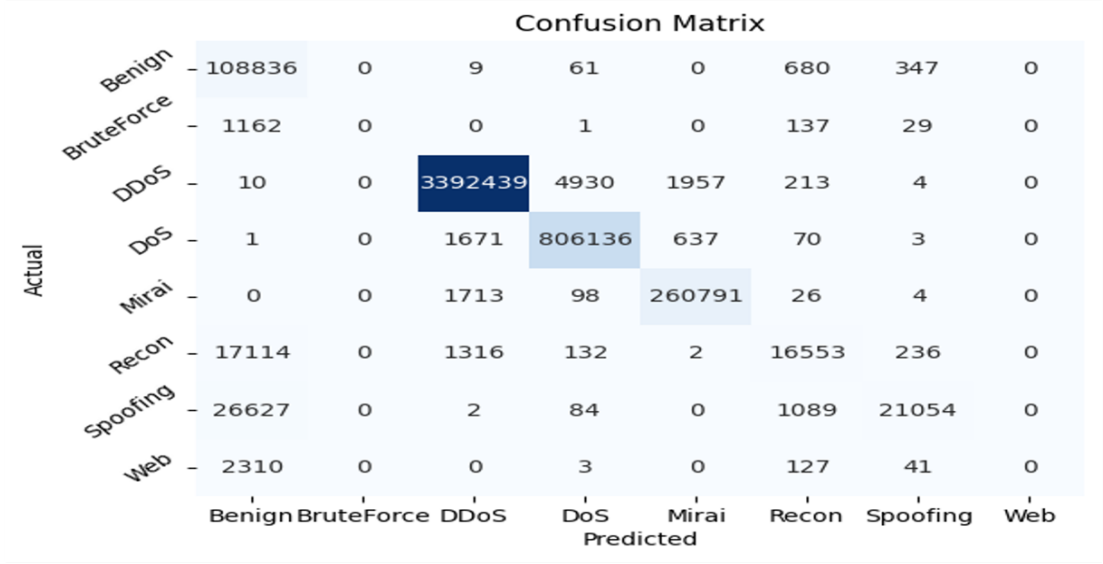
VI. RESULTS AND DISCUSSION

In this section, we present the performance of three distinct deep learning models—CNN, BiLSTM, and a hybrid CNN-BiLSTM—on a multi-class classification task aimed at detecting various types of cyber-attacks in IoT environment. The goal was to determine which model architecture provides the best generalization capability for accurately identifying attack types in a CICIoT2023, which is a complex dataset, considering both spatial and temporal features of the input data.

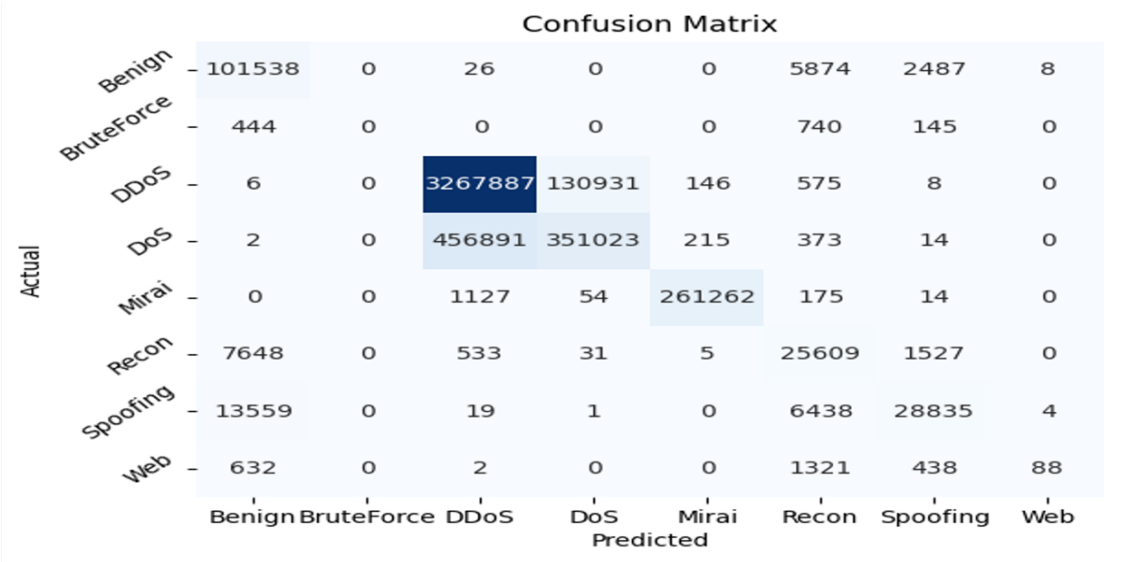
A. Comparative evaluation of model performance metrics

In Fig. 7, the bar chart presents a comparison of key evaluation metrics—*accuracy, F1 score, recall, and precision*—for the three models: CNN, BiLSTM, and CNN-BiLSTM. These metrics offer a holistic view of the models' effectiveness, taking into account not only their overall accuracy but also their ability to balance true positives, false positives, and false negatives across multiple classes. We are going to discuss them in more detail.

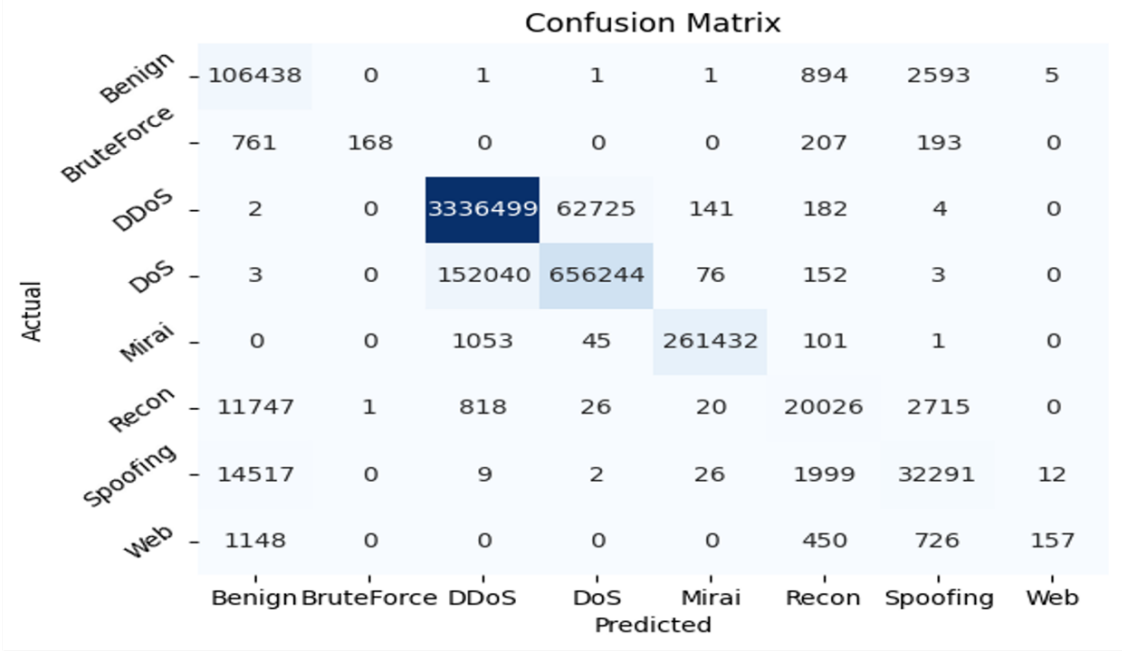
- *Accuracy:* The CNN model achieves the highest accuracy among both the BiLSTM and CNN-BiLSTM models. With an accuracy close to *98%*, the CNN model records superior performance, utilizing its convolutional layers to effectively capture spatial features in the dataset as shown in Fig. 7. The CNN-BiLSTM model follows with an accuracy of approximately *94%*, indicating that the hybrid architecture benefits from incorporating temporal features through the BiLSTM layers. In contrast, the BiLSTM model has the lowest accuracy, around *85%*, highlighting its limitations in handling complex spatial features without convolutional processing.
- *F1 Score:* The CNN model again outperforms with the highest F1 score, reflecting its strong ability to correctly classify both positive and negative instances. The CNN-BiLSTM model shows a slightly lower F1 score but still performs well, suggesting a satisfactory trade-off between precision and recall. The BiLSTM model has the lowest F1 score, which aligns with its reduced accuracy and indicates potential challenges in achieving a balanced classification performance.
- *Recall:* The CNN model exhibits the highest recall, indicating its effectiveness in minimizing false negatives, as shown in Fig. 7. The CNN-BiLSTM model also shows high recall, benefiting from the temporal feature extraction of the BiLSTM layers. However, the BiLSTM model falls behind with significantly lower recall, suggesting that it struggles to identify certain attack types accurately, potentially due



(a) Confusion matrix for the proposed CNN model



(b) Confusion matrix for the proposed BiLSTM model



(c) Confusion matrix for the proposed hybrid CNN-BiLSTM model

Fig.8 : Confusion matrix for the proposed CNN, BiLSTM, and hybrid CNN-BiLSTM models

to its inability to capture spatial dependencies effectively.

- *Precision:* the CNN model consistently maintains the highest precision, showcasing its ability to prevent false positives. The CNN-BiLSTM model shows competitive precision, slightly lower than the CNN model, which may be attributed to the added complexity of the temporal layers introducing some noise. The BiLSTM model has the lowest precision, indicating a higher rate of false positive predictions, likely due to its weaker feature extraction capabilities.

B. Confusion matrix analysis

Fig. 8 shows the confusion matrices for the CNN, hybrid CNN-BiLSTM, and BiLSTM models. The confusion matrices provide a detailed view of the models' classification performance across various network attack types. These matrices highlight how well each model distinguishes between benign traffic and different attack classes, offering insights into the strengths and limitations of each architecture.

1) CNN model:

The confusion matrix for the CNN model in Fig. 8(a) shows strong performance with minimal misclassifications across most classes. The significant diagonal values in the matrix indicate excellent precision in classifying high-volume attack types like DDoS and DoS. The model correctly identifies most instances of benign traffic, with only minor confusion observed for less frequent attack types like *Recon and Spoofing*. Overall, the CNN model demonstrates robust generalization, effectively separating most classes with few false positives, supporting its high overall accuracy of *98%.*

2) Hybrid CNN-BiLSTM model:

The hybrid CNN-BiLSTM model's confusion matrix in Fig. 8(c) shows a few more wrong classifications than the CNN model's, mainly when it comes to telling the difference between types of attacks like *Recon and Spoofing*. While the model performs well on high-volume classes such as *DDoS and DoS*, there are more errors for classes like Mirai and Web attacks. The added complexity of the BiLSTM layers, which may introduce noise instead of meaningful temporal features for this dataset, could be the cause of the increased misclassifications. Despite these challenges, the CNN-BiLSTM model still maintains a strong performance overall, particularly for classes with distinct patterns, achieving an accuracy of *94%.*

3) BiLSTM model:

The BiLSTM model in Fig. 8(b) reveals the highest error rates among the three architectures, as seen in its confusion matrix. Significant misclassifications occur, particularly for classes such as DoS and DDoS, leading to the incorrect classification of many samples as other attack types. This suggests that the BiLSTM model struggles to capture the discriminative spatial features necessary for accurate classification. The model's reliance on temporal dependencies without spatial feature extraction appears to limit its effectiveness, leading to lower overall accuracy. The confusion between *Recon* and *Spoofing* attacks is also notable, indicating difficulties in distinguishing between similar attack patterns.

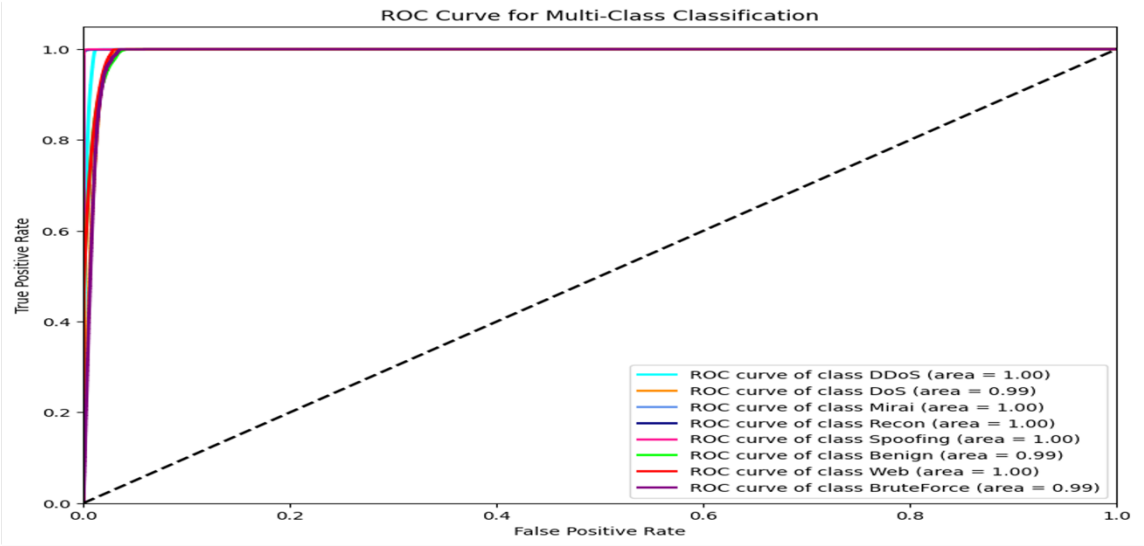
C. ROC curve analysis for multi-class classification

Fig. 9 presents the Receiver Operating Characteristic (ROC) curves, which provide a comprehensive view of the classification performance for each model across multiple classes. By plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) for each class, the ROC curves offer insights into the discriminative ability of the models. The Area Under the Curve (AUC) is used as a metric to quantify this performance, with values closer to 1 indicating better classification.

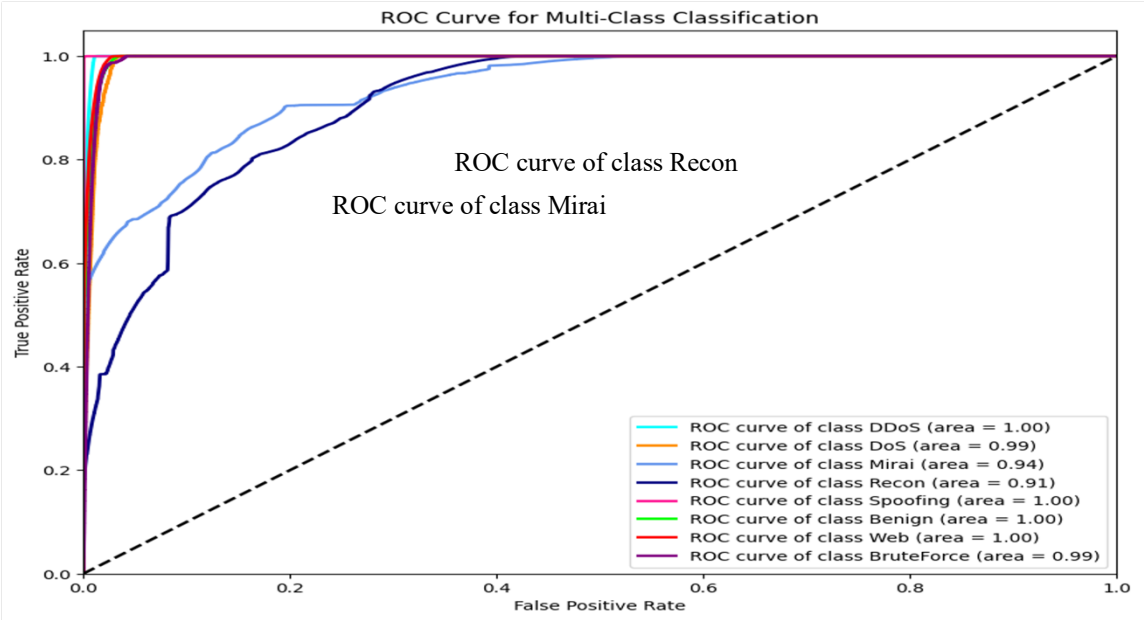
From Fig. 9.(a), it is noticeable that the ROC for the CNN Model demonstrates near-perfect performance, with AUC values close to 1.0 for almost all classes. The classes *DDoS, Benign, Web*, and *Spoofing* exhibit an *AUC* of *1.0*, indicating positive rates. The *DoS, BruteForce, and Mirai* classes also show high *AUC* values, around *0.99*, suggesting strong classification capability. Exceptional discriminative power and near-zero false positives suggest that the slight deviation in classes such as Recon (AUC = 0.98) is due to a minor challenge in differentiating this class from others, potentially due to feature overlap. Overall, the CNN model achieves excellent class separability, underscoring its robustness and effectiveness in handling diverse attack types.

Moreover, the ROC of the hybrid CNN-BiLSTM model, which is shown in Fig. 9(c), also shows strong ROC performance, with AUC values predominantly above 0.99 for most classes. High-volume classes such as DDoS, DoS, and Web attacks achieve AUC values of 1.0, reflecting the model's ability to accurately classify these well-defined patterns. However, the AUC for the Recon class is slightly lower (0.91), suggesting reduced separability compared to the CNN model. The BiLSTM layers added temporal complexity may not significantly contribute to distinguishing between certain attack types in this dataset. Despite these minor variations, the CNN-BiLSTM model maintains a high overall classification performance, benefiting from both spatial and temporal feature extraction.

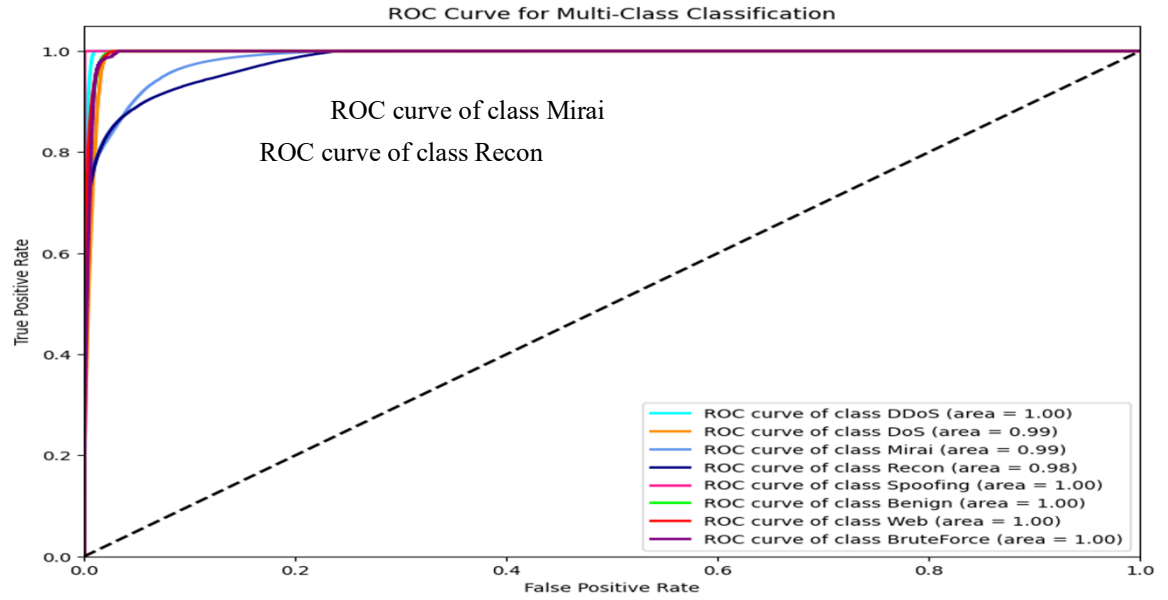
In contrast, Fig. 9(b) depicts the BiLSTM model's *ROC curve* analysis, revealing the highest variability among the three models, with several classes displaying lower *AUC* values. While the model achieves perfect AUC scores for DDoS, Web, and Spoofing, it struggles with classes like Recon and Mirai, where the *AUC* drops to *0.94* and *0.91*, respectively. This decline suggests that the BiLSTM model's reliance on temporal features alone may not be sufficient to differentiate certain classes effectively, particularly when spatial features are more informative. Increased false positives for classes with overlapping temporal patterns hinder the model's overall discriminative ability, despite its strong performance on high-volume classes.



(a) ROC curve for the proposed CNN model



(b) ROC curve for the proposed BiLSTM model



(c) ROC curve for the proposed hybrid CNN-BiLSTM model

Fig.9 : ROC curves for the proposed CNN, BiLSTM, and hybrid CNN-BiLSTM models

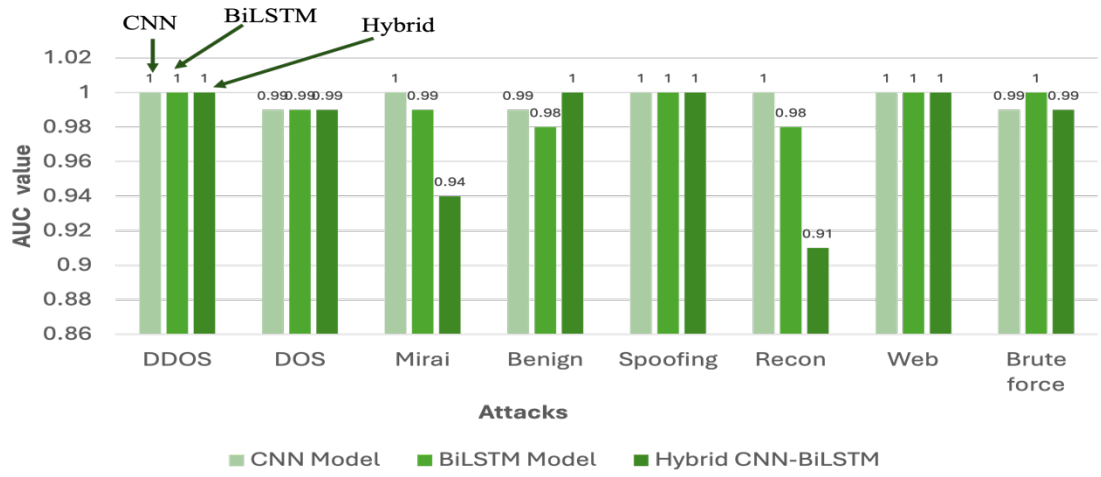


Fig.10: AUC values of 7 types of attacks detected by the three proposed deep learning models

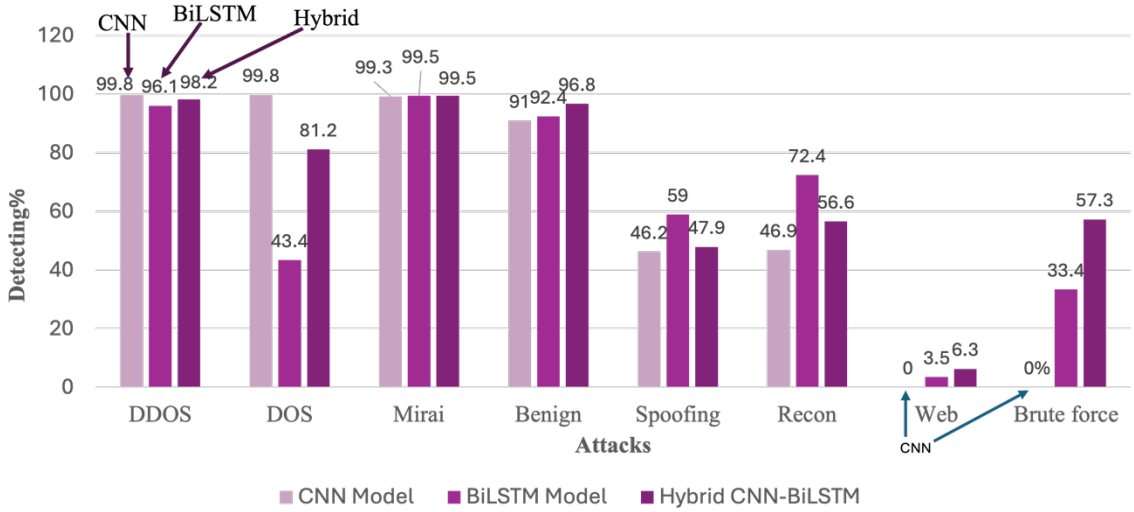


Fig. 11: A comparison of detection accuracy percentages among the three proposed models for various types of cyber attacks

Fig. 10 displays the *AUC* values for various attack types in three different models: the CNN, BiLSTM, and hybrid CNN-BiLSTM. The hybrid model consistently achieves the highest AUC values across most attack types, indicating superior performance. The BiLSTM model shows slightly lower performance in some cases, particularly for benign and reconnaissance attacks. The CNN model generally maintains high AUC values but falls short compared to the hybrid model in specific categories. Overall, the hybrid model demonstrates robust performance in distinguishing attack types, making it the most effective among the three models. It is noticeable that Fig. 11 compares the detection accuracy of three models, namely CNN, BiLSTM, and Hybrid CNN-BiLSTM, across various attack types. The CNN model demonstrates high accuracy for DDOS, DOS, Mirai, and Benign attacks, consistently achieving over 90%. The hybrid CNN-BiLSTM model outperforms BiLSTM in most categories, particularly for DOS, Benign, and Brute Force attacks. BiLSTM shows lower performance in certain cases, such as *DOS and Brute Force. Recon, Spoofing, and Web attacks* exhibit relatively lower detection rates across all models, indicating potential challenges in identifying these attack types effectively. Overall, the hybrid CNN-BiLSTM provides a balanced performance, making it a robust choice.

The results of this study underscore the importance of incorporating spatial feature extraction through convolutional layers in network traffic classification tasks. The CNN model's robust performance highlights its ability to learn discriminative spatial patterns, making it the most effective architecture for this dataset. The hybrid CNN-BiLSTM model is well-balanced because it uses both spatial and temporal features. However, it may need more work, like more regularization or advanced feature engineering, to get rid of noise and improve classification accuracy. The BiLSTM model, on the other hand, demonstrated the limitations of relying solely on temporal dependencies, suggesting the need to integrate convolutional layers to capture local patterns effectively.

Future work could explore the use of various models that combine the predictions of CNN and CNN-BiLSTM architectures, potentially leveraging their complementary strengths. Additionally, advanced data augmentation techniques and hyperparameter tuning could further improve generalization and reduce overfitting. This analysis provides

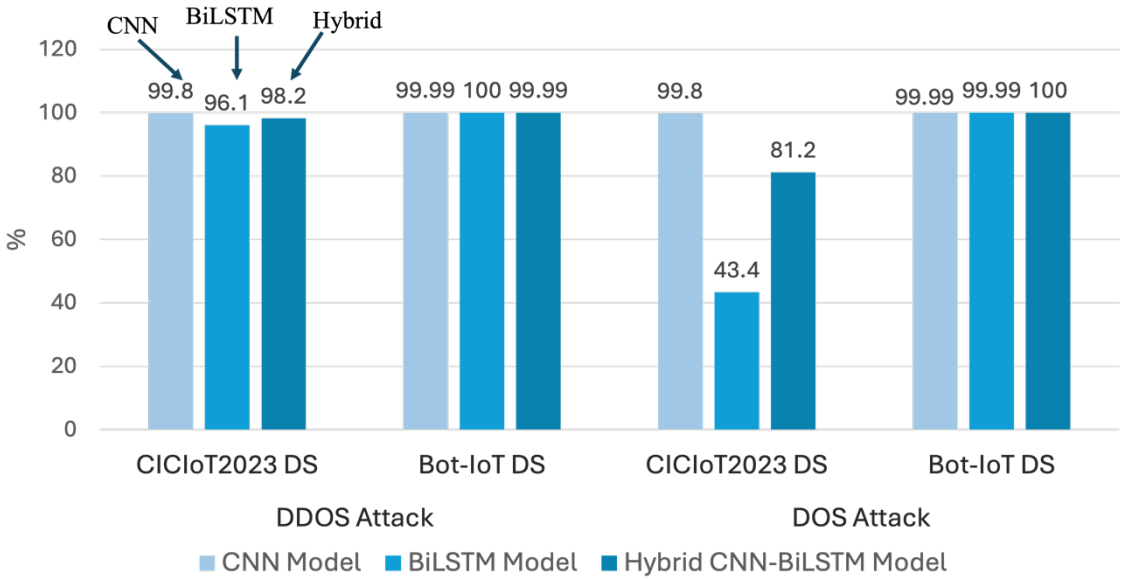


Fig. 12: Performance comparison of the three proposed deep learning models in detecting DDoS and DoS attacks using CICIoT2023 and BoT-IoT datasets

valuable insights into the design and selection of deep learning architectures for network intrusion detection, guiding future research in developing more effective and robust models for cybersecurity applications.

D. Evaluation of the Proposed Models for DDoS and DoS Attack Detection between CICIoT2023 and BoT-IoT Datasets

Fig. 12 illustrates the detection performance of CNN, BiLSTM, and Hybrid CNN-BiLSTM to DDoS and DoS attacks that have been tested for two data sets, that is, the CICIoT2023 and Bot-IoT datasets. Accordingly, the result yielded by applying the CNN-only model outscored all previous models with accuracy as high as 99.8%, though it is very closely followed with 98.2% by the hybridized model CNN-BiLSTM. The CNN model showed high performance for the CICIoT2023 dataset in identifying DDoS attacks, while the BiLSTM model gave relatively low performance with an accuracy of 96.1%. This analysis has been able to achieve complete detection accuracy for the Bot-IoT dataset, while CNN and hybrid CNN-BiLSTM architectures reported almost perfect detection performances with an accuracy of 99.99%. The minor variations among these methods show that each of them performed highly efficiently in the context of the target dataset of DDoS-attack detection and that the model BiLSTM is slightly at an advantage over the others.

What can be concluded that during different datasets and kinds of attacks, the CNN model, in general, has shown great performance, but it was far better for the CICIoT2023 dataset when the DoS attack kind was considered. Another competitive performance is from the hybrid CNN-BiLSTM, which shows one of the highest performances for the Bot-IoT dataset in the detection of DoS attacks. On the other hand, BiLSTM performed very inconsistently: this model completely failed in the case of the DoS attack for the CICIoT2023 dataset but outperformed all other models in the detection of DDoS attacks for the Bot-IoT dataset. Therefore, this analysis justifies the use of model selection based on the characteristics of datasets and types of attacks in IoT security applications. More crucially, an in-depth look is necessary at the reason behind such variations in the performance exhibited by a specific model like BiLSTM.

IX. CONCLUSION

By using the advanced CICIoT2023 dataset, this study did a full comparison of CNN and RNN architectures for finding strange things in IoT networks. The findings highlight the superior performance of CNN and hybrid CNN-BiLSTM models in detecting diverse attack scenarios compared to standalone BiLSTM models. The CICIoT2023 dataset was very helpful in finding real-life IoT threats and providing a strong standard for checking the performance of deep learning-based intrusion detection systems (IDSs). Our methodological rigor, including precise model architecture design, hyperparameter optimization, and thorough performance evaluation, underscores the viability of deep learning in enhancing IoT network security.

As a step forward, efforts will focus on refining the CNN-BiLSTM model by incorporating advanced feature selection methods and shifting toward a more granular classification of attacks based on specific types. In addition, we are going to optimize the hybrid CNN-BiLSTM using the Kepler Optimization Algorithm (KOA), as suggested in [34]. This refinement approach can enhance detection accuracy and scalability. Adding federated learning, a decentralized framework [33], will also make it possible to train across distributed IoT data sources while reducing the computational challenges that come with big datasets like CICIoT2023. Federated learning is a revolutionary way to quickly and safely find anomalies because it allows for parallel processing and lowers the risks of centralization. Ultimately, the insights from this research contribute to the development of robust and scalable IDS solutions for IoT networks, paving the way for future innovations in securing the rapidly expanding IoT ecosystem.

REFERENCES

[1] H. Atlam and G. Wills, "IoT security, privacy, safety and ethics," in *Internet of Things*. Springer International Publishing, pp. 123–149.

[2] S. Abbas, A. Hejaili, G. Sampedro, M. Abisado, A. Almadhor, T. Shahzad, and K. Ouahada, "A novel federated edge learning approach for detecting cyberattacks in IoT infrastructures," *IEEE Access: Practical Innovations, Open Solutions*, vol. 11, pp. 112189–112198.

[3] H. HaddadPajouh, A. Dehghantanha, M. Parizi, A. R. M., and H. Karimipour, "A survey on internet of things security: requirements, challenges, and solutions," *Internet of Things*, vol. 14, no. 100129, p. 100129.

[4] T. Car, L. P. Stifanich, and M. Šimunić, "Internet of Things (IoT) in tourism and hospitality: Opportunities and challenges," in *Tourism in Southern and Eastern Europe*.

[5] A. Srivastava, S. Gupta, M. Quamara, P. Chaudhary, and V. Aski, "Future IoT-enabled threats and vulnerabilities: State of the art, challenges, and future prospects," *International Journal of Communication Systems*, vol. 33, no. 12.

[6] R. Khallaf and M. Khallaf, "Classification and analysis of deep learning applications in construction: A systematic literature review," *Automation in Construction*, vol. 129, no. 103760, p. 103760.

[7] A. Vargas, A. Mosavi, and R. Ruiz, "Deep learning: A review," in *Preprints*.

[8] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access: Practical Innovations, Open Solutions*, vol. 7, pp. 53040–53065.

[9] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Applied Sciences*, vol. 9, p. 4396.

[10] E. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. Ghorbani, "CICIOT2023: A real-time dataset and benchmark for large-scale attacks in IoT environments," *Sensors*, vol. 23, p. 5941.

[11] S. Mohammed, "A machine learning-based intrusion detection of DDoS attack on IoT devices," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 10, pp. 2278–3091.

[12] A. Guezzaz, S. Benkirane, M. Azrour, and S. Khurram, "A reliable network intrusion detection approach using decision tree with enhanced data quality," *Security and Communication Networks*, pp. 1–8.

[13] H. Tyagi and R. Kumar, "Attack and anomaly detection in IoT networks using supervised machine learning approaches," *Revue d'Intelligence Artificielle*, vol. 35, no. 1, pp. 11–21.

[14] M. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cybersecurity intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, no. 102419, p. 102419.

[15] M. Bhavsar, K. Roy, J. Kelly, and O. Olusola, "Anomaly-based intrusion detection system for IoT application," *Discover Internet of Things*, vol. 3, no. 1.

[16] I. Ullah and Q. Mahmoud, "Design and development of a deep learning-based model for anomaly detection in IoT networks," *IEEE Access: Practical Innovations, Open Solutions*, vol. 9, pp. 103906–103926.

[17] M. S. Q. Ahmed, A. Fadhel, and M., "Internet of things (IoT): A technology review, security issues, threats, and open challenges," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 3, pp. 1685–1693.

[18] Kilichev, "Next-generation intrusion detection for IoT devices: Integrating CNN, LSTM, and GRU models," *Mathematics*, vol. 12, no. 4.

[19] A. Meliboev, J. Alikhanov, and W. Kim, "Performance evaluation of deep learning-based network intrusion detection system across multiple balanced and imbalanced datasets," *Electronics*, vol. 11, no. 4, p. 515.

[20] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access: Practical Innovations, Open Solutions*, vol. 7, pp. 41525–41550.

[21] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Anomaly-based network intrusion detection for IoT attacks using deep learning technique," *Computers & Electrical Engineering: An International Journal*, vol. 107, p. 108626.

[22] A. Awajan, M. Bhavsar, K. Roy, J. Kelly, and O. Olusola, "A novel deep learning-based intrusion detection system for IoT networks," *Computers*, vol. 12, no. 2, p. 34.

[23] M. Vishwakarma and N. Kesswani, "DIDs: A deep neural network-based real-time intrusion detection system for IoT," *Decision Analytics Journal*, vol. 5, p. 100142.

[24] R. Alghamdi and M. Bellaiche, "An ensemble deep learning-based IDS for IoT using lambda architecture," *Cybersecurity*, vol. 6, no. 1.

[25] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," Expert Systems with Applications, vol. 185, no. 115524, p. 115524.

[26] L. Ma, Y. Chai, L. Cui, D. Ma, Y. Fu, and A. Xiao, "A deep learning-based DDoS detection framework for Internet of Things," in *ICC2020 - 2020 IEEE International Conference on Communications (ICC)*.

[27] K. A. Alimi, K. Ouahada, A. Abu-Mahfouz, S. Rimer, and O. Alimi, "Refined LSTM-based intrusion detection for denial-of-service attacks in Internet of Things," *Journal of Sensor and Actuator Networks*, vol. 11, no. 3, p. 32.

[28] 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.

[29] I. Ullah and Q. Mahmoud, "Design and development of RNN anomaly detection model for IoT networks," *IEEE Access: Practical Innovations, Open Solutions*, vol. 10, pp. 62722–62750.

[30] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. Bahaj, "Anomaly-based intrusion detection system for IoT networks through deep learning model," *Computers & Electrical Engineering: An International Journal*, vol. 99, p. 107810.

[31] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: BoT-IoT dataset," *Future Generations Computer Systems*, vol. 100, pp. 779–796, 2019.

[32] A. T. Aldaej and I. Ullah, "Deep learning-inspired IoT-IDS mechanism for edge computing environments," *Sensors*, Basel, Switzerland, p. 9869.

[33] C. Hu, J. Jiang, and Z. Wang, "Decentralized federated learning: A segmented gossip approach," *arXiv*, 2019. [Online]. Available: http://arxiv.org/abs/1908.07782

[34] Y. Huang, J. Li, Y. Li, R. Lin, J. Wu, L. Wang, and R. Chen, "An Improved Hybrid CNN-LSTM-Attention Model with Kepler Optimization Algorithm for Wind Speed Prediction," *Engineering Letters*, vol. 32, no. 10, pp. 1957-1965, 2024.

[35] S. Thangam and S. S. Chakkaravarthy, "An Edge-enabled Virtual Honeypot Based Intrusion Detection System for Vehicle-to-Everything (V2X) Security using Machine Learning," *IAENG International Journal of Computer Science*, vol. 51, no. 9, pp. 1374-1384, 2024.

[36] A. Ajiboye, M. Olumoye, D. Aleburu, A. Olayiwola, D. Olayiwola, and S.Ajose, "Dimensionality Reduction for Deep Learning Based Intrusion Detection Systems for IoT," Lecture Notes in Engineering and Computer Science:Proceedings of The International MultiConference of Engineers and Computer Scientists 2023, 5-7 July, 2023,Hong Kong, pp76-81.

Amal M. Al-Ghamdi She holds a bachelor's degree in computer science from King Abdulaziz University, with a dual specialization in Programming and Artificial Intelligence. She further advanced her academic journey by earning a postgraduate's degree in Cybersecurity from Saudi Electronic University in collaboration with Colorado State University (Global Campus). Her research interests are deeply rooted in the fields of Artificial Intelligence and Cybersecurity. She is committed to leveraging her expertise to drive innovation and contribute meaningfully to the advancement of society.

Marwah M. Alansari She is an assistant professor at Saudi Electronic University (SEU) in Riyadh. Previously, she served as an assistant professor at Albaha University. She holds a PhD in Computer Science from the University of Birmingham, UK. Her research interests include managing cloud computing resources, building trust in cloud service-based models, and applying artificial intelligence algorithms for automatic generation processes and software architecture. Additionally, she has interests in applying AI in software engineering development.