electronics

MDPI

Article

IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm

Sami Yaras and Murat Dener *

Department of Information Security Engineering, Graduate School of Natural and Applied Sciences, Gazi University, 06560 Ankara, Turkey
* Correspondence: muratdener@gazi.edu.tr

Abstract: The most significant threat that networks established in IoT may encounter is cyber attacks. The most commonly encountered attacks among these threats are DDoS attacks. After attacks, the communication traffic of the network can be disrupted, and the energy of sensor nodes can quickly deplete. Therefore, the detection of occurring attacks is of great importance. Considering numerous sensor nodes in the established network, analyzing the network traffic data through traditional methods can become impossible. Analyzing this network traffic in a big data environment is necessary. This study aims to analyze the obtained network traffic dataset in a big data environment and detect attacks in the network using a deep learning algorithm. This study is conducted using PySpark with Apache Spark in the Google Colaboratory (Colab) environment. Keras and Scikit-Learn libraries are utilized in the study. 'CICIoT2023' and 'TON_IoT' datasets are used for training and testing the model. The features in the datasets are reduced using the correlation method, ensuring the inclusion of significant features in the tests. A hybrid deep learning algorithm is designed using one-dimensional CNN and LSTM. The developed method was compared with ten machine learning and deep learning algorithms. The model's performance was evaluated using accuracy, precision, recall, and F1 parameters. Following the study, an accuracy rate of 99.995% for binary classification and 99.96% for multiclassification is achieved in the 'CICIoT2023' dataset. In the 'TON_IoT' dataset, a binary classification success rate of 98.75% is reached.

Keywords: DDoS attacks; big data; intrusion detection system; hybrid algorithm; deep learning; machine learning; multi-class classification; IoT security

Citation: Yaras, S.; Dener, M. IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm. *Electronics* **2024**, *13*, 1053. https://doi.org/10.3390/electronics13061053

Academic Editor: Andrei Kelarev

Received: 24 November 2023
Revised: 4 January 2024
Accepted: 5 January 2024
Published: 12 March 2024

1. Introduction

Wireless sensor networks serve as a bridge between the real world and the digital world. The network created by connecting sensors to each other to detect the real world and spread these data to the digital world is generally called a wireless sensor network. Wireless Sensor Networks (WSN) provide real-time data flow in various fields, such as military surveillance, battlefield monitoring, forest fire tracking, building security monitoring, and healthcare services. A wireless data network is part of the Internet of Things (IoT), and the collected data are processed, analyzed, and presented to the user with the help of a base station. A WSN typically includes at least one gateway node that serves as a link between the network and the external world [1].

Communication between these sensors and the gateway node should be conducted in the most energy-efficient way because sensor nodes have limited energy and their batteries cannot be recharged. Due to the characteristic features of wireless sensor networks, the communication methods should be simple, efficient, and easily adaptable to different scenarios. As a result, the resources they can use are limited. Due to factors such as low power consumption, processor constraints, and the inability to add some devices due to cost, wireless sensor networks are vulnerable to attacks. Security and privacy are crucial considerations in these systems. Enhancing them and adding new features will require

overcoming obstacles, such as energy constraints and processor limitations. Applying traditional security measures, such as cryptography, to such networks is challenging because WSNs are highly vulnerable to attacks due to their open and distributed structures and the limited resources of sensor nodes. Additionally, frequent broadcasting of packets in WSNs may be necessary, and sensor nodes can be randomly deployed in an environment, making it easy for a malicious attack to be injected into the WSN [2].

An aggressive sensor can compromise the network, eavesdrop on messages, inject spoofed messages, alter the integrity of data, and waste network resources. Denial of Service (DoS) attacks are considered one of the most common and dangerous threats to WSN security. DoS attacks remain a significant challenge today. There are various forms of this attack, and its main purpose is to interrupt or suspend the services provided by WSNs [2]. The destructive impact of DoS attacks is that they consume the power resources of nodes and significantly shorten their operational lifetimes. Therefore, sensors can die quickly because of DoS attacks. Nodes that run out of power become useless; hence, the intended use of the WSN is compromised.

As modern technology is based on data, DoS attacks are a very dangerous and significant attack type. Data is known as raw information and becomes meaningful after processing. With the beginning of the computer age, the amount of data used has significantly increased. Logs left by network traffic, system events, and system components can be included in big data. Thanks to big data analytics and related technologies, data streams can be continuously monitored, and anomalies and changes in the network can be detected to ensure network security. By working together, big data and artificial intelligence algorithms can analyze the past and present data of the network, determining whether the current network traffic is normal or attacked [3].

In networks with limited resources that are vulnerable to attacks, there are systems capable of detecting these attacks in real time and alerting the relevant sensor node. These systems are called Intrusion Detection Systems (IDS). An IDS is a proactive attack detection tool used to detect and classify unauthorized entries, attacks, or violations of security policies in a timely manner [4]. Due to the limited resources of sensor nodes, these intrusion detection systems should have high accuracy and should not impose an additional burden on the network's resource consumption.

In this study, a new intrusion detection system in a big data environment is developed with a hybrid deep learning algorithm. The algorithm is implemented in Pyspark, Apache Spark's Python support, using the Google Colabs environment. Apache Spark is preferred because of its fast execution and the data used in the algorithm are part of big data. This intrusion detection algorithm is trained and tested using CICIoT2023 and TON_IOT datasets. The system is evaluated for both binary and multiclass classification, using evaluation parameters such as accuracy, precision, recall and F1-score. The developed model was compared with ten traditional machine and deep learning algorithms (Random Forest (RF), Decision Tree (DT), Gradient Boost (GB), AdaBoost (ADA), Naive Bayes (NB), Logistic Regression (LR), K-Nearest Neighbor (KNN), Convolutional Neural Network (CNN), Multi-layer Perceptron (MLP) and Long Short-Term Memory (LSTM)).

The contributions of the proposed study to the literature are as follows:

- A new IDS has been developed efficiently in a big data environment using a new hybrid deep learning algorithm.
- The developed algorithm has been tested for both binary and multiclass classification and achieved high accuracy in both cases.
- The developed hybrid algorithm has been compared with ten mostly used machine and deep learning algorithms. The results showed that the proposed hybrid method has better accuracy than traditional methods.
- Deep learning algorithms, such as CNN and LSTM, were individually tested. It was observed that the hybrid algorithm created using CNN and LSTM performs better than using them separately.

- High accuracy has been achieved in a large dataset such as CICIoT2023, which exhibits an imbalanced distribution of values without the use of any balancing methods.
- The addition of a second dataset to the study resulted in a high intrusion detection rate in a different dataset.

The study consists of seven sections. Section 2 describes the studies conducted with the CICIoT2023 and TON_IOT datasets. Section 3 defines the Distributed Denial of Service (DDoS) attacks present in the dataset and provides a general description of anomaly detection systems developed to prevent them. Section 4 describes the preprocessing stages of the used dataset and introduces the deep learning algorithms employed in the study. Section 5 explains the developed algorithm. The evaluation results of the model and their comparison are given in Section 6. Section 7 includes a discussion of the study. Section 8 consists of the conclusion and future work section.

2. Related Works

Anomaly-based intrusion detection systems define an attack as any deviation from normal behavior. Due to the inherent structure of Wireless Sensor Networks (WSNs), there are certain limitations. Therefore, the IDSs to be used in WSNs must be designed with these limitations in mind. In the literature, there are IDS systems developed for WSNs using classification, clustering, machine learning, and statistical learning algorithms [1]. This section provides a summary of the IDS studies developed for detecting DDoS in WSNs.

Cil et al. [4] developed a DDoS detection system using a classical deep learning algorithm. This algorithm is designed with 69 units in the input layer and three hidden layers with 50 units each between the input and output layers. Two different datasets were obtained using the CICDDoS2019 dataset. The first one includes data labeled as attack or normal for traffic, while the second one contains data with attack types. An accuracy rate of 99.97% in attack detection and 94.57% in attack type detection was achieved.

Almaraz-Rivera et al. [5] used the Bot-IoT dataset released in 2019, which addresses the class imbalance problem, to create a new ıntrusion detection system based on machine learning and deep learning models. Using this dataset, three different subsets were formed by selecting different feature sets. The first subset differs from the other two in timestamps (stime, ltime) and Argus sequence number (seq). In the second subset, timestamps were removed under the assumption that the model would memorize these features. Similarly, the sequence number was also removed. In the last subset, stime and ltime features were removed to evaluate the effect of timestamps. These created datasets were tested separately for binary and multiclass classification by RF, DT, LSTM, MLP, Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), and Support Vector Machine (SVM) machine learning algorithms. The evaluation resulted in an average accuracy of over 99%. Decision tree and MLP models were observed to be the best-performing methods.

Jia and their team [6] designed the FlowGuard algorithm, which consists of two components: flow filter and flow handler. The flow filter is responsible for filtering malicious flows based on filtering rules created by the flow handler and detecting malicious flows that cannot be identified based on traffic changes. The flow handler takes on the responsibility of identifying and classifying malicious flows according to two machine learning models developed, LSTM and CNN. The performance of the FlowGuard algorithm was evaluated using the CICDDoS2019 dataset, achieving a detection accuracy of 99.9% for attack detection and 98.9% for attack type detection.

Alghazzawi et al. [7] developed a hybrid deep learning algorithm using a feature selection approach. The model, created using CNN and BiLSTM deep learning algorithms, was tested in 10 different combinations of variables, such as filter count, filter size, and BiLSTM units. The algorithm achieved an accuracy rate of up to 94.52% using the CICDDoS2019 dataset during training, testing, and validation.

Ferrag et al. [8] proposed a deep learning-based DDoS attack detection system based on three different models: CNN, RNN, and Deep Neural Network (DNN). Each model's performance was trained and tested on CICDDoS2019 and TON_IoT datasets for binary and multi-

class classification. The datasets were divided into three different datasets (Dataset_2_class, Dataset_7_class, and Dataset_13_class) to analyze the efficiency of binary classification and multiclass classification. The CNN algorithm achieved accuracy rates of 99.95%, 95.90%, and 95.12% for binary, seven-class, and thirteen-class classifications, respectively.

Mamoudan et al. [9] designed an algorithm to predict buy–sell signals using technical analysis indicators, a popular tool in financial markets. To obtain the data, Moving Average Convergence Divergence (MACD), Ichimoku, and Moving Average (MA) indicator data of the global gold market were recorded for ten months. The moth-flame optimization (MFO) algorithm was used to determine important features in the created dataset. A hybrid neural network consisting of CNN and BiGRU was developed to make the prediction. As a result of the tests, it was seen that the developed algorithm could determine buy–sell signals with 94% accuracy.

Wei et al. [10] proposed a hybrid approach called AE-MLP for DDoS attack detection and classification. With the Auto Encoder (AE) part, it automatically finds the most necessary features in the network traffic and determines the features to be used. The MLP part enables attack detection by using the reduced feature sets determined by AE as input. Thanks to the AE-MLP algorithm, appropriate features are selected, reducing training costs and allowing more accurate detection. With the MLP algorithm, not only attack detection but also attack types are classified. Tests were carried out by creating 6 different subdatasets from the dataset used. It is intended to classify network traffic of LDAP, MSSQL, NetBIOS, SYN, UDP, and BENIGN types. As a result of the tests, it was seen that the proposed algorithm reached an accuracy rate of 98.34%.

Kumar et al. [11] developed an artificial neural network algorithm with decision trees, random forest, KNN, and naive Bayes algorithms with the Bot-IOT dataset. Later, this algorithm was tested using trafic data from a testbed consisting of 20 IoT devices that had been established. The created dataset includes 3 M DoS, 2.5 M DDoS UDP, 2.5 M DDoS TCP, and 2 M normal traffic data. In the tests performed with this dataset, the best results were achieved with KNN (acc. 99.466%) and hybrid (99.611%) algorithms. The artificial neural network algorithm was tested separately with different activation functions. The best result was obtained using the Rectified Linear Unit (ReLU) function with an accuracy rate of 99.529%. As a result of these comparisons, it is emphasized that machine learning can provide better results in systems with fewer resources and deep learning algorithms can provide better results in systems where more data and resources can be used.

Alzahrani R.J and Alzahrani A. [12] presented an evaluation of six different machine learning algorithms, namely KNN, SVM, NB, DT, RF, and LR, using the CICDDoS2019 dataset through the WEKA tool. In the feature selection process within the dataset, the Random Forest Regressor (RFR) feature selection method was identified as contributing to the accuracy of ML methods in detecting attack traffic. In the assessment, the DT and RF algorithms achieved the highest accuracy rates of 99%. It was also observed that the DT algorithm had a faster computation time compared to RF. The study additionally provided a review encompassing the strengths, weaknesses, and detection methods of both ML- and DL-based IDS systems.

Batchu and Seetha [2] have developed an anomaly detection system based on a feature selection algorithm. The parameters of the algorithms were adjusted to give the best results. In this method, machine learning algorithms, including LR, DT, GB, KNN, and SVM, were evaluated. The evaluation was conducted on the CICDDoS2019 dataset with four different scenarios and various situations. The most successful result was achieved using the GB algorithm on a dataset prepared with hybrid feature selection and hyperparameter tuning algorithms, reaching an accuracy rate of 99.97%.

Patil et al. [13] proposed a Spark Streaming and Kafka-based classification system called SSK-DDoS to classify DDoS attack types in real time. The system was trained and tested to classify six attack types (DDoS-DNS, DDoS-LDAP, DDoS-MSSQL, DDoS-NetBIOS, DDoS-UDP, and DDoS-SYN) and normal traffic found in the CICDDoS2019 dataset. The developed algorithm stores the formulated features with their evaluated classes in HDFS

and can be reused during retraining. As a result of the experiments, it was shown that the proposed detection system divided the network traffic into seven classes, with an accuracy rate of 89.05%.

Al and Dener [3] presented the STL-HDL method, a classification-based anomaly detection system for network traffic in the big data environment. The developed algorithm is a hybrid model combining CNN and LSTM. To address imbalanced distribution in the datasets, SMOTE and Tomek links techniques were applied. The model was evaluated for binary classification using the UNS-NB15 dataset and for multiclass classification using the CIDDS-001 dataset. The proposed approach compared nine different machine and deep learning algorithms. The system's success rate was 99.17% for binary classification and 99.83% for multiclass classification.

Haq et al. [14] presented the PCCNN method against attacks on IoT devices by combining the Principal Component Analysis (PCA) technique, to reduce feature size, and the 13-layer CNN algorithm, used for attack classification. This method was evaluated using the NSL-KDD dataset. The accuracy rate of the method was measured as 99.34% and 99.13% for binary and multiclass classification, respectively.

Iwendi et al. [15] proposed a deep learning-based anomaly detection system to detect DDoS attacks on IoT devices. This system was implemented using LSTM. As a result of these experiments using the CICDDOS2019 dataset, an accuracy rate of 99.97% was achieved in SNMP attack detection.

Gamal et al. [16] presented a new IDS method called CNN-IDS in their study. The proposed system consists of two stages: the first stage is the feature selection in the dataset called the ınformation gain method, and the second stage is the classification of network traffic with the one-dimensional CNN algorithm. The proposed model was trained and validated using UNSW-NB15 and Bot-IoT datasets. A detection rate of 99.9% was achieved in the Bot-IoT dataset.

Gad et al. [17] presented an IDS based on the TON_IOT dataset. Issues such as missing values and class imbalance in the TON_IOT dataset were addressed. They tried to prevent the class imbalance and overlearning problem of the dataset by using the SMOTE technique for class balancing. The Chi2 technique was used for feature selection. By reducing the number of features to 20, faster training time was achieved, and the complexity of the model was reduced. The preprocessed dataset was tested with LR, NB, DT, RF, AdaBoost, KNN, SVM, and XGBoost algorithms. The best result was achieved with the XGBoost algorithm, with an accuracy rate of 99.3% in binary and 98.6% in multiclass classification.

Disha and Waheed [18] developed an anomaly detection system based on a feature selection algorithm. An algorithm called Gini Impurity-Based Weighted Random Forest (GIWRF) was developed as a feature selection algorithm. In this context, two different datasets were used in the experiments. These are the UNSW-NB 15 and TON_IoT datasets. In total, 15 features were selected in the UNSW-NB 15 dataset, and 10 features were selected in the TON_IoT dataset, and the performance was compared compared to the situation where no selection was made. The experiments were conducted using DT, AdaBoost, GBT, MLP, LSTM, and GRU algorithms. When using feature selection, the AdaBoost and GBT algorithms achieved the highest accuracy rate in the TON_IoT dataset, with a rate of 99.98%. Compared to no feature selection, 0.4 higher success was achieved in the DT algorithm, and 0.1 higher success was achieved in the AdaBoost algorithm. There was a decrease in accuracy in the MLP, LSTM, and GRU algorithms.

Kaur et al. [19] developed a two-stage anomaly detection system called P2ADF to detect Man-İn-The-Middle (MİTM) and DoS/DDoS attacks. There is a feature reduction mechanism to remove the dominant features in the datasets used in the first stage. In the second stage, three basic learners (AdaBoost, LR, and KNN) and a meta classifier (XGBoost) were used for model training. IoTID20, TON_IoT, N-BaIoT, UNSW-NB15, and CICDoS19 datasets were used for the experiments. For DDoS detection, 99.98% accuracy values were achieved with CICIDS2019DDoS LDAP and 99.95% accuracy values for TON_IoT DDoS.

Verma and Chandra [20] developed an algorithm called RepuTE for the IoT, which enables the detection of DoS/DDoS and Sybil attacks. In the presented RepuTe algorithm, live traffic coming from the IoT layer is transferred to the fog layer and the traffic data is first pre-processed in this layer. The model, consisting of extra tree, KNN, and quadratic discriminant analysis algorithms, evaluates this incoming live traffic data. In order for the model to reach the highest accuracy rate, a method called soft-voting was applied among these algorithms. This study was evaluated on NSL-KDD, CICDDOS2019, IoTID20, NBaIoT2018, TON_IoT, and UNSW_NB15 datasets. In attack detection, an accuracy value of 99.9988% was achieved with CICDDoS2019_NTP and 99.9851% with TON_IoT_DDoS.

Neto et al. [21] proposed a comprehensive IoT attack dataset to improve the security of IoT operations in their study. To achieve this, a topology consisting of 105 IoT devices was established. With this mechanism, 33 different attack types of traffic were recorded in seven categories. These attack categories are DDoS, DoS, Recon, Web-based, brute force, spoofing, and Mirai. The created dataset is called CICIoT2023. Using the developed dataset, the accuracy values of various machine learning algorithms were tested.

Wang et al. [22] designed the DL-BiLSTM algorithm as a lightweight IoT attack detection model. This algorithm was developed by combining deep learning algorithms DNN and BiLSTM algorithms. CICIDS2017, N-BaIoT, and CICIoT2023 datasets were used in the study. The Incremental Principal Component Analysis (IPCA) algorithm was applied to reduce the features in the datasets. Additionally, dynamic quantization is applied to reduce the computational burden of the model. The developed algorithm was compared with different deep learning algorithms. In the CICIoT2023 dataset, the attack type was detected with an accuracy rate of 93.13%, reaching the highest accuracy rate among the tested algorithms.

Studies on IDS studies conducted in WSN are presented in Table 1. This table includes year, author, algorithm used, dataset used, and accuracy information. Several different algorithms using machine learning and deep learning models have been developed in the literature. With developing technology, the structure of DDoS attacks has changed, and the datasets used have become obsolete. Therefore, the current CICIoT2023 dataset was used in this study. To test the reliability of the algorithm, the TON_IOT dataset was also used.

Table 1. Comparison of other works on DDoS Detection.

Years	Author	Model	Dataset	Accuracy
2021	Cil et al. [4]	DNN	CICDDoS2019	99.97% (b), 94.57% (m)
2022	Almaraz-Rivera et al. [5]	DT, MLP, RNN, RF, GRU, LSTM, and SVM	Bot-IoT	99.972% (b), 99.945% (m)
2022	Jia et al. [6]	DNN, LSTM	CICDDoS2019	99.9% (b), 98.9% (m)
2021	Alghazzawi et al. [7]	CNN, BiLSTM	CICDDoS2019	94.52% (b)
2021	Ferrag et al. [8]	CNN, RNN, and DNN	CICDDoS2019,	99.95% (b), 95.12% (m)
			TON_IoT	99.92% (m)
2021	Wei et al. [10]	MLP	CICDDOS2019	99.96% (b), 98.34% (m)
2022	Kumar et al. [11]	DT, RF, KNN, NB, and ANN	Bot-IOT	99.611% (m)
2021	Al and Dener [3]	CNN, LSTM	UNS-NB15 and CIDDS-001	99.17% (b), 99.83% (m)

Table 1. *Cont.*

Years	Author	Model	Dataset	Accuracy
2021	Alzahrani and Alzahrani [12]	SVM, KNN, DT, NB, RF, and LR.	CICDDOS2019	99% (m)
2021	Batchu and Seetha [2]	LR, DT, GB, KNN, and SVM	CICDDOS2019	99.97% (b)
2022	Patil et al. [13]	DT, MLP, NB, and RF	CICDDOS2019	89.05% (m)
2021	Haq et al. [14]	CNN	NSL-KDD	99.34% (b), 99.13% (m)
2021	Iwendi et al. [15]	LSTM	CICDDOS2019	99.97% (b)
2021	Gamal et al. [16]	CNN	UNSW-NB15 and Bot-IoT	99.9% (b)
2021	Gad et al. [17]	LR, NB, DT, RF, AdaBoost, KNN, SVM, and XGBoost	TON_IoT	99.3% (b), 98.6% (m)
2022	Disha and Waheed [18]	DT, AdaBoost, GBT, MLP, LSTM, and GRU	TON_IoT, UNSW-NB 15	99.98% (b)
2023	Kaur et al. [19]	AdaBoost, LR, and KNN	CICDDOS2019, TON_IoT, IoTID20, N-BaIoT, UNSW-NB15	99.98 (b) (LDAP), 99.95% (b) (DDoS)
2023	Verme and Chandra [20]	Extra Tree, KNN and Quadratic Discriminant Analysis	CICDDOS2019, TON_IoT, NSL-KDD, IoTID20, NBaIoT2018, UNSW_NB15	99.9988% (b) (NTP), 99.9851% (b) (DDoS)
2023	Neto et al. [21]	RF, DNN, MLP, LR, AdaBoost	CICIoT2023	99.68% (b), 99.43% (m) (8 classes)
2023	Wang et al. [22]	DL-BiLSTM	CICIoT2023	93.13% (m)

Note: b—binary classification; m—multiclass classification.

It has been observed that IDSs that can analyze network traffic and detect DDoS attacks using machine and deep learning models have reached a high success rate. The proposed hybrid algorithm developed by combining CNN and LSTM was evaluated by accuracy, precision, recall, and F1 parameters and compered with mostly used machine learning and deep learning algorithms. All these studies were developed in a big data environment. The classification performance of the developed hybrid algorithm as both binary and multiclass was measured.

3. DDoS Attacks and Intrusion Detection System for IoTs

This section provides information about DDoS attacks in IoT and the Intrusion Detection Systems (IDS) used to detect them.

3.1. Intrusion Detection Systems

As communication technologies continue to evolve, the security of the devices used and the network they create have become very crucial. Cyber attacks are defined as all attempts that can threaten the confidentiality, integrity, and accessibility of information [23]. The integrity and confidentiality of the data stored must be protected from these attacks. Intrusion detection systems are used for this purpose. These systems distinguish any attack from normal traffic and can warn the user in case of an attack. An IDS, whose general working mechanism is given in Figure 1, records all activities that differ from normal traffic as anomalies. These systems are also prone to false alarms. Therefore, it is very important that the IDSs that are developed have a high accuracy value. In this study, not only attack detection but also determination of the attack type were made with a high accuracy rate.

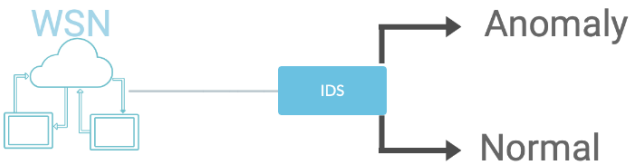

Figure 1. Intrusion detection systems.

3.2. DDoS

A DDoS attack is a malicious attempt to disrupt the normal traffic of the target system. DDoS attacks aim to disrupt service by sending packets that exceed the capacity of targeted source machines to respond to requests. Attackers use zombie computers created with malware inserted into victims' computers to send large amounts of packets. DDoS attacks cause high network traffic with packets sent over the network, causing the system not to respond to the requests of normal users who want to receive service [4]. A DDoS attack is one of the biggest threats to internet-based applications and their resources. The aim of this attack is to incapacitate internet-based services by transmitting a substantial volume of attack traffic [13]. Since the attack types are given in the main category in the TON_IOT dataset, the details of these attacks are not known. For example, it is not given which types are included under the main category of DDoS. In the CICIoT2023 dataset, subcategories of attack types are also given. This dataset includes different attack types under the headings of flood and fragmentation. These attacks are described below.

In a SYN flood attack, the attacker consumes the resources of IoT devices by repeatedly sending half-open synchronization packets for the TCP connection request. These connections are left open for further communication [4]. The victim machine, using all available ports, may respond slowly or not at all to legitimate traffic. A UDP flood attack is an attack in which large packets are sent by attackers without any permission using User Datagram Protocol (UDP), a fast data-sharing protocol [4]. The ICMP (Internet Control Message Protocol) is the network protocol used for IP control/error reporting. In the ICMP flood attack, the attacker aims to take the network offline by sending too many ICMP requests [24]. There are RSTFIN flood attacks made with FIN and RST packets in the TCP protocol. The FIN packet is sent to securely terminate the TCP connection between the current client and server. The RST packet is sent by the server in abnormal situations and is used to forcibly close the connection. In the RSTFIN flood attack, the attacker causes congestion of the system by sending FIN and RST packets that do not belong to the target network [24]. The main focus of the HTTP flood attack is on generating attack traffic that simulates a close resemblance to normal network traffic. Thus, it becomes difficult for the victim to distinguish between legitimate traffic and attack traffic. In the HTTP flood attack, the aim is to exhaust the server's resources by ensuring that session connection request rates are higher than those generated by legitimate users [25]. In the Slowloris attack, HTTP sends the request in pieces and slowly, and the created request is not completed. As a result, the server keeps the relevant connection in a waiting phase to complete the connection and receive the necessary data. In this way, over time, open connection requests

increase and cause the congestion of the system. [25]. The PSH packet used in the PSHACK flood attack is used to ensure that the client receiving this command sends all data to a specified application and that the data are processed. Packets with combinations of PSH and ACK are often seen in normal incoming traffic [26]. The attacker can create a PSHACK flood attack by intensively directing these packet combinations to the target server. A synonymous IP flood attack is a type of DDoS attack that aims to consume the resources of DNS servers by sending a high volume of requests for a nonexistent domain. This attack, which uses the TCP protocol, uses high-speed packets [27].

In fragmentation attacks, the aim is to send packets larger than the MTU (Maximum Transfer Unit) limit that can be transferred at the network entrance, thus ensuring that they are fragmented and sent. This size is 1500 bytes in Ethernet network. Fragmentation attacks are performed by sending frames higher than this value. An ACK fragmentation attack is a version of the ACK and PUSH-ACK flood attack. Fragmented packets pass through switches, firewalls, IDS, and IPS because the router does not reassemble fragmented frames. These packages may contain random and irrelevant information. With this attack, the aim is for the victim to consume resources [28]. In the ICMP fragmentation attack, fragmented ICMP packets are used.

The victim is exposed to ICMP packets that cannot be reassembled. Since these packets contain random and irrelevant information, the victim's resources are consumed by trying to combine them [28]. A UDP fragmentation attack is an adaptation of a UDP flood attack. Since the fragmented UDP packets are deceptive and unrelated to each other, the target server wastes its resources by trying to reassemble them. This type of attack causes the victim's CPUs to overheat and consume their resources unnecessarily [28].

4. Materials and Methods

In this section, information about the CICIoT2023 and TON_IOT datasets used in the study is provided. Afterwards, the preprocessing steps of the dataset are explained. Then, the deep learning algorithms used in the study are defined.

4.1. Dataset

Information about the CICIoT2023 and TON_IOT datasets used in the study is included in Section 4.

4.1.1. CICIoT2023

This dataset was produced by Neto et al. [21] and published in the University of New Brunswick (UNB)—Canadian Institute for Cybersecurity (CIC) database. An IoT topology consisting of 105 IoT devices was established. A total of 67 IoT devices were directly involved in the attacks, and another 38 Zigbee and Z-Wave devices were connected to five hubs. This topology is designed to mimic a real physical IoT smart home environment. The testbed consists of smart home devices, cameras, sensors, and microcontrollers that are connected and configured to allow the execution of various attacks. The test environment is also equipped with various tools and software that allow it to perform various attacks and capture both benign and malicious attack traffic. The testbed produced 33 different attack types. These attacks are classified into seven categories: DDoS, DoS, Recon, Web-based, brute force, spoofing, and Mirai. The dataset contains 47 features. Details of these features are given in Table 2.

Table 2. List of features of CICIoT2023.

Feature	Description
ts	Timestamp
flow_duration	Duration of the packet's flow
Header_Length	Header length

Table 2. *Cont.*

Feature	Description
ProtocolType	IP, UDP, TCP, IGMP, ICMP, Unknown (integers)
Duration	Time-to-live (ttl)
Rate	Rate of packet transmission in a flow
Srate	Rate of outbound packets' transmission in a flow
Drate	Rate of inbound packets' transmission in a flow
fin_flag_number	FIN flag value
syn_flag_number	SYN flag value
rst_flag_number	RST flag value
psh_flag_number	PSH flag value
ack_flag_number	ACK flag value
ece_flag_number	ECE flag value
cwr_flag_number	CWR flag value
ack_count	Number of packets with ACK flag set in the same flow
syn_count	Number of packets with SYN flag set in the same flow
fin_count	Number of packets with FIN flag set in the same flow
urg_count	Number of packets with URG flag set in the same flow
rst_count	Number of packets with RST flag set in the same flow
HTTP	Indicates if the application layer protocol is HTTP
HTTPS	Indicates if the application layer protocol is HTTPS
DNS	Indicates if the application layer protocol is DNS
Telnet	Indicates if the application layer protocol is Telnet
SMTP	Indicates if the application layer protocol is SMTP
SSH	Indicates if the application layer protocol is SSH
IRC	Indicates if the application layer protocol is IRC
TCP	Indicates if the application layer protocol is TCP
UDP	Indicates if the application layer protocol is UDP
DHCP	Indicates if the application layer protocol is DHCP
ARP	Indicates if the application layer protocol is ARP
ICMP	Indicates if the application layer protocol is ICMP
IPv	Indicates if the application layer protocol is IPv
LLC	Indicates if the application layer protocol is LLC
Totsum	Summation of packets' lengths in flow
Min	Minimum packet length in the flow
Max	Maximum packet length in the flow
AVG	Average packet length in the flow
Std	Standard deviation of packet length in the flow
Totsize	Packet's length
IAT	The time difference with previous packet
Number	The number of packets in the flow
Magnitue	(Average of the lengths of incoming packets in the flow + average of the lengths of outgoing packets in the flow) $\times$ 0.5

Table 2. *Cont.*

Feature	Description
Radius	(Variance of the lengths of incoming packets in the flow + variance of the lengths of outgoing packets in the flow) × 0.5
Covariance	Covariance of the lengths of incoming and outgoing packets
Variance	Variance of the lengths of incoming packets in the flow/variance of the lengths of outgoing packets in the flow
Weight	(Number of incoming packets) × (Number of outgoing packets)

Table 3 shows the number of attacks in the dataset and the attack classes they belong to. Figure 2 shows the distribution graph of attacks according to attack class. As can be seen in the figure, the DDoS is the most common class in the dataset, with 73%. This is followed by the DoS class, with 17%. Web-based and brute force attack classes are below 1%.

Table 3. Count of attack type of CICIoT2023.

Label	Class	Count
DDoS-ICMP_Flood	DDoS	7,200,047
DDoS-UDP_Flood	DDoS	5,411,768
DDoS-TCP_Flood	DDoS	4,497,763
DDoS-PSHACK_Flood	DDoS	4,094,563
DDoS-SYN_Flood	DDoS	4,059,403
DDoS-RSTFINFlood	DDoS	4,045,410
DDoS-SynonymousIP_Flood	DDoS	3,598,454
DoS-UDP_Flood	DoS	3,318,467
DoS-TCP_Flood	DoS	2,671,471
DoS-SYN_Flood	DoS	2,028,995
BenignTraffic	Normal	1,098,282
Mirai-greeth_flood	Mirai	991,846
Mirai-udpplain	Mirai	890,708
Mirai-greip_flood	Mirai	751,891
DDoS-ICMP_Fragmentation	DDoS	452,557
MITM-ArpSpoofing	Spoofing	307,598
DDoS-UDP_Fragmentation	DDoS	286,968
DDoS-ACK_Fragmentation	DDoS	285,089
DNS_Spoofing	Spoofing	178,902
Recon-HostDiscovery	Recon	134,375
Recon-OSScan	Recon	98,269
Recon-PortScan	Recon	82,267
DoS-HTTP_Flood	DoS	71,844
VulnerabilityScan	Recon	37,379
DDoS-HTTP_Flood	DDoS	28,795
DDoS-SlowLoris	DDoS	23,414
DictionaryBruteForce	Brute Force	13,048
BrowserHijacking	Web-Based	5858

Table 3. *Cont.*

Label	Class	Count
CommandInjection	Web-Based	5419
SqlInjection	Web-Based	5253
XSS	Web-Based	3852
Backdoor_Malware	Web-Based	3221
Recon-PingSweep	Recon	2262
Uploading_Attack	Web-Based	1253
	Total	46,686,691

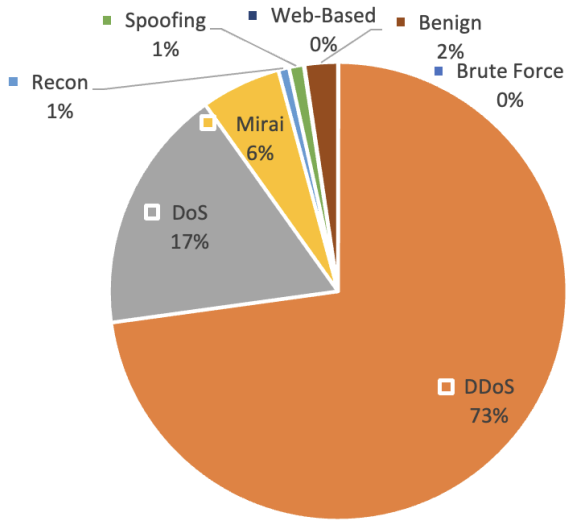

Figure 2. Distrubition of CICIoT2023.

4.1.2. TON_IOT

The TON_IoT [29] dataset was collected in a realistic, large-scale test environment at the UNSW Canberra Cyber Institute's IoT Lab in 2019. The dataset includes a number of modern IoT attacks, such as scanning, DoS, DDoS, ransomware, backdoor, injection, cross-site scripting (XSS), password cracking, and Man-In-The-Middle (MITM) attacks. Figure 3 shows a data structure of the TON_IoT dataset. In this study, the Processed Windows 10 dataset, which is a subset of the TON_IoT, was used.

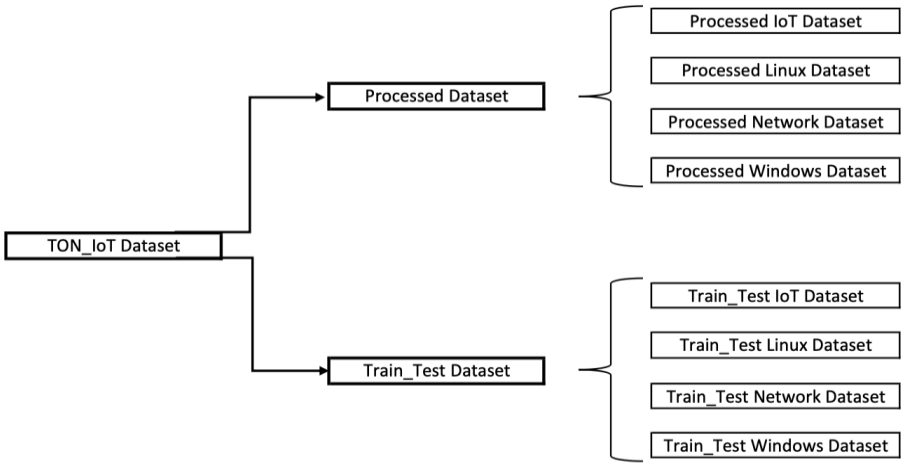

Figure 3. Data structure of TON_IoT dataset.

The test environment where the dataset would be recorded was designed with three layers consisting of edge, fog, and cloud to emulate a realistic physical IoT network. The edge layer includes the physical devices and operating systems required for the infrastructure required for configuration and virtualization technologies to be used in the fog and cloud layers. This layer includes multiple IoT/IIoT devices, such as Modbus, light bulb sensors, smartphones, smart TVs; host systems, such as workstations and servers used to intercept IoT/IIoT devices; hypervisors; and physical network relays [30]. The fog layer encompasses the virtualization technology responsible for managing virtual machines (VMs) and their associated services. This layer enables the creation of a dynamic experimental IoT/IIoT network within the ToN IoT framework, allowing communication between the edge, fog, and cloud layers [30]. The cloud layer contains online-configured cloud services within the testbed. The fog and edge services connect cloud virtualization and cloud data analytics services. Additionally, the public vulnerable website is used to create injection hacking events against websites. The other cloud services are set up to transmit sensor data to the cloud and visualize the corresponding patterns [30].

The created dataset was made more suitable for artificial intelligence training and testing by using the performance monitoring tool and extracting information such as disk, process, processor, and memory in CSV format. The Windows 10 dataset contains 125 features and 2 class labels. Descriptions of the features in TON_IoT dataset are available on UNSW Canberra Cyber Institute database [31]. As given in Table 4, there are 35,974 records collected for the Windows 10 dataset.

Table 4. Count of attack type Processed Windows 10 of TON_IOT.

Label	Windows 10
Normal	24,871
DDoS	4608
Injection	612
XSS	1268
Password	3628
Scanning	447
DoS	525
MITM	15
Total	35,974

4.2. Preprocessing

It is not appropriate to use datasets in deep learning algorithms without preprocessing. Data collected from real-world environments often contain many errors and irregularities and need to be cleaned. For example, if there are string values in the dataset, they cannot be used in deep learning training without numerical conversion. Preprocessing aims to provide the algorithm with smoother data, thereby enhancing the efficiency of the model. Figure 4 presents a flow diagram of the data processing stages of proposed algorithm.

The first operation performed on the dataset is to delete cells that do not contain any data and remove blank values. Rows containing empty data were removed to prevent any negative effects on the model.

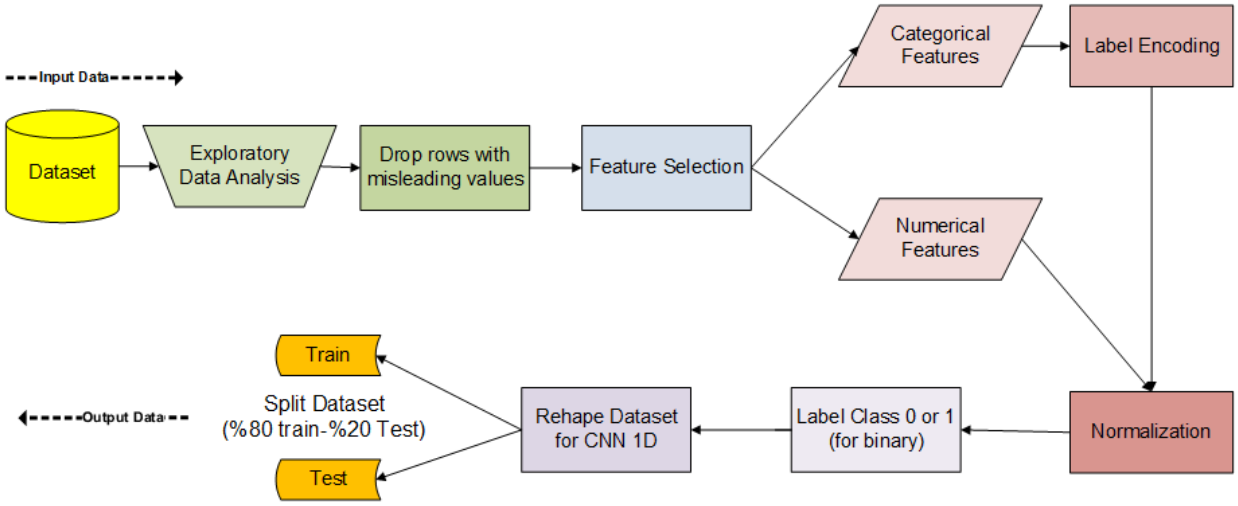

Figure 4. Flow diagram of preprocess.

It is not always beneficial to include all features in large datasets in training. Features in the dataset may be correlated with each other and may not benefit the result. Additionally, having too many values also increases the cost of education. In order to see unnecessary attributes, the correlation matrix of the features in the dataset is extracted, and the features with high correlation values are removed from the dataset. In this study, feature selection was made using the Pearson correlation coefficient method. The PCC Formula (1) is given below. Here, μ is the mean of variable, and σ is the standard deviation.

$$PCC(X,Y) = \frac{\Sigma[(Xi-\mu_x)(Yi-\mu_y)]}{\sigma x.\sigma y} \quad (1)$$

Correlation resulted between −1 and 1. This means that when the PCC approaches positive 1, it signifies a positive correlation between the two variables. This implies that when one variable decreases or increases, the other positively correlated variable also moves in the same direction. Similarly, two variables that are negatively correlated behave in the reverse direction [32]. Features with correlation values above a certain value are removed. The correlation matrix of the CICIoT2023 dataset is given in Figure 5. This value was determined as 0.99 for the dataset given the correlation matrix. As a result, 40 features were selected in the CICIoT2023 dataset and 85 features were selected in the Processed Windows 10 of TON_IOT dataset.

The next process applied to the dataset is label encoding, which involves converting non-numeric features into numerical values. Label encoding has been applied to categorical features within the dataset. There is no need to apply this process to numeric features in the dataset. Following the numerical conversion of string values, normalization (2) has been performed on the dataset.

$$x' = (x - \mu)/\sigma \quad (2)$$

x is the original value, x' is normalized value, and μ and σ are the mean and standard deviation values, respectively. Thanks to the normalization process, numerically large features are prevented from negatively affecting the result and performance of the deep learning model [3]. Concerning the label attribute of the dataset, if binary evaluation is performed, normal values are labeled as 0 and attack values are labeled as 1. This step is not performed if multiclass classification is conducted. The next process is to add a new dimension to the data shape and make it compatible for the CNN layer. The last stage of data processing is to divide the dataset into two: training and testing. This ratio was chosen as 0.8.

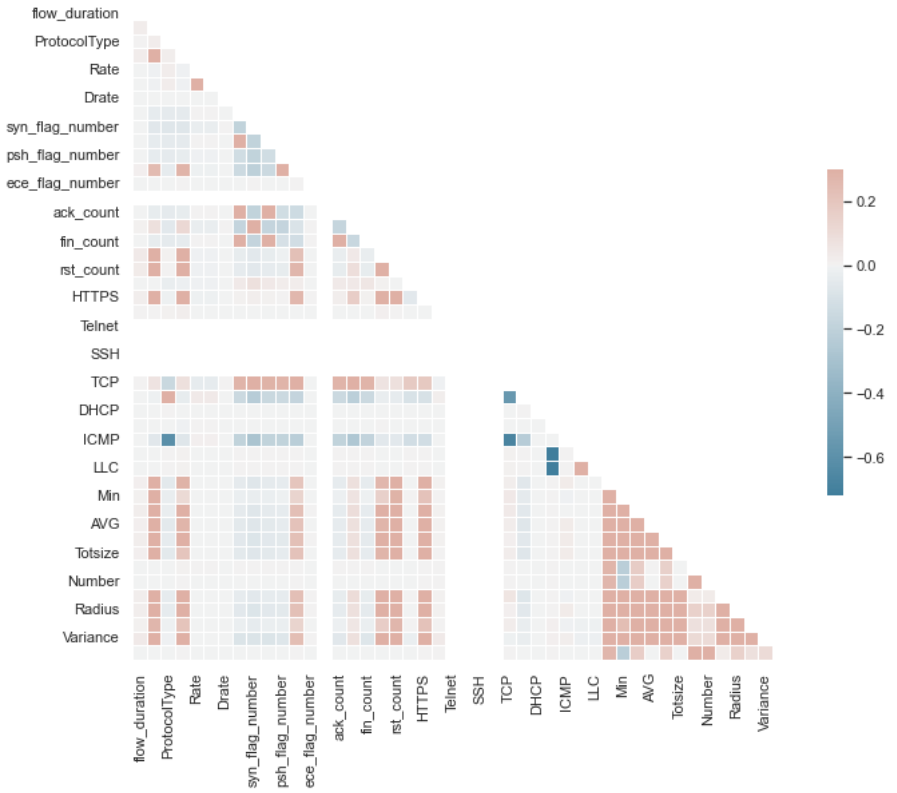

Figure 5. Diagonal correlation matrix of CICIoT2023 dataset.

After the steps shown in Figure 4 were completed in order, the attack classes in the dataset were selected. There are 12 different attack types under the category of DDoS in the CICIoT2023 dataset. By removing the least common DDoS attacks, which are UDP fragmentation, ACK fragmentation, HTTP_Flood and SlowLoris, 8 DDoS attack types were selected. Since there is no subtype of DDoS attack in the TON_IoT dataset, only binary evaluation was made here by taking only DDoS classes.

4.3. Deep Learning Algorithms

The fundamental logic in machine learning and deep learning algorithms involves creating a model by learning features extracted from the dataset during training. Subsequently, this model is used to make predictions for unknown data. Artificial intelligence-based anomaly detection systems utilize this structure to enable the developed model to detect abnormal situations in the network. This section explains the deep learning algorithms used in the study.

4.3.1. Convolutional Neural Network (CNN)

The CNN algorithm is one of the deep learning algorithms based on an artificial neural network. The basic algorithm of CNN is given in Figure 6. It consists of convolution, pooling, flattening, and fully connected layers. The convolution layer is the cornerstone of CNN. The convolutional layer is responsible for processing data from a receiving cell. Equation (3) for the size of the output volume (Wo) is described as follows, where P is the stride, Wi is the size of the input volume, S is the kernel size of the convolutional layer neurons, and M is the amount of zero padding [8].

$$Wo = \frac{Wi - S + 2M}{P} + 1 \quad (3)$$

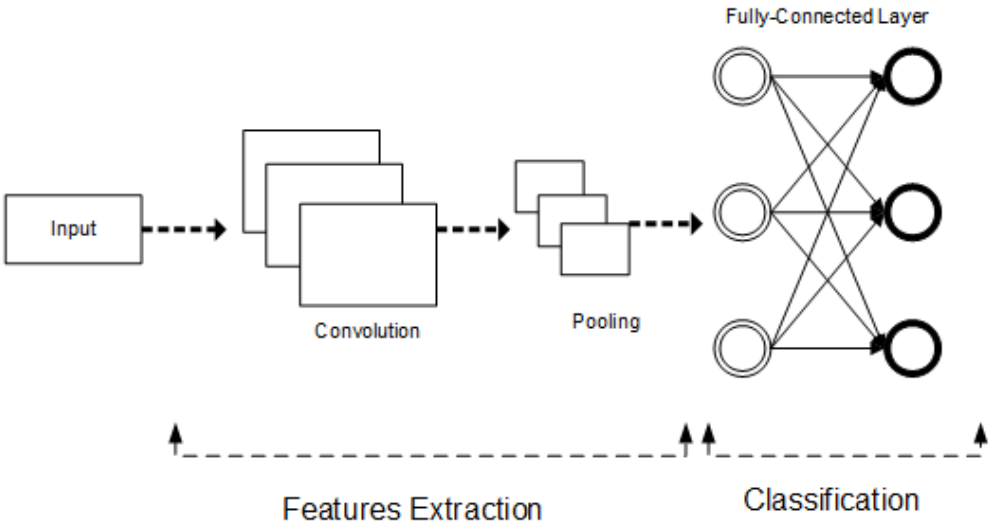

Figure 6. CNN basic algorithm.

After, convolution is performed with a filter that extracts the characteristics of the input value. The feature map is created with this layer. In order to reduce the number of parameters to be calculated and to make training easier, the size of the input data is reduced by pooling. The pooling layer can be selected in two different ways: the largest of the values within the area of the selected size (max pooling) or the average of the values (average pooling). There may be more than one convolution/pooling layer in the created algorithm. This stage is known as feature extraction. The feature extracted data become available for calculation. The CNN algorithm can be one-dimensional, two-dimensional, or three-dimensional. One-dimensional CNN was used in the model. The differences between 1D, 2D, and 3D CNN are as follows:

- In 1D CNN, the filter moves in one dimension. Input and output data must be two-dimensional. It can be used in time series-type data.
- In 2D CNN, the filter moves in two dimensions. Input and output data must be three-dimensional. It can be used in algorithms that use images as input.
- In 3D CNN, the filter moves in three dimensions. Input and output data must be four-dimensional. It can be used in algorithms that use video as input.

The classification region consists of layers known as flattened and fully connected. After the convolutional stage, the data need to be flattened to be usable in the fully connected stage. This step is carried out in the flatten layer. The fully connected layer uses the classical artificial neural network model. Classification results are determined by calculating the weight values in this layer. In order to apply CNN to nonimage data, the dimensions of input must be transformed. Thus, CNN can be used with one-dimensional convolutional layers [33].

4.3.2. Long Short-Term Memory (LSTM)

The LSTM algorithm is a type of RNN that can learn long-term dependencies and retain sequential data in memory. It solves the vanishing gradient problem caused by gradual decay in gradient inversion operations during calculation. LSTM is an algorithm suitable for use in matters related to time series [34]. Thanks to these features, it can be used in algorithms such as language processing, video processing, and speech recognition. The LSTM algorithm consists of memory blocks called cells, and these are the main components of the algorithm. The LSTM algorithm is shown in Figure 7. The LSTM algorithm consists of three parts: forget gate (f_t), input gate (i_t), and output gate (o_t). Input and output gates represent the input and output of data at time t. The forget gate decides whether the data will be forgotten or not by comparing instantaneous data inputs with the previous data state [3].

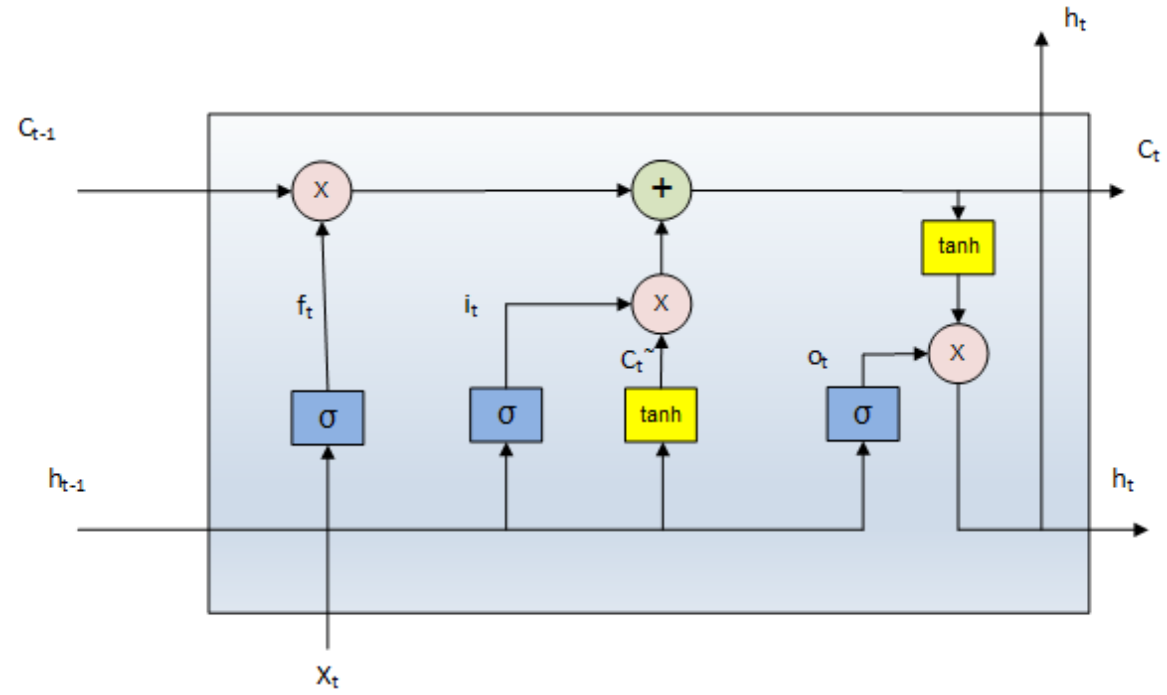

Figure 7. LSTM basic algorithm.

The mathematical equation describing the relationship between the gates in an LSTM cell is as follows [3]:

$$i_t = \sigma(w_i \cdot [h_{t-1}, X_t]) \quad (4)$$

$$f_t = \sigma(w_f \cdot [h_{t-1}, X_t]) \quad (5)$$

$$o_t = \sigma(w_o \cdot [h_{t-1}, X_t]) \quad (6)$$

$$C_t^{\sim} = \tanh(w_t \cdot [h_{t-1}, x_t]) \quad (7)$$

$$C_t = f_t \times C_{t-1} + i_t \times C_t^{\sim} \quad (8)$$

$$h_t = o_t \times \tanh(C_t) \quad (9)$$

5. Definition of Model

A hybrid deep learning model was developed using one-dimensional (1D) CNN and LSTM algorithms to detect DDoS attacks. The model has a sequential algorithm consisting of layers. Figure 8 shows the flow diagram of the proposed model.

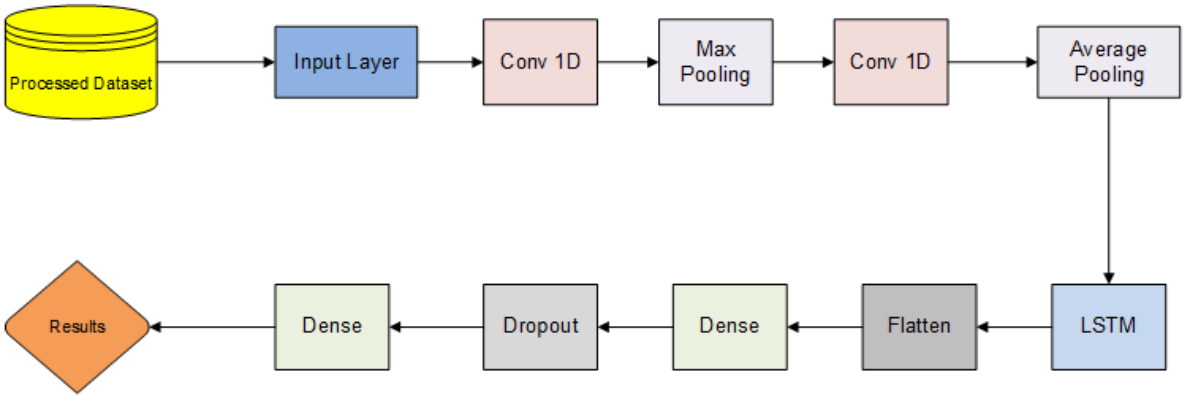

Figure 8. Flow diagram of proposed model.

The developed sequential model starts with the input layer. This layer describes the dataset size used. As seen in Figure 8, 2 CNN algorithms were applied. The parameter settings of the CNNs used were adjusted to provide the best results. These parameters are presented in tabular form in Table 5. After the convulation process, the pooling layer was applied. The main task of the pooling layer is to reduce the dimensionality of the extracted feature matrix. While the computational load is reduced in the pooling layer, important information is preserved [35]. Max pooling was applied between the first and second CNN, and average pooling was applied after the second CNN. After CNN, the LSTM algorithm was used sequentially. The parameters of the LSTM algorithm used are set to units = 140, dropout = 0.2, and recurrent_dropout = 0.4. After the LSTM algorithm, the dimensions in the flatten layer were made suitable for the dense layer. The flatten layer is followed by dense layers known as fully connected. The dropout function was implemented at a ratio of 30%. The reason for using the dropout layer is to prevent the algorithm from overlearning. The function of this layer is to ignore some nodes randomly. This partly refers to a situation in which neurons can change the way they correct the errors of other neurons [36].

Table 5. Used CNN parameters.

	Filter Number	Kernel Size	Activation Function
Conv1D_1	128	4	Relu
Conv1D_2	128	2	Relu

The ReLu function (10) was used as the activation function in both CNN algorithms. The ReLU function was used because it provides computational simplicity and eliminates negative values.

$$f(x) = \max(0,x) \quad (10)$$

The output layer of the model ends with a softmax activation function. This function returns the probability of the maximum value for the sample evaluated in a multiclass probability problem to have the most accurate label in terms of probability [37]. With the softmax function, the result is produced as a probability distribution.

6. Experiments and Results

In this section, evaluation parameters and test results are given. In the study, PySpark, which provides the opportunity to write in Python programming language on Apache Spark, was used through the Google Colab platform. Scikit-learn and Keras libraries were used to create deep learning algorithms.

Training and testing of the model was conducted on a computer with the following configuration:

- MacOS v12.6 operating system;
- M1 Apple Silicon (2020);
- 13.3″ screen;
- 8-core CPU;
- 8-core GPU;
- 8 GB RAM;
- 256 GB SSD.

The CICIoT2023 dataset consists of more than one data file, and by combining these files, considerable data to be processed emerged. Among the studies examined, there are studies conducted by taking samples from the dataset [10,13]. This both reduces the training cost and does not have a serious impact on the outcome. At the same time, using the entire dataset consumes computer resources and makes processing inoperable. In this way, it eliminates the need to use high-capacity computers and servers, which are expensive and difficult to access. Instead of the entire dataset, a subspace set of the dataset was used, reduced to 20%. The attack class ratio of the subspace cluster is the same as the original

version. Thus, training and testing costs and time were saved. In addition to the proposed algorithm tests, the dataset was also tested with ten machine learning and deep learning algorithms.

The parameter values of the artificial intelligence and machine algorithms used are given below:

- Random forest: max_depth = 4, n_estimators = 100;
- Decision tree: max_depth = 5, random_state = 0;
- Gradient boost: n_estimators = 10, max_depth = 3, learning_rate = 0.1;
- AdaBoost: n_estimators = 10, learning_rate = 0.1, random_state = 0;
- Naive Bayes: default;
- Logistic regression: default;
- K-nearest neighbour: n_neighbors = 3, leaf_size = 50;
- MLP: hidden_layer_sizes = (5,10,5), max_iter = 5;
- CNN: filters = 64, kernel_size = 2, activation = 'relu';
- LSTM: units = 100, dropout = 0.2, and recurrent_dropout = 0.2.

The results obtained in the study were evaluated from different perspectives. Firstly, multiclass and binary evaluations of DDoS attack classes in the CICIoT2023 dataset were made. Secondly, binary evaluation was made for the TON_IOT-Windows10 dataset. The developed algorithm was compared with ten machine and deep learning algorithms in both datasets. Details of the hybrid algorithm used are explained in the "Five Definitions of Model" section. The evaluations were made with the parameters accuracy, precision, recall, and F1 Score. Additionally, ROC curves and confusion matrix graphics were also created and included in the study.

6.1. CICIoT2023 Dataset Results

The CICIoT2023 dataset, whose preprocessing was completed, was first tested as binary with the proposed hybrid algorithm. The developed algorithm has been compared with machine learning and deep learning algorithms, including random forest, decision tree, gradient boost, AdaBoost, naive Bayes, logistic regression, K-nearest neighbour, CNN, MLP, and LSTM. Table 6 shows the binary evaluation results of the algorithms for the CICIoT2023 dataset. Figure 9 shows the results graphed.

Table 6. CICIoT2023 dataset binary classification results (%).

	Accuracy	**Precision**	**Recall**	**F1 Score**
DT	99.91	99.91	99.91	99.91
RF	99.85	99.86	99.85	99.86
LR	99.86	99.86	99.86	99.86
GB	99.98	99.98	99.98	99.98
ADA	99.75	99.75	99.75	99.75
KNN	99.97	99.97	99.97	99.97
MLP	99.98	99.98	99.98	99.98
NB	99.28	99.41	99.28	99.32
CNN	99.98	99.98	99.98	99.98
LSTM	98.73	99.74	99.74	99.74
Proposed	99.995	99.995	99.995	99.995

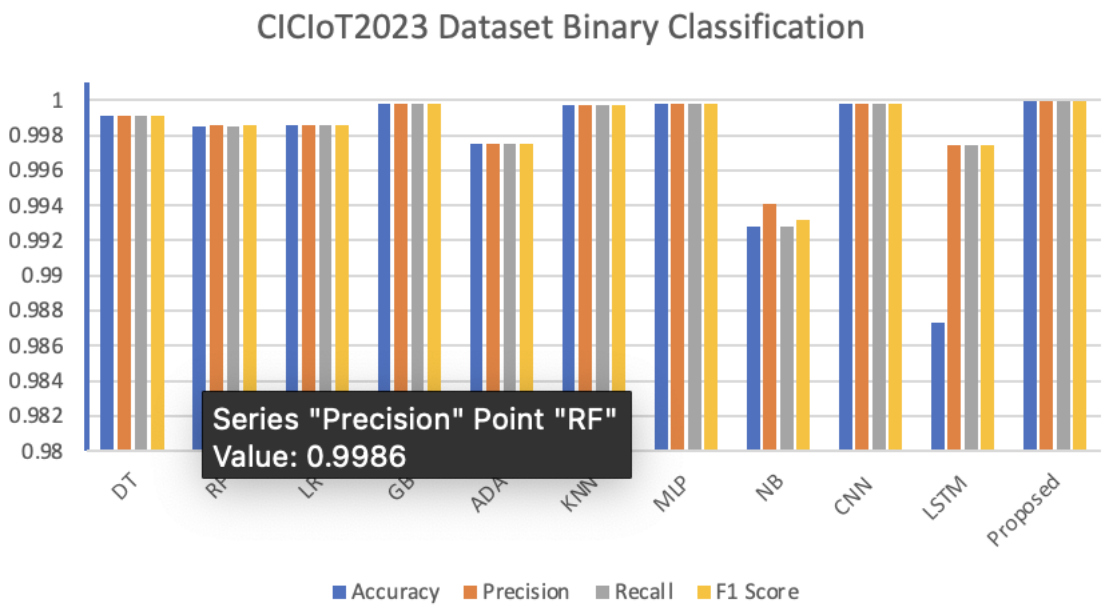

Figure 9. CICIoT2023 dataset binary classification results graph.

As can be seen in Table 6 and Figure 9, the proposed algorithm reached the highest binary classification accuracy value. This was followed by the GB, MLP and CNN algorithms. It can be seen that the lowest result is the NB algorithm. The confusion matrix for testing the developed hybrid algorithm in binary is given in Figure 10. The diagram of the created ROC curve is presented in Figure 11.

According to the confusion matrix in Figure 10, it can be seen that the false positive rate (FPR) is almost negligible; only about a hundred records were misclassified. True positive (TPR) records were quite high. According to the ROC chart in Figure 11, the AUC-ROC value was above 0.99.

The algorithms were also evaluated as multiclass classification. The developed algorithm and ten machine learning and deep learning algorithms were tested. Table 7 shows the multiclass evaluation results of the algorithms for the CICIoT2023 dataset. In Figure 12, the results are presented graphically.

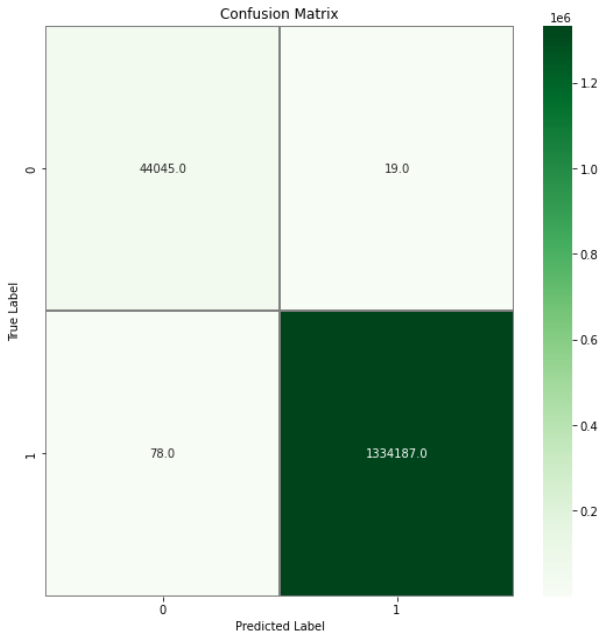

Figure 10. Confusion matrix for binary classification of CICIoT2023 dataset.

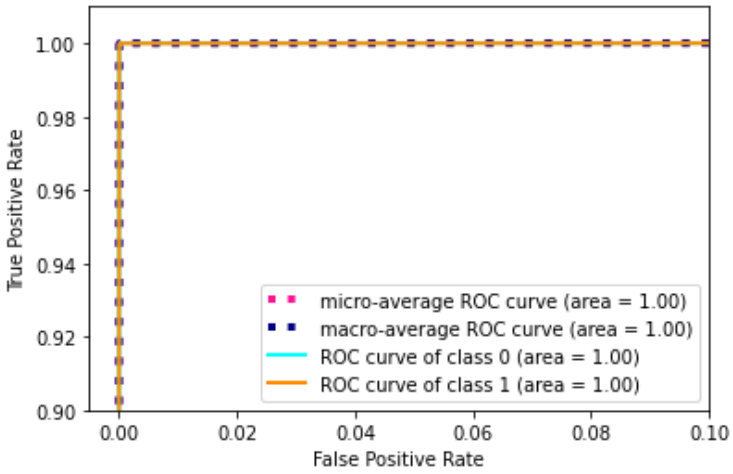

Figure 11. ROC curves for binary classification of CICIoT2023 dataset.

Table 7. CICIoT2023 dataset multiclass classification results.

	Accuracy	Precision	Recall	F1 Score
DT	86.34	82.39	86.34	82.69
RF	96.58	96.98	96.58	96.51
LR	99.43	99.44	99.43	99.43
GB	99.88	99.88	99.88	99.88
ADA	86.14	79.44	86.14	81.91
KNN	99.86	99.86	99.86	99.86
MLP	99.91	99.91	99.91	99.91
NB	99.09	99.13	99.09	99.10
CNN	99.90	99.93	99.90	99.91
LSTM	98.66	98.71	98.63	98.67
Proposed	99.96	99.96	99.96	99.96

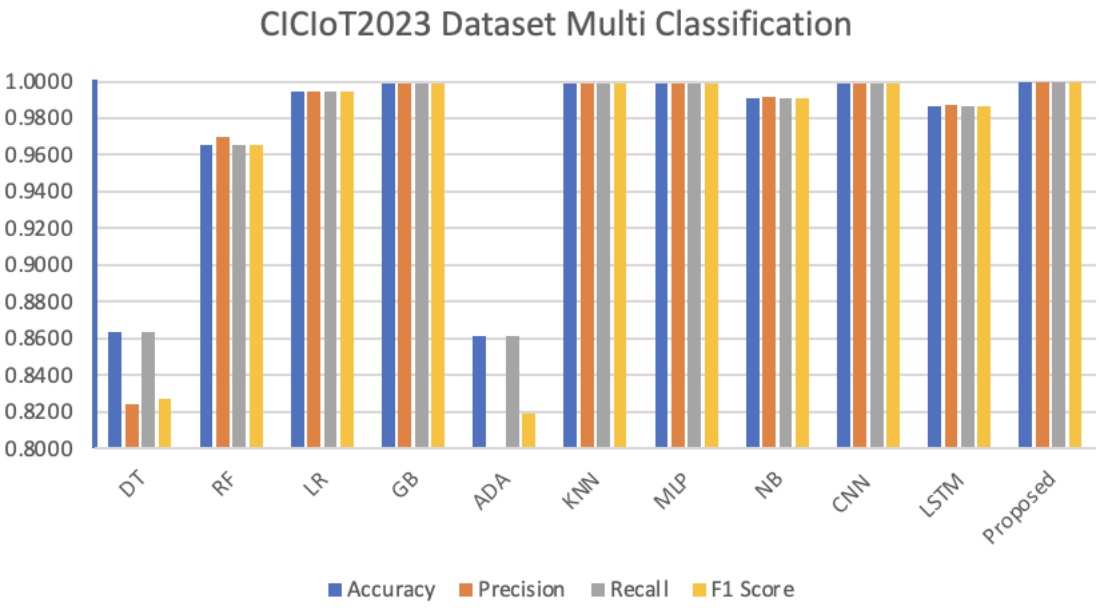

Figure 12. CICIoT2023 dataset multiclass classification result graph.

As can be seen in Table 7 and Figure 12, the proposed algorithm reached the highest multiclass classification accuracy value. This was followed by the MLP, CNN, and GB algorithms. The lowest results belong to the ADA and DT algorithms. The confusion matrix for testing the developed hybrid algorithm in multiclass is given in Figure 13. The diagram of the created ROC curve is presented in Figure 14.

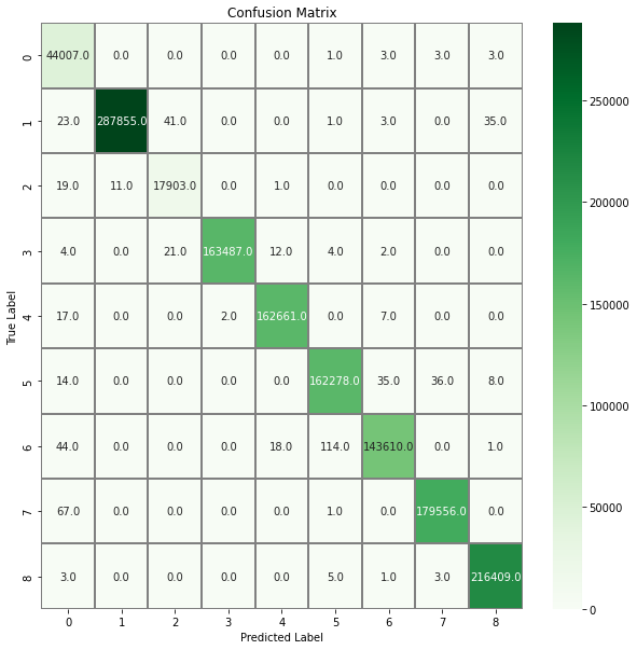

Figure 13. Confusion matrix for multiclassification of CICIoT2023 dataset.

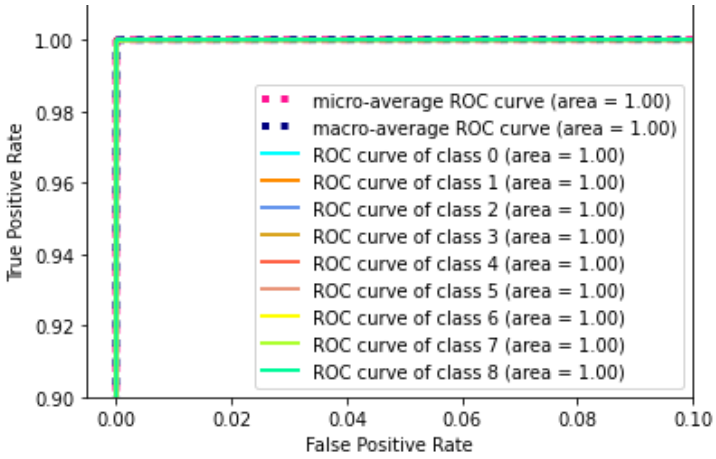

Figure 14. ROC curves for multiclassification of CICIoT2023 dataset.

According to the confusion matrix in Figure 13, the false positive rate (FPR) remained at very low levels, reaching a maximum value of 41 records in all combinations. True positive (TPR) records showed that the performance in one-to-one matching of classes reached high values. According to the ROC chart of the multiclass classification in Figure 14, the AUC-ROC value is close to 0.99 in all attack classifications. Table 8 compares the performance of our work with other state-of-the-art methods that are tested under the CICIoT2023 dataset. The comparison is conducted with respect to model, dataset, and accuracy value.

Table 8. Comparison of other works using CICIoT2023 dataset.

Article	Model	Dataset	Accuracy
Neto et al. (2023) [21]	RF, DNN, MLP, LR, AdaBoost	CICIoT2023	99.68% (b), 99.43% (m) (8 classes)
Wang et al. (2023) [22]	DL-BiLSTM	CICIoT2023	93.13% (m)
Proposed	Hybrid Deep Learning	CICIoT2023	99.995% (b), 99.96% (m) (9 classes)

According to the evaluation results above, there is a significant difference between binary classification and multiclass classification. As the attack classes in the dataset increase, the accuracy value of the algorithm decreases. In binary classification on the CICIoT2023 dataset, the machine learning and deep learning algorithms tested resulted in approximately similar outcomes. In multiclass classification, there was a significant decrease in the DT and AdaBoost algorithms. There is no serious decrease in the developed hybrid algorithm. In studies conducted using the CICIoT2023 dataset, the best results were achieved with the proposed hybrid algorithm, with rates of 99.995% in attack detection and 99.96% in attack type detection. The proposed algorithm reaches the highest value in terms of accuracy compared to other studies and other tested algorithms.

6.2. TON_IOT Dataset Result

The presented hybrid algorithm was also evaluated using the TON_IOT dataset. The attack detection accuracy rate of the ProcessedWindowsDataset-Windows10 dataset included in the dataset is given in Table 9. Figure 15 shows the results graphed.

As can be seen in Table 9 and Figure 15, the proposed algorithm reached the highest accuracy value in binary classification in the TON_IOT dataset. This was followed by the MLP, KNN, and CNN algorithms. The lowest result belongs to the NB algorithm. The confusion matrix of testing the developed hybrid algorithm as binary on the TON_IOT dataset is given in Figure 16. The diagram of the created ROC curve is presented in Figure 17.

Table 9. TON_IOT-Processed-Windows10 dataset binary classification results (%).

	Accuracy	Precision	Recall	F1 Score
DT	97.67	97.67	97.67	97.67
RF	98.23	98.23	98.23	98.23
LR	96.69	96.69	96.69	96.69
GB	97.50	97.49	97.50	97.44
ADA	92.96	93.01	92.96	92.22
KNN	98.50	98.58	98.50	98.52
MLP	98.54	98.57	98.54	98.55
CNN	98.32	98.32	98.32	98.32
LSTM	90.94	90.94	90.94	90.94
NB	76.98	88.07	76.98	77.83
Proposed	98.75	98.75	98.75	98.75

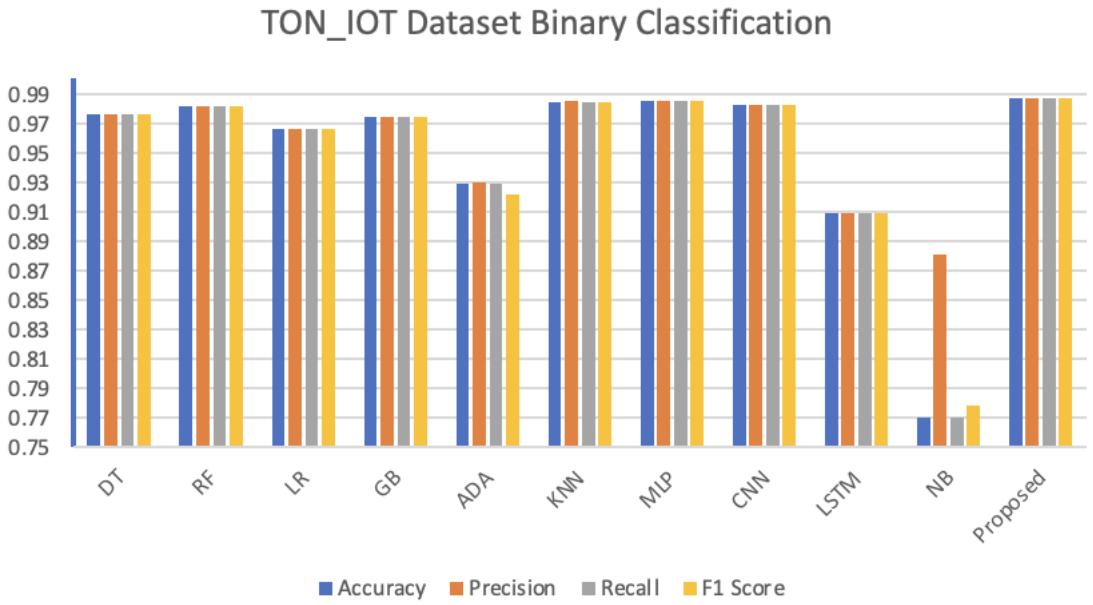

Figure 15. TON_IOT dataset binary classification results graph.

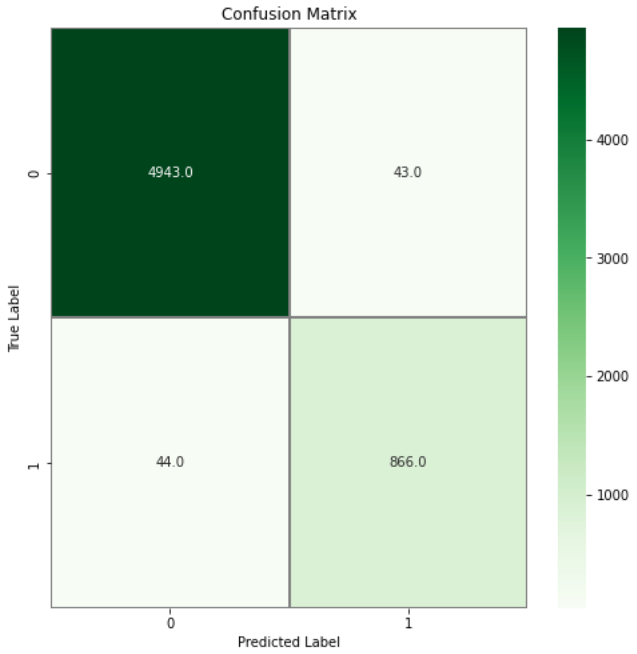

Figure 16. Confusion matrix for binary classification of TON_IOT dataset.

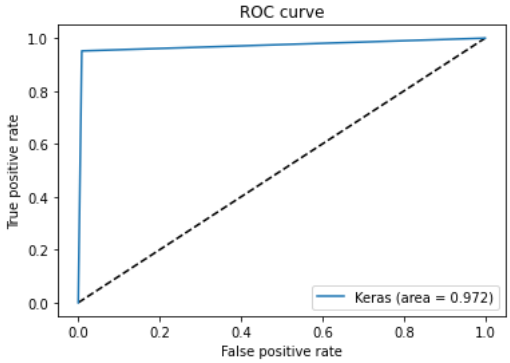

Figure 17. ROC curves for binary classification of TON_IOT dataset.

According to the confusion matrix in Figure 16, the false positive rate (FPR) remained quite low with a total of 87 records. According to the ROC curves of binary classification in Figure 17, the AUC-ROC value is close to 0.99. Table 10 lists previous studies using the TON_IOT dataset.

Table 10. Comparison of other works using TON-IoT dataset.

Article	Dataset	Accuracy
Kumar et al. (2020) [38]	TON_IoT	96.35%
Rehab et al. (2022) [39]	TON_IoT	98.39%
Hairab et al. (2023) [40]	TON_IoT	97.94%
Dobrojevic et al. (2023) [41]	TON_IoT	96.65%
Proposed	TON_IoT	98.75%

The developed algorithm was also tested on a different dataset, the TON_IOT dataset, so the reliability of the algorithm was demonstrated. Compared to state-of-the-art studies, the highest attack detection accuracy value rate of 98.75% was achieved in the TON_IOT dataset. As can be seen in Tables 8 and 10, the proposed algorithm has achieved better accuracy value than the studies performed on two datasets.

7. Discussion

The reason for using LSTM in the algorithm is that LSTMs, which are deep learning algorithms, are effective in capturing flow dynamics and maintaining information throughout the cycle. The LSTM algorithm is able to learn long-term dependencies and keep sequential data in memory. The forget gate in the LSTM algorithm decides whether the previous data will be forgotten or not. It has been regarded appropriate to use this structure in intense attacks, such as DDoS, thanks to the calculations made using sequential data. Another algorithm used in the classification stage is CNN. The CNN algorithm provides successful results in image classification, audio classification, and video classification and has strong capabilities to deal with classification problems by changing the depth and width of the network. Thanks to the convolution process, CNN can detect time-sensitive attack situations with fewer connections and parameters compared to standard feedforward neural networks with a similar number of layers. Since many features are extracted from the incoming data, it is effective in detecting attack types with unique features, such as DDoS [6]. The hybrid utilization of LSTM and CNN, leveraging their complementary features, has been observed to yield better results in conducted tests compared to their individual use.

Moreover, the analysis of the dataset in preprocessing steps and the removal of missing data have enabled making the data usable. By selecting the most relevant features from the dataset, the computational load of the algorithm is reduced, resulting in decreased training and testing costs. The preprocessing steps employed in our algorithm ensure that both training and test data are processed with noncomplex information.

Consequently, the developed algorithm has achieved a higher accuracy rate than all tested algorithms and state-of-the-art studies. The scope of the study has been expanded by evaluating the developed algorithm in multiclass classification and by assessing its performance on a different dataset.

Classification accuracy may give different results in different datasets even though the algorithms used are the same. This situation could be observed from the evaluation results of the TON_IOT dataset, which was used as the second dataset in the study. Compared to CICIoT2023, the accuracy values of the ADA and DT algorithms have increased in the TON_IOT dataset. Although the NB algorithm gave high results in multiclass evaluation, it remained at the lowest accuracy result in both binary evaluations. The NB algorithm does not consider interdependencies between features, which affects its accuracy [42]. The low

results of the NB algorithm in binary can be thought of as there are intense dependencies in binary class evaluation, and this affects the accuracy value.

8. Conclusions and Future Works

In our modern world, needs such as internet use and communication of devices with each other are inevitable. In addition to the benefits these technologies provide us, there are also cases of their abuse. One of the types of malicious use of network and communication channels is cyber attacks. The most commonly used method for this is DDoS attacks, which aim to restrict or completely make the use of target systems inaccessible. Detecting DDoS attacks is very crucial to be able to counter them. In this study, a new hybrid deep learning algorithm using CNN and LSTM deep learning models was developed to detect DDoS attacks. CICIoT2023 and TON_IOT datasets, which are current datasets, were used in training and testing this algorithm. Firstly, preprocessing and feature selection steps were applied to datasets. After, the proposed algorithm was tested as binary, and then it was tested as multiclass in the CICIoT2023 dataset. Algorithm evaluation was made by calculating accuracy, precision, recall, F1-score, and ROC data. As a result of these trainings and tests, a 99.995% attack detection rate and a 99.96% attack type detection rate were achieved. By achieving this high accuracy rate, a reference point has been created for future studies, contributing to the literature. In the evaluation made also using the TON_IOT dataset, an attack detection rate of 98.75% was reached. The proposed hybrid deep learning algorithm developed in this study is aimed to reach the highest accuracy value.

The accuracy of the developed hybrid algorithm may be increased by optimizing the deep learning algorithm parameters. One of these optimization methods is the metaheuristic approach. In future studies, it can be combined with modern and effective metaheuristic techniques to improve the optimization of errors in the algorithm to be developed [43].

To achieve high accuracy, a large volume of data must be used. The high volume of data used also increases training and testing times. In the physical world, it is crucial to detect attack traffic, such as DDoS, that requires rapid intervention by using system resources as efficiently as possible. As a future study, optimizing these training times and developing intrusion detection systems that have both high accuracy rates and low cost will be a great contribution to the literature.

Author Contributions: Conceptualization, S.Y. and M.D.; methodology S.Y.; software, S.Y.; validation, S.Y.; formal analysis, S.Y. and M.D.; investigation, S.Y.; resources, S.Y. and M.D.; data curation, S.Y.; writing—original draft preparation, S.Y.; writing—review and editing, S.Y. and M.D.; visualization, S.Y. and M.D.; supervision, M.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: We used the CICIoT2023 dataset and TON_IoT dataset, which are publicly accessed datasets (https://www.unb.ca/cic/datasets/iotdataset-2023.html, https://ieee-dataport.org/documents/toniot-datasets) (accessed on 24 November 2023), for the evaluation of the proposed IDS.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dener, M.; Al, S.; Orman, A. STLGBM-DDS: An Efficient Data Balanced DoS Detection System for Wireless Sensor Networks on Big Data Environment. *IEEE Access* **2022**, *10*, 92931–92945. [CrossRef]
2. Batchu, R.K.; Seetha, H. A generalized machine learning model for DDoS attacks detection using hybrid feature selection and hyperparameter tuning. *Comput. Netw.* **2021**, *200*, 108498. [CrossRef]
3. Al, S.; Dener, M. STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment. *Comput. Secur.* **2021**, *110*, 102435. [CrossRef]
4. Cil, A.E.; Yildiz, K.; Buldu, A. Detection of DDoS attacks with feed forward based deep neural network model. *Expert Syst. Appl.* **2021**, *169*, 114520. [CrossRef]

5. Almaraz-Rivera, J.G.; Perez-Diaz, J.A.; Cantoral-Ceballos, J.A. Transport and Application Layer DDoS Attacks Detection to IoT Devices by Using Machine Learning and Deep Learning Models. *Sensors* **2022**, *22*, 3367. [CrossRef]
6. Jia, Y.; Zhong, F.; Alrawais, A.; Gong, B.; Cheng, X. Flowguard: An intelligent edge defense mechanism against IoT DDoS attacks. *IEEE Internet Things J.* **2020**, *7*, 9552–9562. [CrossRef]
7. Alghazzawi, D.; Bamasag, O.; Ullah, H.; Asghar, M.Z. Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection. *Appl. Sci.* **2021**, *11*, 11634. [CrossRef]
8. Ferrag, M.A.; Shu, L.; Djallel, H.; Choo, K.-K.R. Deep learning-based intrusion detection for distributed denial of service attack in Agriculture 4.0. *Electronics* **2021**, *10*, 1257. [CrossRef]
9. Mamoudan, M.M.; Ostadi, A.; Pourkhodabakhsh, N.; Fathollahi-Fard, A.M.; Soleimani, F. Hybrid neural network-based metaheuristics for prediction of financial markets: A case study on global gold market. *J. Comput. Des. Eng.* **2023**, *10*, 1110–1125. [CrossRef]
10. Wei, Y.; Jang-Jaccard, J.; Sabrina, F.; Singh, A.; Xu, W.; Camtepe, S. Ae-mlp: A hybrid deep learning approach for ddos detection and classification. *IEEE Access* **2021**, *9*, 146810–146821. [CrossRef]
11. Kumar, P.; Bagga, H.; Netam, B.S.; Uduthalapally, V. SAD-IoT: Security analysis of ddos attacks in iot networks. *Wirel. Pers. Commun.* **2022**, *122*, 87–108. [CrossRef]
12. Alzahrani, R.J.; Alzahrani, A. Security Analysis of DDoS Attacks Using Machine Learning Algorithms in Networks Traffic. *Electronics* **2021**, *10*, 2919. [CrossRef]
13. Patil, N.V.; Krishna, C.R.; Kumar, K. SSK-DDoS: Distributed stream processing framework based classification system for DDoS attacks. *Clust. Comput.* **2022**, *25*, 1355–1372. [CrossRef]
14. Haq, M.A.; Khan, M.A.R.; AL-Harbi, T. Development of PCCNN-Based Network Intrusion Detection System for EDGE Computing. *Comput. Mater. Contin.* **2021**, *71*, 1769. [CrossRef]
15. Iwendi, C.; Rehman, S.U.; Javed, A.R.; Khan, S.; Srivastava, G. Sustainable Security for the Internet of Things Using Artificial Intelligence Architectures. *ACM Trans. Internet Technol.* **2021**, *21*, 1–22. [CrossRef]
16. Gamal, M.; Abbas, H.M.; Moustafa, N.; Sitnikova, E.; Sadek, R.A. Few-Shot Learning for Discovering Anomalous Behaviors in Edge Networks. *Comput. Mater. Contin.* **2021**, *69*, 1823–1837. [CrossRef]
17. Gad, A.R.; Nashat, A.A.; Barkat, T.M. Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset. *IEEE Access* **2021**, *9*, 142206–142217. [CrossRef]
18. Disha, R.A.; Waheed, S. Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique. *Cybersecurity* **2022**, *5*, 1–22. [CrossRef]
19. Kaur, J.; Agrawal, A.; Khan, R.A. P2ADF: A privacy-preserving attack detection framework in fog-IoT environment. *Int. J. Inf. Secur.* **2023**, *22*, 749–762. [CrossRef]
20. Verma, R.; Chandra, S. RepuTE: A soft voting ensemble learning framework for reputation-based attack detection in fog-IoT milieu. *Eng. Appl. Artif. Intell.* **2023**, *118*, 105670. [CrossRef]
21. Neto, E.C.P.; Dadkhah, S.; Ferreira, R.; Zohourian, A.; Lu, R.; Ghorbani, A.A. CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. *Sensors* **2023**, *23*, 5941. [CrossRef] [PubMed]
22. Wang, Z.; Chen, H.; Yang, S.; Luo, X.; Li, D.; Wang, J. A lightweight intrusion detection method for IoT based on deep learning and dynamic quantization. *PeerJ Comput. Sci.* **2023**, *9*, e1569. [CrossRef] [PubMed]
23. Guven, E.N. Examination, Design and Implementation of Intelligent Intrusion Detection Systems. Master's Thesis, Gazi University Graduate School of Natural and Applied Sciences, Ankara, Turkey, 2007.
24. Cebeloglu, F.S.; Karakose, M. A cyber security analysis used for unmanned aerial vehicles in the smart city. In Proceedings of the 2019 1st International Informatics and Software Engineering Conference (UBMYK), Ankara, Turkey, 6–7 November 2019; pp. 1–6.
25. Sreeram, I.; Vuppala, V.P.K. HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm. *Appl. Comput. Inform.* **2019**, *15*, 59–66. [CrossRef]
26. Chen, E.Y. Detecting TCP-based DDoS attacks by linear regression analysis. In Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, Athens, Greece, 21 December 2005; pp. 381–386.
27. Raptis, G.E.; Katsini, C.; Alexakos, C. Towards Automated Matching of Cyber Threat Intelligence Reports based on Cluster Analysis in an Internet-of-Vehicles Environment. In Proceedings of the 2021 IEEE International Conference on Cyber Security and Resilience (CSR), Rhodes, Greece, 26–28 July 2021; pp. 366–371.
28. Kumari, P.; Jain, A.K. A Comprehensive Study of DDoS Attacks over IoT Network and Their Countermeasures. *Comput. Secur.* **2023**, *127*, 103096. [CrossRef]
29. Ton IoT Dataset. Available online: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-ton-iot-Datasets/ (accessed on 22 October 2023).
30. Moustafa, N.; Keshky, M.; Debiez, E.; Janicke, H. Federated TON_IoT Windows datasets for evaluating AI-based security applications. In Proceedings of the IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020–1 January 2021; IEEE: Piscataway, NJ, USA; pp. 848–855.
31. Description of Windows 10 Features. Available online: https://cloudstor.aarnet.edu.au/plus/s/ds5zW91vdgjEj9i?path=/Descrip[]tion_stats_datasets/Description_stats_Windows_dataset#pdfviewer (accessed on 23 October 2023).
32. Kumar, P.; Gupta, G.P.; Tripathi, R. TP2SF: A Trustworthy Privacy-Preserving Secured Framework for sustainable smart cities by leveraging blockchain and machine learning. *J. Syst. Arch.* **2020**, *115*, 101954. [CrossRef]

33. Tsimenidis, S.; Lagkas, T.; Rantos, K. Deep Learning in IoT Intrusion Detection. *J. Netw. Syst. Manag.* **2022**, *30*, 8. [CrossRef]
34. Lin, S.; Tian, H. Short-Term Metro Passenger Flow Prediction Based on Random Forest and LSTM. In Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020; pp. 2520–2526. [CrossRef]
35. Khattak, A.; Asghar, M.Z.; Ali, M.; Batool, U. An efficient deep learning technique for facial emotion recognition. *Multimedia Tools Appl.* **2021**, *81*, 1649–1683. [CrossRef]
36. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
37. Chartuni, A.; Márquez, J. Multi-Classifier of DDoS Attacks in Computer Networks Built on Neural Networks. *Appl. Sci.* **2021**, *11*, 10609. [CrossRef]
38. Kumar, P.; Gupta, G.P.; Tripathi, R. An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks. *Comput. Commun.* **2020**, *166*, 110–124. [CrossRef]
39. Mohamed, R.H.; Mosa, F.A.; Sadek, R.A. Efficient Intrusion Detection System for IoT Environment. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 4. [CrossRef]
40. Hairab, B.I.; Aslan, H.K.; Elsayed, M.S.; Jurcut, A.D.; Azer, M.A. Anomaly Detection of Zero-Day Attacks Based on CNN and Regularization Techniques. *Electronics* **2023**, *12*, 573. [CrossRef]
41. Dobrojevic, M.; Zivkovic, M.; Chhabra, A.; Sani, N.S.; Bacanin, N.; Amin, M.M. Addressing Internet of Things security by enhanced sine cosine metaheuristics tuned hybrid machine learning model and results interpretation based on SHAP approach. *PeerJ Comput. Sci.* **2023**, *9*, e1405. [CrossRef] [PubMed]
42. Asharf, J.; Moustafa, N.; Khurshid, H.; Debie, E.; Haider, W.; Wahab, A. A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions. *Electronics* **2020**, *9*, 1177. [CrossRef]
43. Zhan, C.; Zhang, X.; Yuan, J.; Chen, X.; Zhang, X.; Fathollahi-Fard, A.M.; Wang, C.; Wu, J.; Tian, G. A hybrid approach for low-carbon transportation system analysis: Integrating CRITIC-DEMATEL and deep learning features. *Int. J. Environ. Sci. Technol.* **2023**, 1–14. [CrossRef] [PubMed]